

Distributed Machine Learning

Improved data protection for AI applications?

AT A GLANCE Distributed Machine Learning

- promises improved data protection by design and higher performance.
- trains Machine Learning (ML) models decentrally on end devices instead of centrally on a server.
- uses edge computing for AI and distributes the computing load.
- enables – often personal – training data to remain on end devices and thus with the the users.
- can use this data sovereignty to ensure the protection of personal data and increase informational self-determination.
- is already in use in the federated learning variant; other approaches are still at the research stage or on the threshold of market entry.
- can be used in a variety of ways, such as for mobility or health applications.

However, distributed Machine Learning creates new gateways for attackers and potentially creates a deceptive sense of security. Some experts therefore warn against exaggerated expectations in terms of data protection.

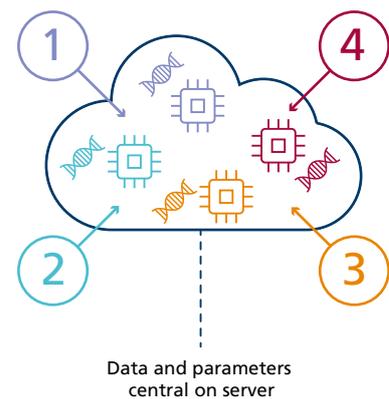
Starting Point

AI systems are based on training with large amounts of – sometimes sensitive – data. The use of this data is sometimes in tension with data protection and the individual's right to decide for himself or herself on the disclosure and use of personal data (informational self-determination). This case, for example, when an AI system only makes certain suggestions to users based on their search history and hides others that may be more suitable. At the same time, there are legal uncertainties for companies when training AI systems: According to the General Data Protection Regulation (GDPR), personal data may generally only be used for a specific purpose; for other purposes it may be necessary to obtain the subsequent consent of these persons or to balance the individual interests. The latter is complex and open to interpretation.

However, there are technical approaches that effectively combine data use and data protection – and may create new market opportunities for privacy-preserving AI solutions. These include the approach of distributed Machine Learning.

Classic Approach: Centralized Machine Learning

Currently common in AI development is the method of centralized Machine Learning (ML): A statistical model is trained centrally on a server (on the user’s premises or in the cloud). For this purpose, the server collects data from end devices such as smartphones or sensors (so-called clients) and bundles them centrally. The trained model can then be distributed to the end devices or applied to them. This form of Machine Learning is used, for example, in industrial production for predictive monitoring and maintenance of plants (predictive maintenance).



①②③④ Access of the end devices to ML model

Source: Own representation according to Warnat-Herresthal et al. 2021

Advantages		Disadvantages	
<p>Ex-ante: Compliance with data protection requirements can be ensured through central requirements</p>	<p>Data Privacy</p>	<p>Points of Attack: Sensitive data can be decrypted or accessed from a trained model under attack and central data collection enables direct access to sometimes sensitive (raw) data</p>	
<p>Concentration: Only server-side protection with regard to training of the ML model necessary</p>	<p>Security</p>	<p>Single Point of Attack: Possible attacks on servers threaten system security</p>	
<p>Single Point of Truth: Centralized architecture easy to grasp and to maintain</p> <p>Scaling: Compatible devices can be added without great effort</p>	<p>Technology</p>	<p>Interfaces: Integration of incompatible end devices or data formats not always possible</p>	
<p>Speed: Given uniform data source hardly any delay (latence time) between data collection and start of training of the ML model</p> <p>High Data Availability: Central instance for data processing strengthens efficiency and accuracy</p>	<p>Performance</p>	<p>Limited Possibilities for Real-Time Learning: Cumbersome uploading of complete data sets from end devices and distribution of the ML model from the server to end devices required</p>	
<p>Unambiguity: Clear attribution of the responsibility for training of an ML model, which always takes place on a central server operated by a provider</p>	<p>Ethics</p>		

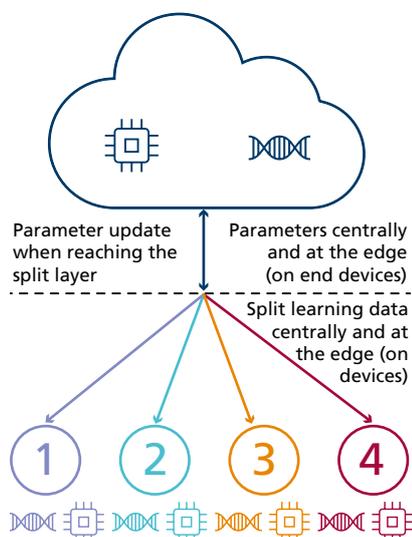
New Approach: Distributed Machine Learning

In distributed Machine Learning, the ML model is not trained on a central server. Instead, each end device (so-called client) accesses the current ML model and trains it locally with its own data set. In order to update and improve the ML model, only the training results (so-called weights), and not the data, are used. Three technical approaches to distributed Machine Learning are presented below.

Technical Approaches

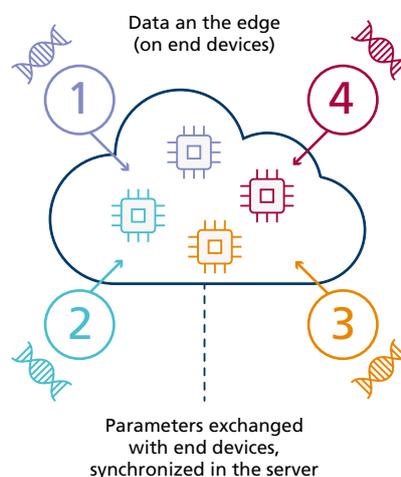
Split Learning – Learning on both, end devices as well as on server

- ML model is split into different submodels (so-called links) and is trained on both, end devices (clients) and on the server without sharing raw data (efficient distribution of the computing load)
- Iterative training process: at the split point of the ML model (so called split layer) end devices and servers swap only results (weights) of the trained ML model section (instead of raw data) and continue training with these results on their own data set (lower communication costs)
- Iterations end when convergence between the ML models of the end devices and the server has been established



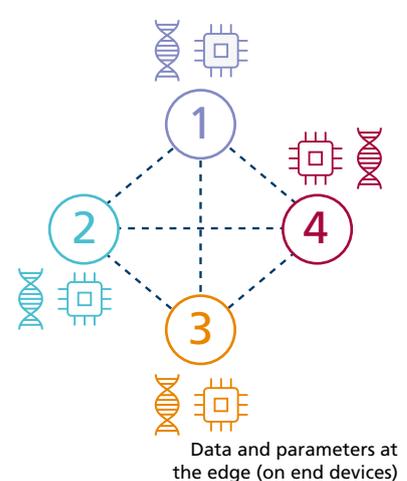
Federated Learning – Learning with central server as aggregation instance

- End devices download parameters of the ML model from the server
- ML model is trained on end devices with local data set
- Solely weights are sent to the server from end devices, local data set remains with the end device
- No training but only composition of the weights for central updating of the ML model (inference) on the server
- Server provides parameters of the improved, because synchronized ML model to end devices for new training
- Repeatable process, in which the distributed ML model is constantly being optimized



Swarm Learning – Learning on distributed devices without aggregation instance

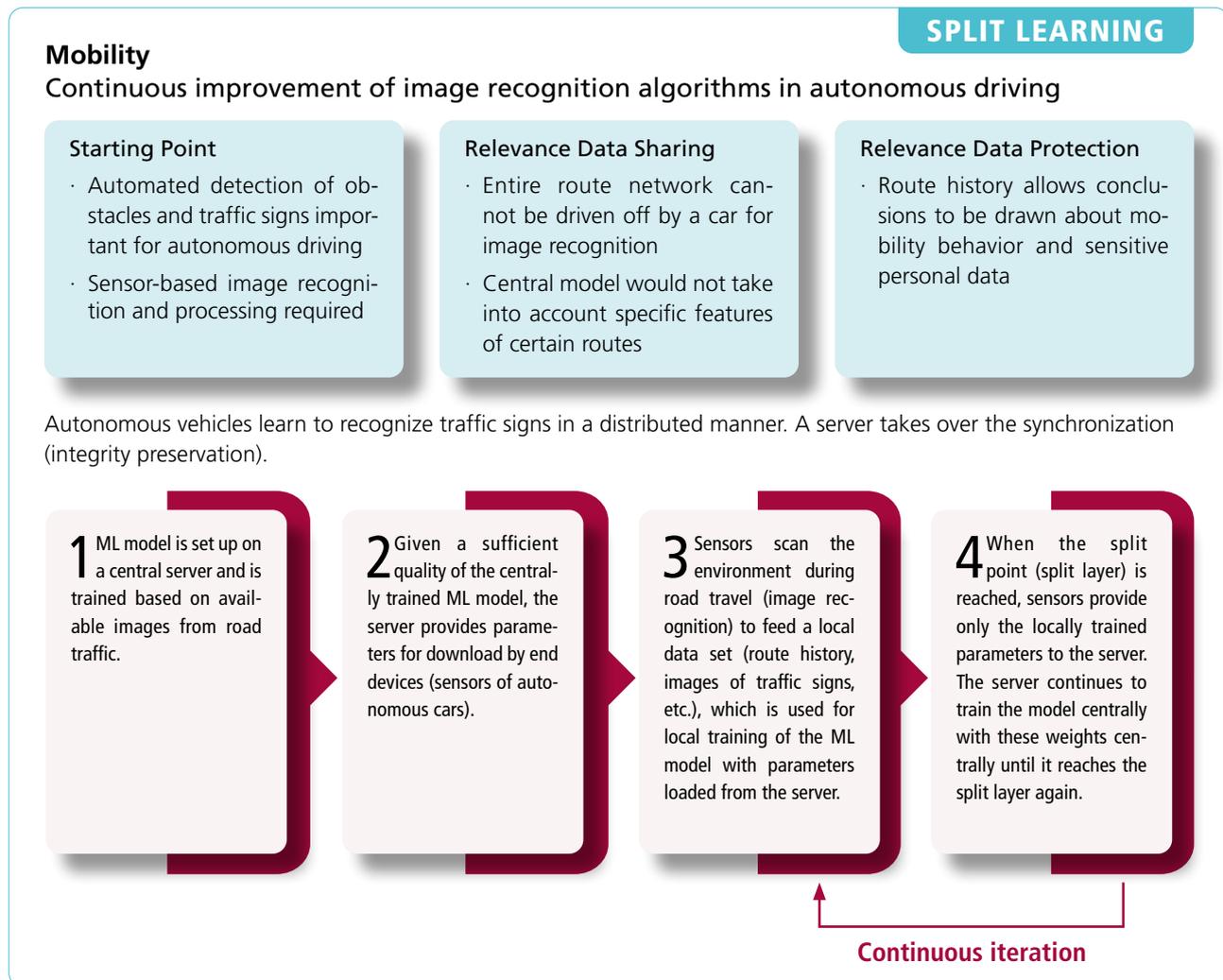
- Parameters of the ML model stored in access-restricted blockchain instead of on central server
- No coordination instance, but a central instance for the pre-authorization of end devices necessary for access to blockchain
- End devices load parameters of the ML model from blockchain and train it with local data set
- After training, only the adjusted weights are stored in the blockchain
- Adjusted weights and parameters of the ML model can be read out by end devices and be assembled locally to the overall model



Source: Own representation

Application Examples

Distributed Machine Learning is starting to be used in business – especially in the mobility and healthcare sectors, but also in other areas. In the following, an application example is outlined for each of the three technical approaches presented above. In practice, these applications could also be implemented by other distributed Machine Learning approaches.



FEDERATED LEARNING

Autocompletion and Correction

Continuous improvement of smartphone word suggestions

Starting Point

- Autocompletion important for user experience
- Adaptivity to individual (language) habits increases its quality and accuracy

Relevance Data Sharing

- Quality potential through scaling effects across billions of smartphones
- Upload of all text data to central server would exceed capacity limits

Relevance Data Protection

- Written texts allow inferences, e.g. about life situations or trade/business secrets

For synchronization and improvement of autocompletion, the ML model is trained on the smartphones of the users (end devices); weights are uploaded to the server.

1 The smartphone stores information about the context when text is created and whether users click on a search suggestion (creation of data set).

2 The ML model is trained locally on the smartphone with the data set.

3 Weights are uploaded to the server. The server synchronizes the adjusted parameters of all smartphones to improve the suggestion model.

4 The server provides the synchronized parameters for download for smartphones to further improve autocompletion.

Continuous iteration over the first four steps

SWARM LEARNING

Health

Identification of disease cases (leukemia, tuberculosis, covid-19)

Starting Point

- Big Data Analytics based on individual health data enables detection of diseases
- Prediction via blood transcriptomes (totality of all RNA molecules of a cell)

Relevance Data Sharing

- Statistical predictions only possible with many data points (large-N-problem)
- Individual parameters allow conclusions only in comparison with totality

Relevance Data Privacy

- Personal health data extremely sensitive
- Danger of violation of personal rights (e.g. improper use for health-related classification of individuals by insurance companies)

For transcriptome-based disease prediction, the ML model is trained in a decentralized manner at clinics with locally collected patient data.

1 The ML model is distributed on the blockchain at different clinics (so-called nodes), which have blockchain access via health insurance approval.

2 Transcriptomes of patients are recorded individually in each clinic (local data set).

3 Current parameters of the ML model are retrieved by individual clinics from the blockchain and assembled locally to form the overall model.

4 ML model is trained with respective local data set. Weights (parameters of the updated ML model) are stored in the blockchain.

Continuous iteration

Potentials and Challenges: Analyzed at a Glance

The practical application of distributed Machine Learning approaches seems promising. Nevertheless, there are challenges that cannot be neglected and are already being addressed. In the following, potentials and challenges of distributed Machine Learning are compared.

Potentials	Challenges	
<p>Data Sovereignty: Data remains on the user's end device for training of the ML model</p> <p>No Data Pooling: No technical requirement to exchange personal data</p> <p>Cooperation: Different organizations can use the ML model jointly without the need to exchange critical data</p>	<p>Data Privacy</p> 	<p>Privacy: Model updates allow conclusions to be drawn about personal data</p> <p>De-Anonymization: Still technically possible to check whether data of specific persons are contained in the training data set (but harder)</p>
<p>Distributed Risk: Data set-based attacks more difficult due to distribution of data across end devices and due to lower reach</p> <p>Resolution of Single Point of Attack: Model and data set are separated from each other complicating extensive attacks</p>	<p>Security</p> 	<p>Low Local Protection: Less computing power available for attack detection on end devices (weakness of edge computing)</p> <p>New Attack Targets: Data sets on end devices (e.g. against data poisoning) as well as indirect access to ML model via local data sets (e.g. against model poisoning) must be protected</p>
<p>Swarm Intelligence: Quality of overall ML model increases with the number of participating end devices</p> <p>Tolerance: Learning with diversified data sets and heterogeneous end devices simplified, since only weights are exchanged</p> <p>Hardware Efficiency: Distribution of ML training across end devices reduces server hardware requirements</p>	<p>Technology</p> 	<p>Dependency: Training an ML model with a too small number of end devices can compromise model quality</p> <p>Interoperability: Compatibility of heterogeneous end devices and mastery of statistical heterogeneity must be ensured</p>
<p>Low Latency: Distribution of computing power over end devices (edge computing) allows faster model training with larger data volumes</p> <p>Real-Time Predictions: Can be performed directly without an Internet connection if current model parameters are available locally.</p>	<p>Performance</p> 	<p>Risk Factor Internet Connection: Required for parameter exchange between server and end devices and could generate latencies in case of instable connection</p> <p>Communication as Bottleneck: Efficient methods necessary to keep communication effort for parameter exchange low</p>
<p>Data Sovereignty: Informational self-determination is strengthened through paradigm of decentralization</p> <p>Sovereignty: Users with a stronger position vis-à-vis data processors</p>	<p>Ethics</p> 	<p>Discrimination: If the end devices involved in the training do not adequately represent the population, the distributed ML model does not adequately account for minorities</p>

Expertise from Plattform Lernende Systeme



Distributed Machine Learning opens up new possibilities for effective and scalable use of data without having to share it. This enables many useful applications with sensitive data possible in the first place.

Prof. Dr. Ahmad-Reza Sadeghi, Head of System Security Lab, Technical University of Darmstadt

Distributed Machine Learning does not require the merging of sensitive data. This avoids the risks of centralized data collection creating advantages in terms of data protection. The task now is to clarify the open legal, technical and organizational questions for legally compliant use.

Dr. h.c. Marit Hansen, State Commissioner for Data Protection Schleswig-Holstein



AI systems in medicine can only be successful if they have the necessary amounts of data to achieve high accuracy. Distributed Machine Learning represents one of the most important technical options for making this possible while preserving the informational self-determination of the individual.

Prof. Dr. Björn Eskofier, Chair for Machine Learning and Data Analytics, Friedrich-Alexander-University Erlangen-Nürnberg

Outstanding Issues

- **Cost-benefit assessment:** Do the expected benefits of distributed Machine Learning in terms of privacy and performance really prevail potential disadvantages?
- **Practical test:** How do the technical approaches prove themselves in economic practice and how big is their market?
- **Target orientation:** Which technical approaches are suitable for which application domains?
- **Security:** How can emerging attack vectors (e.g. against exchange of weights between endpoints) be closed without limiting performance?

Glossary

Adversarial Attack: Attack aimed at manipulating the training data set of an AI system, e.g., by misclassification inputs; attackers inject malicious content into the filter of a Machine Learning algorithm so that the system misclassifies a certain data set.

Edge AI: Shifting the training of ML models to end devices and at most exchange of metadata with central server.

Machine Learning Model (ML Model): Statistical model that has been trained to recognize certain types of patterns. The ML model enables new data to be analysed and to make predictions about this data.

Weights: “Result” of a (locally) trained ML model, which can be assembled into an overall model. Weight exchange could make the exchange of whole data sets obsolete.

Further Reading

Beyerer, J., Müller-Quade, J. et al. (2022): KI-Systeme schützen, Missbrauch verhindern. Maßnahmen und Szenarien in fünf Anwendungsgebieten. Whitepaper der Plattform Lernende Systeme. Online unter: https://doi.org/10.48669/pls_2022-2 (letzter Zugriff: 20.06.2022)

Houdeau, D. (2022): Wie wir KI-Systeme vor Cyberangriffen schützen. Plattform Lernende Systeme. Online unter: <https://www.plattform-lernende-systeme.de/reden-und-beitraege-newsreader/wie-wir-ki-systeme-vor-cyberangriffen-schuetzen.html> (letzter Zugriff: 20.06.2022)

Kaissis, G. A. et al. (2020): Secure, privacy-preserving and federated machine learning in medical imaging. Nature Machine Intelligence, 2, 305–311.

Warnat-Herresthal et al. (2021): Swarm Learning for decentralized and confidential clinical machine learning. Nature, 594, 265–270.

Content from AI AT A GLANCE may be used for editorial purposes provided the source, Plattform Lernende Systeme, is acknowledged.

Imprint

Expertise: Björn Eskofier, Marit Hansen, Ahmad-Reza Sadeghi

Editors: Jan Biehler, Birgit Obermeier

Editorial staff: Lernende Systeme – Germany’s Platform for Artificial Intelligence | Managing Office | c/o acatech | Karolinenplatz 4 | D-80333

Munich kontakt@plattform-lernende-systeme.de | www.plattform-lernende-systeme.de

Status: September 2022 | Photo credits: Kurt Fuchs, Markus Hansen, Technical University of Darmstadt, p. 7

Follow us on [Twitter](#) und [LinkedIn](#).

SPONSORED BY THE



Federal Ministry
of Education
and Research