FeatureCloud

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

Privacy preserving federated machine learning and blockchaining for reduced cyber risks in a world of distributed healthcare

Deliverable D2.4
Set of (novel) attack vectors and countermeasures

Work Package
WP2 Cyber risk assessment and mitigation

Disclaimer

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078. Any dissemination of results reflects only the author's view and the European Commission is not responsible for any use that may be made of the information it contains.

Copyright message

Document information

Grant Agreement Number: 826078			**Acronym: FeatureCloud**	
Full title	Privacy preserving federated machine learning and blockchaining for reduced cyber risks in a world of distributed healthcare			
Topic	Toolkit for assessing and reducing cyber risks in hospitals and care centres to protect privacy/data/infrastructures			
Funding scheme	RIA - Research and Innovation action			
Start Date	1 January 2019	**Duration**	60 months	
Project URL	https://featurecloud.eu/			
EU Project Officer	Christos MARAMIS, Health and Digital Executive Agency (HaDEA) - Established by the European Commission, Unit HaDEA.A.3 – Health Research			
Project Coordinator	Jan BAUMBACH, UNIVERSITY OF HAMBURG (UHAM)			
Deliverable	D2.4 Set of (novel) attack vectors and countermeasures			
Work Package	WP2 Cyber risk assessment and mitigation			
Date of Delivery	**Contractual**	31/12/2023	**Actual**	17/12/2021
Nature	Report	**Dissemination Level**	Public	
Lead Beneficiary	05 SBA			
Responsible Author(s)	Rudolf Mayer & Anastasia Pustozerova (SBA Research); Walter Hötzendorfer (RI)			
Keywords	Mitigation Mechanisms, Input & Output Privacy, Model Watermarking & Fingerprinting, Data Exfiltration			

History of changes

Version	Date	Contributions	Contributors (name and institution)
V0.1	25/11/2021	First draft	Rudolf Mayer, Anastasia Pustozerova, Maroua Jaoua & Isabell Lederer (SBA), Walter Hötzendorfer (RI)
V0.2	08/12/2021	Comments & Review	Andreas Holzinger (MUG)
V0.3	13/12/2021	Draft	Rudolf Mayer & Anastasia Pustozerova (SBA)
V1	17/12/2021	Final version	Rudolf Mayer (SBA), Walter Hötzendorfer (RI)
V1	17/12/2021	Final approval	Jan Baumbach (UHAM)
V1	17/12/2021	Submission	Miriam Simon (concentris)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

Table of Content

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

1 Objectives of the deliverable based on the Description of Action (DoA)

The main objective of WP2 is the definition of a security and privacy architecture **based on the cyber risk assessment and legal requirements**.

"While the approach of the FeatureCloud project by design mitigates most major concerns regarding security and privacy, the underlying foundations of the platform to be developed need to be secured thoroughly, especially considering the diversity of the local execution platforms on the hospital sites. Important attacker goals include the theft of data on a local level, as well as the theft and manipulation of results, with the inclusion of possible insider attacks."

Objective 1 of this work package thus aims ***"to select, adapt and further develop mitigation mechanisms, especially considering the detection of malicious algorithms pushed to the local execution platforms, as well as the exfiltration of data besides feature vectors"***.

The corresponding task in this work package is ***Task 4: Mitigation Mechanisms***:
"SBA and RI will develop mitigation strategies in order to solve the issues detected by the risk assessment of the architecture and in order to follow all technical and legal requirements. In addition, methods will be developed for dealing with malicious providers of algorithms that might have been designed in order to reveal sensitive information, or simply to extract and exfiltrate the whole data set provided in the respective local execution platform. This task also includes the development of reactive methods, i.e., methods that can be used to detect misuse of data like data fingerprinting, in order to thwart information theft. The main reason for this is that not only privacy is an important asset requiring protection, but that also the feature vectors, while irrelevant with regard to privacy, possess a potentially high (monetary) value."

2 Executive Summary

Deliverable D2.1 "Risk assessment methodology" identified several risks associated with federated learning; among different cybersecurity frameworks identified, the CIA triad is a model that groups risks among the three dimensions Confidentiality, Integrity and Availability. While each of these is important, the main focus of the FeatureCloud project is on privacy-preserving, federated machine learning, and as a consequence, the mitigation mechanisms are focused on attacks that threaten the privacy of individuals and the obtained assets, and thus focus on the threats to **confidentiality**. We can roughly distinguish the following areas of mitigation mechanisms addressing confidentiality threats within the FeatureCloud platform

- Threats to the confidentiality of input data used for learning, which consist of
  - Threats of directly disclosing input data, e.g., with **data exfiltration** techniques.
  - Threats to indirectly disclosing information from the learned federated, aggregated models, e.g., membership inference.
- Threats to the confidentiality of the assets themselves, e.g., protecting the learned models from unauthorised usage and sharing.

FeatureCloud apps that perform the model learning (or other steps, such as pre-processing or visualisation) are very diverse and can be developed and implemented by a large range of contributors. Consequently, it is important to provide generic components that can be utilised by a large number of apps, and the proposed mechanisms follow this strategy. To address these, we have designed a number of mechanisms that are generic and can be used in specific apps. Therefore, generic mechanisms to protect the **input** to the aggregation via secure multi-party computation, and mechanisms to protect input data that is encoded in the **output** of the aggregation

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

(e.g., the models) via differential privacy will be available in the platform. Further, watermarking and fingerprinting mechanisms will protect the intellectual property of the output (e.g., the models). Finally, data exfiltration will be countered on the one hand by the mechanisms designed in Deliverable D2.2 “KPIs and metrics for local execution platforms”, such as the Key Performance Indicator (KPI) #3 “Privacy Requirements for Federated Algorithms Working on Patient Data”, which analyses the amount of exchanged data and can thus spot mass-exfiltration, and on the other hand by analysing the machine learning model itself as an exfiltration vehicle, and providing generic defences against this threat.

3 Considered Threats & Mitigations

Based on the description of action from the grant agreement, as well as the risk assessment carried out in Deliverable D2.1 “Risk assessment methodology”, for this deliverable, first the most relevant threats are identified. Deliverable D2.1 identified, as one means to categorise threats against federated machine learning, the **CIA** (Confidentiality, Integrity and Availability) **model**. In this model, confidentiality deals with **limiting access to information**, integrity is the assurance that the **information is trustworthy and accurate**, and availability is a guarantee of **reliable access to the information by authorised people.** Vulnerabilities can be viewed by the angle of one or more of these three concepts. A graphical representation is given in Figure 1.

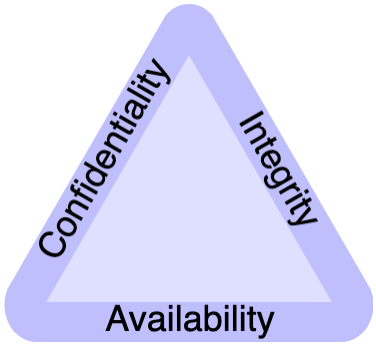


Figure 1: Confidentiality, Integrity and Availability (CIA) triangle / triad

FeatureCloud provides a privacy preserving machine learning framework based on federated learning. FeatureCloud services aim to ensure primarily the confidentiality of sensitive medical data. Integrity and availability are very relevant aspects for any system in the medical domain and providers of individual hardware and software components within the FeatureCloud ecosystem are strongly encouraged and monitored to follow the best-practises recommendations identified in Deliverable D2.1 “Risk assessment methodology”. The main objective of FeatureCloud is to provide **privacy-preserving analysis** of (mostly) healthcare and medical data. This corresponds to the legal requirements of data protection law. Therefore, this deliverable focuses on mitigation mechanisms addressing mostly threats to the privacy of individuals that go beyond aspects that can be solved with traditional security hardening and testing. However, it is emphasized again that these mechanisms need to be implemented as well, mostly in the local nodes, which is also a requirement of data protection law.

In many settings, confidentiality is roughly equivalent to privacy, though that it might, in general, apply to any other valuable, sensitive information. We also address these issues more generically, threats to the **confidentiality** of data and derived assets along the entire analysis process. Measures ensuring confidentiality are designed to prevent sensitive information from reaching non-intentioned targets.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

We focus specifically on novel attack vectors in federated learning systems. Principally, the participants in a federated learning-based system do not have to share their private or sensitive data. Instead, they train a machine learning model on their data and share this model with other participants of federated learning. These machine learning models, however, still can leak information about the data they were trained on. Consequently, we propose defence mechanisms that will ensure the confidentiality of federated learning. Privacy aspects and vulnerabilities of federated learning are still being actively researched, and novel vulnerabilities, threats, and attack vectors are constantly researched, as well as novel or improved countermeasures to those. This document is reflecting the landscape and state-of-the-art at the time of writing; however, activities within work package 2 of FeatureCloud, as well as other related work packages providing apps and the overall framework, will continue to monitor the quickly changing threat landscape, and react to arising issues.

Deliverable D2.1 “Risk assessment methodology” identified the following attacker models in federated learning (for more details, please refer to D2.1):

- An **insider** attacker participates in the federated learning process and has **access to** (some of) **the models** during training. We distinguish:
  - A participant insider attacker is represented by one or several nodes, the owners of the data, who train models locally (e.g., described by (Bagdasaryan et al., 2020) .
  - A coordinator plays an aggregator role that collects locally trained models from data owners (e.g., considered by Wang et al., 2019).
- An **outsider** attacker has access only to the final model after the federated learning process is finished.

We further distinguished between two typically considered adversary models. **Semi-honest** (or honest-but-curious) adversaries perform a “passive” attack, following the protocol, but trying to gather more information than the protocol allows. **Malicious** adversaries perform “active” attacks and arbitrarily deviate from the protocol.

Regarding the attacker’s knowledge of the targeted system, we observed the following specialisation to machine learning settings (e.g. considered by (Shokri et al., 2017) and (Truex et al., 2019c) :

- In the **white-box setting**, the adversary has full access to the model, and **knowledge about its architecture**, model weights and hyperparameters.
- In **grey-box** access, the attacker has access only to some information about the model, e.g., information on specific layers of the model, or some intermediate results.
- In the **black-box** scenario, the attacker can **use the model for making predictions** (via some API / service), but there is no access to any other information about the model. In this case, the attacker can use the machine learning model only as a service to query output for some specific input, to infer some valuable information for implementing attacks.

Figure 2 shows that there are, in principle, two broad categories of attack surfaces to breach data confidentiality in federated learning:

- Through local training resp. models, which is mostly congruent with the case of an insider attack
- Through global models e.g., if the global model gets available to an outside attacker

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

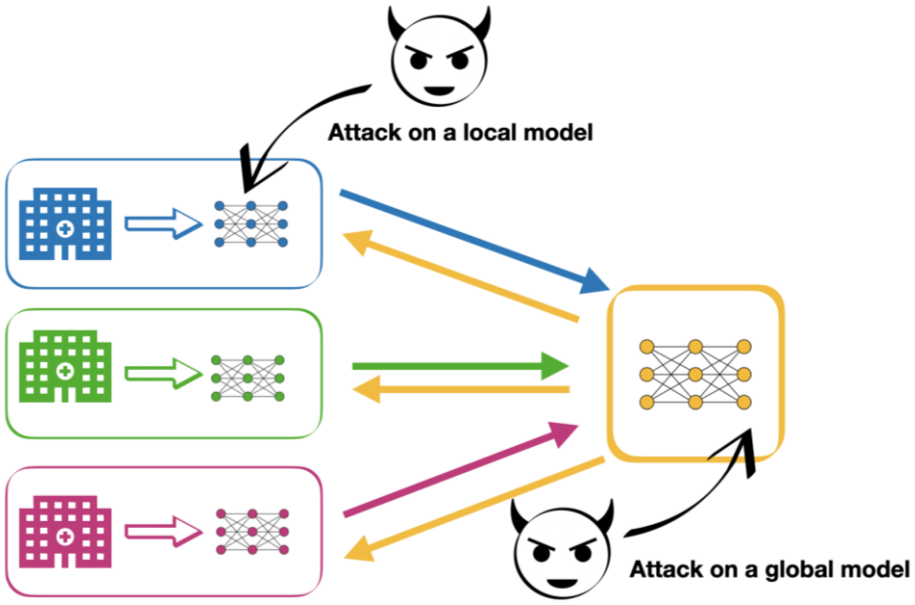


Figure 2: Attacks in federated learning with the goal to breach confidentiality through access to the local or global models.

Among the confidentiality axis, we thus, in particular, provide mitigations to the following risks, **derived from Deliverable D2.1** and the **objectives in the FeatureCloud grant agreement**:

- Insider attacks by the coordinator of the federated learning process. The coordinator has access to the **inputs** to the aggregation process, which corresponds to the outputs from the participants (local nodes), and might be e.g., in the form of machine learning models, gradients, or other parameters. To avoid access to this information in clear text, several approaches are possible. On the one hand, homomorphic encryption (C. Zhang et al., 2020) could be utilised to provide the aggregator with only encrypted data. Another option is to use **Secure Multiparty Computation** (SMPC, or sometimes just MPC) (Bonawitz et al., 2017), which allows computing an output without the need of a central aggregator.
- Insider and outsider attacks on the final model with the aim to **infer information** on the underlying training data, e.g., a model inversion, membership inference, and similar attacks. These attacks are often very scenario-specific, e.g., they depend on the exact type of data used, or the type of (federated) learning performed, as well as the model types. Thus, it is difficult to provide custom mechanisms for each setting. However, several studies (Truex et al., 2019b) in these areas have suggested that **Differential Privacy** can mitigate these risks. As Differential Privacy allows to provide a relatively generic solution that is independent of the exact scenario, this will be provided as a generic mitigation mechanism within FeatureCloud.
- Direct attacks on the original training by performing a **data exfiltration** from the local, secured environments. Common attacks try to utilise a dedicated channel through which the data is exfiltrated, e.g., a connection to an external server where the data will be stored (through whichever protocol, e.g., SSH or FTP). In the setting of FeatureCloud, we aim to prevent this by two means. On the one hand, the code audit and certification of apps can provide certain protection, complemented by mechanisms to secure the local environment. On the other hand, the KPIs identified in Deliverable D2.2 “KPIs and metrics for local execution platforms”, address this as well, from a more grey-box to black-box approach. As

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

such, KPI3 “Privacy Requirements for Federated Algorithms Working on Patient Data” analyses the amount of exchanged data and can thus spot mass-exfiltration attacks. We consider another channel as relevant within the FeatureCloud project, namely using the machine learning model itself as an exfiltration vehicle. This deliverable details this attack and generic mitigation mechanisms.

- Finally, a different type of confidentiality attack is more related to the **theft of intellectual property**, in the form of a trained machine learning model. Training these models, in a centralised or federated way, often requires a significant number of resources (valuable training data, computing resources, domain and data science experts, …), and thus represent a significant value. Attackers might want to obtain such a model from a legitimate source, and use it for their own purposes, e.g., a monetisation. We thus provide a reactive mitigation mechanism, by model watermarking.

This leads to the following mitigation mechanisms as highest-priority mechanisms to implement: **Secure Multiparty Computation** to ensure confidentiality of local models (**input** confidentiality), **Differential Privacy** to ensure the privacy of a global model (**output** privacy), **watermarking and fingerprinting** as a reactive method for protecting the property of trained models, and **defences** against model-based **data exfiltration** methods.

This is also depicted in detail in Figure 3, which shows the detailed FeatureCloud architecture, as well as some of the attack points and corresponding mitigation mechanisms. The “App Instance” needs to be secured against data exfiltration attacks. This is achieved by sandboxing the app, the code audit, and defences against data exchanged via the learned model. Further, the network connection needs to be secured by standard encryption and other techniques laid out in Deliverable D2.1 “Risk assessment methodology” and Deliverable D2.2 “KPIs and metrics for local execution platforms”. Deliverable “D7.3 Federated machine learning apps running in app store” already utilises encryption for securing the communication via the relay server (cf. Section 4.2 in that deliverable). SMPC will further mitigate the issue of confidentiality of the exchanged parameters during aggregation, as even if the encryption mechanism would be broken, they are still e.g., just representing the shares.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

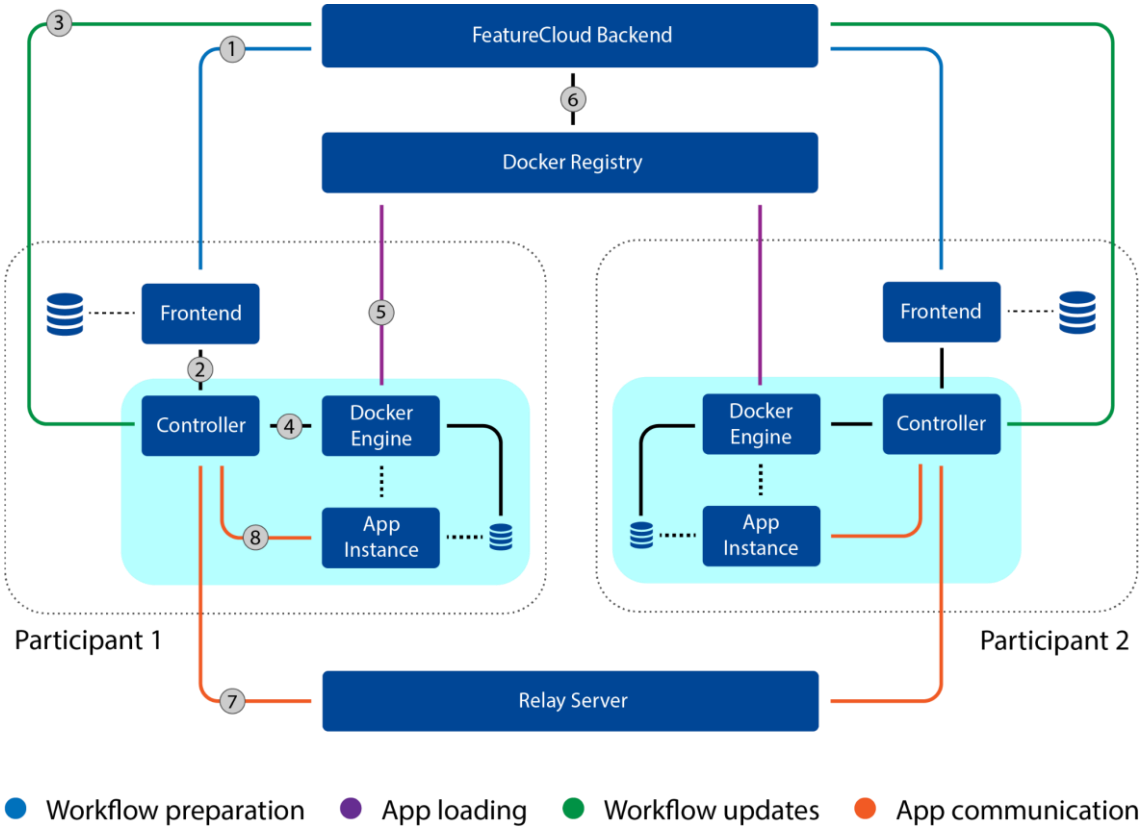


Figure 3: FeatureCloud architecture and communication within the system, with two participants

4 General Process for Privacy Assessment

Besides the mitigation mechanisms that are discussed in Section 5, which will be generically and globally provided, every FeatureCloud app should individually be evaluated in terms of vulnerabilities to the confidentiality of the data released, i.e. the global model (final or intermediate versions), exchanged local models or parameters (gradients, …), and other information derived from the original input data, i.e. assess the risks to the confidentiality of the output of the local and global training.

While the exact threats are very specific to each algorithm, the required exchanged data, as well as to the data set, we provide general guidelines and recommendations in the following, to derive proper threat and attacker models for each app, and to assess risks of inference.

4.1 Threats to Confidentiality and Disclosure Models

Common taxonomies (e.g., (Xiong et al., 2009), (Prasser et al., 2014)), mostly originating from the domain of data publishing, distinguish the following types of disclosure:

- **Identity disclosure** is generally considered the strongest form of disclosure, and means that an attacker can associate an individual to a specific record. This is also referred to as **re-identification**. It is often achieved via a record-linkage attack, where the target dataset is correlated to other data available to the attacker (public, or private). Several scenarios can result in identity disclosure, (Garfinkel, 2015) mentions the following three (for published datasets):

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

- *Insufficient de-identification* may lead to identifying information inadvertently remaining in a de-identified dataset (e.g., (Sweeney, 2013)).
- *Re-identification by linking* allows to re-identify specific records by linking some of the remaining data with similar attributes in another, identifying dataset.
- *Pseudonym reversal*

- **Attribute disclosure** (Elliot, 2005) (El Emam, 2011) means that an attacker can learn (exactly, or approximately) the **value of one (or more) attributes of an individual** that are contained in the targeted database. For example, an attacker might learn the approximate salary or the medical diagnosis, which were previously unknown. This is performed by knowing the other attributes of an individual in the database, maybe even by knowing just some of the quasi-identifying attributes. It might be achieved even without uniquely associating an individual to a specific record in a dataset, if the set of records an attacker might narrow a match down to still has all (or most) records containing the same (or similar) sensitive values.
- **Membership disclosure** is generally considered to be the weakest form of disclosure. Here, an attacker can infer whether or not an individual is contained in a dataset. Unlike the other disclosure settings, it does not directly release attributes from the dataset, i.e., it **does not directly unveil information from the dataset itself**. However, an attacker might infer additional information from the membership of an individual in a dataset. For example, if this dataset is a medical dataset containing information about patients carrying a certain disease, the attacker might infer that this individual also suffers from this disease (as otherwise, (s)he might not have been included in the dataset.

While this categorisation was primarily developed for data publishing, it can also be applied to publishing models learned from this data, as we will discuss in Section 4.3.

4.2 Attacker Models

We can then further distinguish multiple attacker models, depending on the information available to the attacker, and the attacker's goal (Elliot and Dale, 1999). While these models were primarily developed for the re-identification risks in statistical disclosure control of datasets, they can also be used for distinguishing different attackers inferring information from machine learning models (which in most broad terms could be seen as a perturbation of the original input data):

- The **prosecutor model** means an attacker that targets a specific individual. The attacker is assumed to already know that the data about that specific individual is contained in the dataset that was used to train the model. The attacker is thus assumed to have access to an identifying database that contains all and only the records from the disclosed anonymised database, and wants to re-identify a specific person in a de-identified database. In this model, one can identify which samples are the most vulnerable to de-identification.
- The **journalist model** still targets one individual, but does not care which specific individual is re-identified. However, it is not assumed that the attacker has knowledge on whether the individual was part of the training set, or not. In other words, the identifying dataset that the attacker has available is not equal to the de-identified dataset, but has a number of samples in their intersection.
- The **marketer** model differs in the target - here, the attacker aims at re-identifying a large(r) number of individuals. The Marketer model (Dankar and El Emam, 2010) therefore does not focus on a specific individual, but on the probability of any random disclosed record being re-identified (hence the ”marketer” name). The attacker evaluates the proportion of records that would be correctly re-identified. The marketer model thus does not identify which records are likely to be re-identified, unlike the journalist and prosecutor models.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

Therefore, it is important to consider how these attacker models differ in how success of an attack shall be evaluated. In the prosecutor and journalist model, the success evaluation is focused on whether or how much information of that individual can be inferred, e.g., if that individual is correctly identified in the database in case of a re-identification attack. For the marketer model, an attack can be considered successful if a larger fraction of the records could be re-identified. This applies similarly also for the other two types of disclosure mentioned above, i.e., attribute and membership disclosure. Literature discusses further details on how to evaluate the disclosure risks, e.g. (Banerjee et al., 2011) or specifically for the marketer model in (Dankar and El Emam, 2010).

4.3 Confidentiality Attacks on Machine Learning models

Releasing a trained model *might* cause unintentional leakage of information from the training dataset, similar to leakage from published data sets, as discussed above. This section discusses these threats in more detail than Deliverable D2.1 “Risk assessment methodology”, to provide app developers with a solid understanding of potential threats.

(Ateniese et al., 2015) are among the first to show that it is possible to extract some characteristics about the training set, which the performance of the classifier might depend on, such as the prevalent accents of voice samples used to train a speech recognition software, concluding a risk for intellectual property rights. A similar technique was used in (Ganju et al., 2018) to infer training set properties from fully connected neural networks. In particular, a meta-classifier was trained on proxy models with the same classification task as the target model, whereas the training sets were explicitly designed to have or not have the global or class-related target property. In (Carlini et al., 2019) generative sequence models, often used for text completion, were demonstrated to suffer from unintentional memorisation of secret sequences, like social security numbers, occurring even without overfitting the model to the training data, and thus disclosing information about an individual instance from the training set.

For many settings, it is important to consider the information available to the attacker, which is called *adversarial knowledge*. It can be divided into the following categories (Hu et al., 2021):

- **Data knowledge** denotes the information about the dataset and its distribution. An adversary often knows how the training set is distributed and, therefore, is able to acquire a dataset that holds data with a similar or the same data distribution (sometimes referred to as shadow data). In membership inference settings, it is often assumed that the original and shadow data are disjoint; however, they can also be joint, if the attacker has more information on the data set.
- **Training knowledge** denotes the knowledge about the learning algorithm, e.g., how the model was trained (the optimiser, number of iterations/epochs, and other hyperparameters). In many membership inference attack settings, it is assumed that the adversary has this knowledge.
- **Model knowledge** denotes knowledge about the architecture and parameters of the trained model, e.g., the type of neuronal network, the activation functions and number of layers, and the learned parameters themselves.
- **Output knowledge** is the knowledge of the predictions, e.g., the class probability vector in a multi-class setting.

In the following, we map literature on attacking the confidentiality of machine learning models to the threats and attacker models from above.

4.3.1 Identity Disclosure from ML Models

Identity disclosure is until now not specifically discussed, as the learned model parameters generally cannot be correlated directly to an individual, and thus e.g., no linkage is possible. The attack closest

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

corresponding to identity disclosure is likely the model inversion attack (Fredrikson et al., 2015); this attack re-creates training data from a learned model, and while that training data then in most cases might still not be easily identified, (i) the re-creation of the training data can be considered a precondition of identity disclosure, and (ii) the severity of the attack resembles identity disclosure. The idea of model inversion is that, since (in a supervised learning setting) a trained model stores a mapping between the input and output space, it can not only be used to infer predictions one way (i.e., from a given input sample to the output), but may also be inverted to yield information about the training samples themselves, from a specific output (e.g., a specific class label).
(Fredrikson et al., 2015) were among the first to show this attack. They invert decision trees trained on benchmark datasets and indicate that an adversary with a *white box* access to the model can predict a sensitive feature with perfect precision (i.e., no false positives). Furthermore, an adversary with no auxiliary information about the target class other than the label and a large feature space (an image with floating point pixel values) is considered, and logistic regression and (simple) neural networks are considered as models. The approach is based on exploiting the predictive power of the models by finding an optimal input so that the discrepancy between the predicted value and the target response is minimised. This can be achieved by the usual gradient descent method, computing the local value of a loss function and incrementally approaching the most "correct" input. However, it has to be taken into consideration that a machine learning classifier is intentionally trained to generalise relevant class-inherent features and therefore the optimised input will in most cases not represent a specific sample from the original training dataset, but rather a weighted average of the features with the highest *influence* on the classifier's decision. This characteristic of the model inversion attack explains why it is feasible in specific settings only (such as e.g., face recognition, where one class resembles on individual) but not others (such as gender classification, where a class represents all members of that gender), and why even in the case of face recognition, the results may rather resemble an unnatural caricature rather than a plausible photograph.
Other works ((Aïvodji et al., 2019) (Y. Zhang et al., 2020)) have therefore tried to constrain the reconstructed data, by using generative adversarial networks. In both studies a generative neural network was built and optimised in order to produce more realistically looking images. The results show that taking advantage of public data improves the readability by far, though it does not necessarily lead to images closer to the original training data.

Model inversion attacks in general pose **relatively few requirements on adversarial knowledge**. Primarily, output knowledge is essential, as well as model knowledge (to be able to e.g., perform the gradient descent for the input optimisation).

4.3.2 Attribute Disclosure from ML Models

Attribute disclosure could be seen as a **special case of model inversion**, i.e., the setting where we want to invert one (or more) attributes from an otherwise known input sample.
One of the first attacks was presented by (Fredrikson et al., 2014)[1], where the genetic markers of individual patients were recovered. More specifically, a pharmacogenetic linear regression model trained to predict Warfarin (anticoagulant medication) dosing for patients based on their clinical history, demographics and genotype was exploited to disclose sensitive training attributes, such as the named the genetic markers. This is done by a rather broad search and testing for all possible attribute value combinations, and eventually selecting those that produce the highest confidence (e.g., smallest distance of the predicted to the ground truth value in the actual regression task). (Fredrikson et al., 2014) also investigate the protective capacity of differential privacy against their attack, and show that it is not an effective countermeasure - for differential privacy to reduce the

[1] Coincidently, they call their approach in hat paper also a *model inversion* attack, though they have a different assumption than in their later paper (Fredrikson et al., 2015), as they try to infer only a small number of attributes, while in their later paper, they infer the complete feature vector.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

success rate of the attack substantially, the model would lose its predictive power to an extent unacceptable in a clinical environment.

Attribute disclosure attacks in general **require more adversarial knowledge than model inversion** attacks. Similarly, the demand output knowledge as well as model knowledge, but in addition, at least for the training data to disclose information on, partial training vectors are needed.

4.3.3 Membership Disclosure from ML Models

Membership disclosure from machine learning models has been extensively studied. The first work by (Shokri et al., 2017) focussed on a supervised attack. It assumes that similar models trained on similar data must behave in an alike manner. Thus, the attacker tries to create similar models (so-called shadow models) trained on data assumed to be similar to the original training data. For these models, the attacker knows which samples were in the training set, or not. Based on this information, and the prediction each of these samples returns, another model (the so-called attack model) is trained. The assumption is that the prediction (e.g., the vector containing the class-likelihoods in a multi-class classification task) will exhibit distinguished patterns whether they were used for training or not - an assumption that is based on the model being more confident on samples it had learned from, which is closely related to overfitting. Other attacks are unsupervised, i.e., they do not train an attack model. As such, (Yeom et al., 2018) e.g., base their decision on whether a sample was a member of the training set solely on the correctness of the prediction the model will perform on the original task (e.g., if an image showing a dog was actually classified as dog). (Salem et al., 2019) compare several measures computed from the prediction, e.g., the prediction loss compared to the average loss of all samples, and the confidence and entropy of the prediction compared to a designated threshold.

Membership inference attacks, broadly speaking, **require the most adversarial knowledge of the discussed attacks**. For supervised settings, all types of information are required, i.e., data, training and model knowledge to create the shadow models, and output knowledge to learn the attack model. (Salem et al., 2019) showed in their unsupervised attacks that membership inference is possible at a lower cost than was considered by (Shokri et al., 2017). They omit the assumptions about the adversary's knowledge of model architecture and data distribution and still preserve a high level of membership inference attack performance.

4.4 Guidelines for the FeatureCloud Project

For the FeatureCloud project, for each app, the following privacy and confidentiality risk and mitigation assessment steps should be carried out

- A **specific threat model** should be developed. This can, in most cases, be **based on and adapted from the above outlined types** of disclosure threats (identity, attribute and membership disclosure) and the attacker models that detail the level of background and target of the attacker (e.g., the prosecutor, journalist and marketer model). The threat and attacker models might need to be adapted for types of machine learning models that are not covered specifically by existing attacks.
- **Multiple scenarios** of the threat models with different adversarial knowledge and access should be explored and be evaluated based on their likelihood. Also, multiple different attacker models (marketer, prosecutor, or journalist) need to be considered. The potential impact of such attacks being successfully carried out should be estimated. Ideally, the most likely scenarios should also be **empirically tested**, in a form of security testing against the model.
- Based on these attacks, fitting mitigation mechanisms shall be chosen. These might e.g., be the below presented generic mechanisms for output privacy, such as objective or output differential privacy. It might however be necessary to additionally develop further,

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

application specific mitigation mechanisms addressing particular risks, e.g., further mechanisms of output perturbation besides differential privacy, or customised versions of objective perturbation, or suppression.

- In (Veale et al., 2018) potential threats imposed by model inversion and membership inference attacks are analysed with respect to the current European legislative framework (General Data Protection Regulation, GDPR), suggesting that releasing a trained machine learning model could be treated equivalently to publishing personal data. However, the degree of completeness up to which these attacks recover training data is still limited. Still, these considerations should be incorporated into the analysis of specific apps within FeatureCloud, and if the attack success rates seem likely enough, appropriate measures to fulfil legal requirements need to be considered.

5 Mitigation Mechanisms

In this section, we will describe the mitigation mechanisms that were chosen for the FeatureCloud framework to enhance confidentiality. Besides SMPC and Differential Privacy as generically available modules, we also discuss novel attack vectors like black-box and white-box exfiltration attacks in federated learning and discuss potential mitigations strategies against these attacks. Lastly, we discuss watermarking, as a reactive defence to protect the intellectual property a machine learning model constitutes.

5.1 Input Privacy via Secure Multiparty Computation

Secure Multiparty Computation (SMPC) (Evans et al., 2018) is a cryptographic protocol allowing to compute a public function over private data distributed across multiple parties (data owners), where no individual party can see the other parties' data. It is thus very well suited for settings where multiple participants want to collaboratively learn a function (e.g., simple functions like computing the mean value or maximum value, or a more complex function such as a predictive machine learning model) without disclosing their (private and sensitive) inputs to the other participants, but where it is in principle accepted that the common output is shared back to the participants. Thus, the main functionality of SMPC is to provide **input privacy**. It is thus well suited for the setting the FeatureCloud project deals with.

The clients in federated learning can jointly compute an aggregation function (a global model) over their private input data (local models) using an SMPC protocol. In this scenario, no party will be able to see other parties' local models. Therefore, there is no more need for a (trusted) central aggregator collecting all local models and computing the global model. Clients do not have to transfer their potentially sensitive data (models, or other parameters) to a third party. SMPC thus protects the confidentiality of the **input** to the model aggregation.

5.1.1 SMPC Protocols, Garbled Circuits and Secret Sharing

There are several cryptographic protocols for implementing SMPC. Yao's *Garbled Circuit* protocol is a solution to the two-party computation as a specific sub-problem limited to two parties. It runs in constant rounds and avoids costly latency associated with other protocols (Evans et al., 2018). Garbled Circuit protocol models the function as a boolean circuit, and then masks the inputs and outputs of each gate so that the party executing the function cannot infer any information about the inputs or intermediate values to the function (Snyder, 2014). (Goldreich et al., 1987) extended the idea of garbled circuits for more than two parties.

For more than two parties, SMPC protocols can also utilise methods based on **secret sharing** (Beimel and Chor, 1994). Secret sharing allows one to distribute a secret among several parties by distributing shares to each party (see Figure 4). From these shares, it is then possible to compute

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

the final function. The first secret sharing schemes have been published independently by (Blakley, 1979) and (Shamir, 1979). Both introduced a (t,n) - threshold secret sharing scheme, where the secret s is divided into n shares. The threshold $t \leq n$ defines the minimum number of shares that must be combined to derive the secret s. (t,n) - secret sharing schemes thus split the secret into n shares, such that any $t-1$ of the shares reveal no information about the secret, while any t shares allow complete reconstruction of the secret. Frequently, (n,n) - schemes are used, i.e., schemes where all n shares are necessary to reconstruct the secret (Evans et al., 2018). In some schemes an added encryption layer is implemented to ensure additional privacy and security, allowing the shares to be distributed amongst a network or group that are unknown to the secret owner.

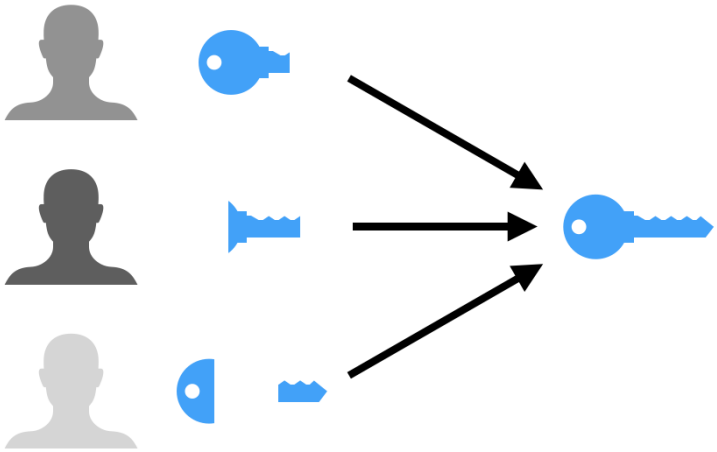


Figure 4: Secret sharing scheme. The secret is split and distributed among different parties.

Shamir's Secret Sharing scheme (Shamir, 1979) is a (t,n) - threshold secret sharing scheme, i.e., it splits the secret into *n* shares and requires a fraction *t* (the threshold) of those shares to reconstruct the original secret. Shamir's secret sharing allows a secret owner to add, amend or remove shares at any time if they wanted to, without modifying the original secret. Shamir's method for secret sharing relies on polynomial interpolation, which is an algebraic method of estimating unknown values in a gap between two known data points - without needing to know anything about what is on either side of those points. In this $(t,n)-$ *threshold scheme*, the secret $S \in \{S_1,\ldots,S_n\}$is divided into n shares by choosing random $t-1$ elements $a_0,\ldots,a_{t-1}$ with $a_0 = S$ and constructing a polynomial with degree $t-1$: $g(x) = a_0 + a_1 x + \ldots + a_{t-1}x^{t-1}$. Each of n participants receive one share of a secret S_i such that and $S_i = g(i), i = [1,\ldots,n]$. Given any t shares S_i of the secret S , one can find coefficients of a polynomial $g(x)$ by interpolation and retrieve the secret $S = g(0) = a_0$. Knowledge of $t-1$ shares or less is not sufficient to calculate the secret, and in fact, it does not give any information about the secret at all.

Blakley's Secret Sharing scheme (Blakley, 1979) is based on hyperplanes in t-dimensional spaces. As well as in Shamir's secret sharing, one needs to have at least t shares to compute the secret. Blakley's secret sharing scheme assumes that the secret is a point in t-dimensional space. If t hyperplanes will intersect at this point, one can reconstruct a secret key S. If one has $t-1$ or fewer hyperplanes, that enables to identify the line that passes the hyperplane, but no concrete information about the point S can be derived. In contrast to Shamir's scheme, the participants that cooperate in the reconstruction are not able to calculate the shares of the participants that were not part of the secret reconstruction. This is because the shares have no relation to each other, except that all shares pass through the same point S (Blakley, 1979).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

5.1.2 Threats against SMPC

Most SMPC protocols are resistant against a *semi-honest* adversary model (e.g., Yao's Garbled Circuit). The semi-honest adversary model (also called "*honest-but-curious"*) refers to **passive** adversaries who follow the protocol but can try to learn (or infer) sensitive information - beyond what they should learn from just following the protocol. A *malicious* adversary is an **active** adversary, which means a corrupted party may deviate from the protocol in an attempt to violate security. A Malicious adversary can try to control, manipulate or inject messages on the network. An example of SMPC protocol resistant against malicious adversary could be the Cut-and-Choose protocol - an extension on Garbled Circuit (Lindell et al., 2008). Semi-honest protocols can be elevated into the malicious model. However, that introduces a significant cost overhead which may not be acceptable in practice.

5.1.3 SMPC in Federated/Collaborative Learning

There are several approaches for using SMPC protocols for collaborative training of different machine learning models. (Gascón et al., 2017) utilised an SMPC protocol to securely compute a linear regression model on the data distributed among different clients. (Mohassel and Zhang, 2017) presented protocols for privacy preserving machine learning for linear regression, logistic regression and neural network training based on SMPC. (Bonawitz et al., 2017) designed an SMPC protocol that could be used in federated learning. The authors utilise secure aggregation to privately combine the parameters of local machine learning and compute a global model. With secure aggregation, the service provider will only see the update after it has been averaged with those of other users. They consider the task with mobile devices and focus on minimising communication while guaranteeing robustness to dropout, which is common to that setting. They present two protocols: one is more efficient and secure against honest-but-curious adversaries and another one guarantees privacy against active adversaries, but is less efficient. (Zhu et al., 2020) utilised secret sharing for a federated learning framework. Each client's private model updates are split into random shares among several independent servers, which compute models average using secret sharing. Their protocol is secure against honest-but-curious adversaries.

5.1.4 Shortcomings of SMPC

One of the main issues of SMPC is efficiency (Evans et al., 2018). Secret sharing involves communication between clients. SMPC has a computational and communication overhead compared to standard not-private execution, especially SMPC protocols with protection against malicious adversaries. Combining SMPC with federated learning makes it possible to offload the actual training to clients. The only function executed by SMPC is the aggregation of these locally trained models, instead of computing all steps via SMPC. This means that if the effectiveness of the federated learning approach is close enough to the effectiveness of a theoretical centralised approach, then also the federated learning via SMPC will be effective, but also achieves input privacy.

Another problem of SMPC could be the dropout or unavailability of the clients. Users may drop from the federated learning system due to various reasons (e.g., connectivity issues), therefore the design of the SMPC-based aggregation needs to be robust in the situation where the clients can drop out at any stage of the protocol execution (So et al., 2021). This is especially considered in settings with massively federated learning, and/or settings with relative instability in availability of the clients, e.g., in a setting of many clients being mobile phones. For FeatureCloud, it can be expected for many settings that the participation in the federation is stable, thus large numbers of drop out will not be a frequent setting. This is due to the generally different nature of collaboration, where fewer nodes collaborate, and studies are prepared to a great extent. However, the system should still

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

accommodate for dropout to potentially happen; thus, two principal approaches are feasible within FeatureCloud:

- The usage of an SMPC scheme and settings that does not require all shares to compute the final result. The above mentioned (t, n) - threshold secret sharing schemes can be utilised to this extent, where the parameter t indicates how many shares need to be received from the clients, or inversely, that $n - t$ clients are allowed to dropout.
- If more than $n - t$ clients drop-out (which also implies that, if $n = t$, if not all clients are transmitting their results), the aggregation cycle could be repeated. This introduces a potential computational overhead, as certain computation needs to be repeated. However, this only involves the step computing, exchanging and aggregating the shares, but the results from the local training can be reused. Thus, it is only required to repeat the SMPC process of aggregation from the locally computed updates to the model, which can be repeated with a subset of the original available clients, and does not require a complete recomputation.

While SMPC provides input privacy and allows protecting the privacy of intermediate results, it reveals the final result - the output of the function. In federated learning, setting the output can be also potentially sensitive and vulnerable to inference attacks. Another technique needs to be applied to ensure output privacy as a complement to SMPC, e.g., Differential Privacy.

5.1.5 Alternatives to SMPC

Another frequently discussed technology to achieve privacy-preserving computation, besides Secure Multiparty Computation, is Homomorphic Encryption (HE) (Rivest et al., 1978). While encryption schemes, in general, try to only protect the confidentiality of data, homomorphic encryption allows, in addition, performing mathematical operations on the ciphertexts. When decoding the result, this is equivalent to having performed the operations on the plaintext. Fully homomorphic schemes (Gentry, 2009) allow arbitrary computation. Partially or somewhat homomorphic schemes allow only a certain subset of operations but provide increased efficiency. Newer schemes include e.g., CKKS (Benhamouda et al., 2017). This would in principle allow envisioning an architecture where the coordinator receives all local models in the ciphertext of a homomorphic encryption scheme, and computes (averages) the global model still in ciphertext, before sharing it back to the clients, which can then decrypt the global model for further use.
However, even with recent improvements, HE is in general less efficient than SMPC. State-of-the-art HE implementations are thousands of times slower than SMPC in typical application (Evans et al., 2018). Another aspect that would need to be solved when employing HE in a federated setting is key sharing, which would require a secure scheme for key exchange. If all clients use the same encryption key, leakage of such a key would allow deciphering all local models. Thus, SMPC seems more promising to utilise within the FeatureCloud architecture.

5.1.6 Summary and Integration in FeatureCloud

Secure Multi-party Computation (SMPC) allows to compute a common function, such as the model aggregation by e.g., averaging, without having to share the inputs (i.e., the local models) in plain-text. It thus is an ideal complement to other approaches enabling confidentiality of inputs during the communication, such as encryption. It further eliminates the need for a (trusted) party, the aggregator, to receive all inputs in clear text. The overhead from the cryptographic protocol is reduced within the setting of federated learning, where not all of the model computation, but just the aggregation needs to be secured through SMPC.
Within FeatureCloud, the platform already implements one scheme based on secret sharing, implemented as part of work package 7, in collaboration with this work package. It is available via the FeatureCloud API; for more details, please refer to. Section 4.3 in Deliverable “D7.3 Federated machine learning apps running in app store”. The SMPC schemes need to be extended to provide

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

functionality for further forms of computation, besides the widely used summation and averaging building thereupon, to accommodate for future apps that require such computation.

5.2 Output Privacy via Differential Privacy

While Secure Multiparty Computation, as described above, protects the confidentiality of the inputs to the model aggregation functions employed in FeatureCloud, it does not protect the output, i.e., the merged global model from attacks on its confidentiality. Thus, this needs to be addressed additionally. While protection of input data against inference from a model learned therefore is in principle very specific, several approaches have highlighted the potential of differential privacy as a mitigation mechanism. We therefore provide differential privacy as a generic app that can be used in various analysis workflows within FeatureCloud.

5.2.1 Differential Privacy

Differential Privacy (DP) is a mechanism that provides privacy by process (Dwork, 2006). It allows hiding individuals' information when publishing information about a dataset (statistics or models). Differential privacy tries to solve the paradox of learning nothing about an individual while learning useful information about a population. DP protects against individuals' identification and neutralises linkage attacks (Dwork and Roth, 2014). DP provides an opportunity to quantify privacy loss (privacy budget) and set a parameter of how private an algorithm should be. DP applied to machine learning allows publishing model parameters with a guarantee that adversaries are severely limited in the information they can learn about the original training data based on analysing the parameters. Therefore, DP can be used to ensure output privacy in federated learning.

An algorithm analysing a dataset is considered differentially private if by looking at its output, one cannot infer if some particular record was included in the dataset or not. In other words, it ensures that any sequence of outputs (responses to queries) is equally likely to occur, independent of the presence or absence of any individual (Dwork and Roth, 2014).

The formal definition of differentially private mechanism was given by (Dwork, 2008):

A randomised function K *gives* $\varepsilon - differential\ privacy$ *if for all datasets* $D1$ *and* $D2$ *differing on at most one element, and all* $S \subseteq Range(K)$,

$$Pr[K(D1 \in S)] \leq exp(\varepsilon) * Pr[K(D2) \in S]$$

(Dwork, 2008)

Epsilon (ε) is a **privacy budget,** a metric of privacy loss denoting the maximum distance between a query on one database and the same query on another database. The smaller ε, the better privacy, but the less accurate the response. DP has limitations in settings where the same or similar analysis needs to be repeated several times, as the privacy budget may be used up. By combining data from multiple trials, adversaries may be able to infer sensitive information about the individuals, as the noises added to each result will eventually cancel each other out, and a value close to the true value will be returned.

Delta (δ) denotes the probability of information accidentally being leaked. $(\varepsilon, \delta) - differential\ privacy$ gives the possibility to relax the original concept of DP against unlikely events. An unlikely event takes place if $Pr[K(D) \in S] \leq \delta$ is valid. When δ is negligible or equal to zero: It is called $\varepsilon - differential\ privacy$ or $(\varepsilon, 0) - differential\ privacy$. When δ is set too high, too many events are unlikely and the privacy guarantee of ε can be misleading.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

A randomised function Kgives $(\varepsilon, \delta) - differential\ privacy$if for all datasets $D1$ and $D2$ differing on at most one element, and all $S \subseteq Range(K)$,
$Pr[K(D1 \in S)] \leq exp(\varepsilon) * Pr[K(D2) \in S] + \delta$

(Dwork et al., 2006a)

In general, DP can be achieved by adding noise. In order to randomise a function, noise can be added anywhere, e.g., in machine learning, the input can be distorted, noise could be added to the output or during the learning process. Epsilon (ε) defines how much noise has to be added.

The Laplace and Gaussian mechanisms are among the most popular approaches to achieve DP. They are based on the idea of adding noise to the output of a function $f: D \rightarrow R^k$. In the Laplace mechanism, for example, any function $f: D \rightarrow R^k$ fulfils $\varepsilon - differential\ privacy$ if noise z is added according to the Laplace distribution to the function f, with the parameter $Sen\ (f)_1 | \varepsilon$:

For a function $f: D \rightarrow R^k$, the mechanism that returns $f(D) + z$, where each $z_i \in z$ is drawn from $Lap(Sen\ (f)_1 | \varepsilon)$ satisfies ε-differential privacy.

(Dwork and Roth, 2014)

$Sen\ (f)_1$is a sensitivity function, defining the amount of noise together with an epsilon (ε). Generally, sensitivity refers to the impact of a change in the underlying data set on the result of the query. In other words, the sensitivity describes how much a single element can change the output of the function. *Global sensitivity* refers to the consideration of all possible data sets differing in at most one element. *Local sensitivity* is the change in one data set with differing at most one element.

Given a function $f: D \rightarrow R^k$, its sensitivity is:
$Sen\ (f)_1 = max_{D1 \sim D2} || f(D1) - f(D2) ||_1$
where $|| \cdot ||_1$ is the L_1norm; $D1 \sim D2$ means that the Database differs in only one tuple.

(Dwork and Roth, 2014)

Privacy comes at a cost: DP can result in a significant reduction of the accuracy of the model ruining its prediction capacity. One should acknowledge the *tradeoff between privacy and utility* of the model. Several works suggest how to calibrate the noise in differentially private algorithms to preserve the utility and keep sufficient privacy level (Dandekar et al., 2021) (Dwork et al., 2006b).

5.2.2 Differential Privacy for Machine Learning

DP is widely considered as a solution to privacy-preserving machine learning, as it can provide a provable privacy guarantee for individuals (Gong et al., 2020). One can utilize DP when training different machine learning algorithms, e.g. Naive Bayes (Vaidya et al., 2013), or Decision Tree (Jagannathan et al., 2009) (X. Liu et al., 2018). In such cases, one usually combines machine learning algorithms with Laplace, Gaussian or exponential mechanisms to achieve DP. Another approach to achieve DP is output or objective perturbation. The output perturbation mecha-nism can be implemented by adding noise to the model output. Objective perturbation is performed by adding noise to the objective function and optimizing the perturbed objective function. (Zhang et al., 2012) show an approach to achieving DP in linear and logistic regression with perturbing objective function by adding noise to its coefficients. (Chaudhuri et al., 2011) demonstrate how logistic regression and SVM can be combined with DP by applying output perturbation.

To train a differentially private machine learning model one also can utilise a differentially private stochastic gradient descent (SGD) algorithm (Abadi et al., 2016). This algorithm uses DP by adding

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

noise to the gradient at each iteration of SGD. The Adaptive Laplace Mechanism (Phan et al., 2017) can be used to train differentially private artificial neural networks. The work of (McMahan et al., 2017) shows that achieving differential privacy comes at the cost of increased computation, rather than decreased utility when there is enough data. They introduce a "noised version" of the federated averaging algorithm based on DP SGD (Abadi et al., 2016). They demonstrate that differentially private Long-Short-Term-Memory (LSTM) language models are quantitatively and qualitatively similar to models without DP when they are trained on a large dataset.

5.2.3 Differential Privacy for Federated Learning

DP was also considered in federated learning settings in several works. (Shokri and Shmatikov, 2015) suggest using distributed selective SGD in collaborative learning. In selective SGD each client chooses a fraction of parameters to be updated and shared at each iteration. They use DP to randomly select a subset of gradients whose values are above a threshold, and share perturbed values of selected gradients. (Zhang et al., 2017) propose a method for privacy preserving multi-party deep learning by applying differentially private randomization to local gradients. To improve the utility of the global model, they suggest ensuring that participating parties learn the global model only if a sufficient number of local models are aggregated. (Geyer et al., 2017; Zhang et al., 2017) suggest an algorithm allowing to hide clients' contributions during training. They focus on protecting a single data point but rather ensuring that a learned model does not reveal whether a client participated in federated learning. They show that, with a sufficiently large number of participating clients, a client's participation can be hidden while model performance is kept high in federated learning. (Wei et al., 2020) study differential privacy application in the federated learning setting. They suggest adding noise to the parameters at the client's side before aggregating to ensure the privacy of the client's data. They analyse convergence properties of federated learning with differentially private updates. They suggest a scheduling strategy for the updates finding the optimal number of clients needed to participate at each aggregation step.

There are works suggesting frameworks which combine SMPC and DP for privacy preserving federated learning. (Chase et al., 2017) design a framework for privacy preserving collaborative learning of neural networks using DP and SMPC. In order to avoid dramatic degradation of global model accuracy, they suggest avoiding adding noise to each client's updates. For that they utilise SMPC protocol to aggregate local updates and only then introduce random noise to the global model. (Truex et al., 2019a) also presented a hybrid approach for federated learning combining SMPC and DP. The clients use a DP mechanism depending on a machine learning algorithm to add noise according to the privacy budget allocated at each step. Then noisy parameters are encrypted with homomorphic encryption and sent to a central aggregator. Due to homomorphic encryption, the aggregation of encrypted models happens and the aggregator queries a number of clients to decrypt the aggregate value. Due to adding DP, the system prevents privacy leakages even if parties actively collude.

5.2.4 Summary and Integration in FeatureCloud

Differential privacy allows preserving the global models obtained in a collaborative learning setting, and thus complements input confidentiality mechanisms. As making a specific algorithm (process) differentially private is heavily dependent on the nature of the algorithm, and parameters such as the sensitivity need to be derived specifically for each algorithm, it is difficult to generically make all possible FeatureCloud apps differentially private with the same flavour of a mitigation mechanism. Thus, we will pursue the following approach:

- For commonly used algorithms, key elements of making the learning algorithm functionally differentially private will be provided. This means providing templates and mechanisms e.g. for differentially private optimising techniques such as stochastic gradient descent (Abadi et al., 2016), which can then be used in a multitude of learning algorithms (e.g. linear and logistic

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

regression, decision tree, support vector machine, or neural networks), and other frequently used estimation approaches (including expectation maximisation (Park et al., 2017)). Where applicable, these are provided as "compute modules" within the FeatureCloud system (cf. Section 5.3 in Deliverable "D7.3 Federated machine learning apps running in app store")

- Modules providing output differential privacy, e.g., on the learned model parameters, will be available for learning algorithms that cannot easily utilise the above mentioned functional (objective) differential privacy mechanisms. Instead, these will be available as a FeatureCloud app, to be applied on the output of the (local) app, i.e., on the learned parameters. The mechanisms provide include e.g., Laplacian noise adding mechanisms, which was shown to be asymptotically optimal in a high privacy regime (Geng and Viswanath, 2016). The app developer will be responsible for identifying correct parameters (such as the desired ε, etc) when integrating the output differential privacy app.

5.3 Defending against Data Exfiltration

The FeatureCloud architecture ensures that the training data can reside in the original location (e.g., at the hospital gathering the patient data), as with federated learning, only intermediate models as a derivative form of this (local) data are exchanged. However, parts of the FeatureCloud platform, specifically the algorithms training on the participants' local infrastructure, albeit executed in a secured environment, do still directly access data in their original, unabridged (i.e., not anonymised) form. Exfiltration of this data by components of the FeatureCloud platform is thus a risk to consider. **Data exfiltration** is an unauthorised movement (transfer) of the data or data theft. Data exfiltration attacks are usually referred to as attacks targeting web traffic, e.g. exploiting various network protocols (Van Antwerp, 2011), or vulnerabilities in Internet-of-Things devices (D'Orazio et al., 2017). Traditional attacks thus try to utilise a dedicated channel through which the data is exfiltrated, e.g., a connection to an external server where the data will be stored (through whichever protocol, e.g., SSH or FTP). However, in our mitigation strategy, also novel channels will be considered and mitigated against.

While it can generally be assumed that the apps provided by the FeatureCloud consortium itself are trustworthy, the openness of the platform, where new apps can be developed by third parties and made available in the app store and subsequently be included in the analysis process, opens up potential attack vectors for "malware" apps that could try to exfiltrate the data. The FeatureCloud project will address this in several manners.

5.3.1 Traditional Exfiltration Attacks

On the one hand, the certification of apps that is envisioned for the FeatureCloud app store will be able to spot some of the malicious apps. However, the certification process might fail to spot all vulnerabilities. On the other hand, the KPIs identified in Deliverable D2.2 "KPIs and metrics for local execution platforms", address this risk from a rather grey-box to black-box approach, in a way that is agnostic of the actual algorithm and implementation. As such, KPI3 "Privacy Requirements for Federated Algorithms Working on Patient Data" analyses the amount of exchanged data and can thus indicate anomalies, e.g., mass-exfiltration attacks. Further, local systems are encouraged to be tightened so that connections to external services are limited resp. well-controlled, and thus traditional exfiltration channels are not available.
In this regard, many types of data exfiltration attacks can be mitigated by traditional security mechanisms, e.g., sandboxing (create an isolated environment), intrusion detection, regular monitoring network services to ensure that only known acceptable services are running at any given time, and firewalls blocking unauthorised access to resources and systems storing sensitive information.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

5.3.2 Machine-Learning based Exfiltration Attacks

Data exfiltration can be triggered by malicious third-party algorithms which are executed on (resp. have access to) the sensitive data in an otherwise secure environment. In the case of machine learning, a data owner might want to delegate model creation to a third party, or to use a third-party library, within the data owner's environment. The attacker's goal of such a data exfiltration attack is to extract information about the training data, using the trained machine learning model as a channel, as shown in (Song et al., 2017).

We consider a scenario when an adversary has a direct influence on the training process but do not have access to the private data. The client wants to use a training algorithm as a service on sensitive data and keep the data private. The adversary modifies the algorithm to be malicious in order to perform memorization of the training data. During the training process, the malicious algorithm simultaneously trains a machine learning model whilst also encoding training data information into the model parameters. Thus, information hiding (steganography) methods are applied to hide the data to be exfiltrated in the machine learning model. Later, the adversary is able to exfiltrate private training data using this machine learning model. Steganography has been used for various use cases (Morkel et al., 2005), for example, as a method for communication of concealed information. Information can be hidden in various media types such as text (Por and Delina, 2008) or image data (Morkel et al., 2005) (Younes et al., 2008). Statistical models have been used to conceal secret data as well (Sallee, 2003), which is related to ML-based data exfiltration.

5.3.2.1 ML-based Exfiltration in Federated Learning

Applied to the setting of FeatureCloud, we consider a novel exfiltration attack that would utilise information hiding approaches to exfiltrate data among an existing, legitimate channel. Such a channel in a federated learning setting is the transmission of the learned (local) models from the participants to the global model.

For extracting the data from the model in the subsequent stage, multiple scenarios are possible. The data encoding depends on the type of access the adversary has to the model once it is trained, and we can consider the following scenarios:

- **White-box access**, i.e., full access to the model, including the learned model parameters. In this setting, the adversary can encode the training data directly into the machine learning model parameters by means of their malicious algorithm (Song et al., 2017). That can be done e.g. (i) by replacing the *n* least significant bits (LSB) of the parameters with the binary secret vector values, by correlating the parameters to the values of the secret vector, or (ii) by matching the signs of the parameters to the secret vector. The adversary then decodes the information hidden in the learned parameters, and reconstructs the training data or some parts of it. These techniques are closely related to steganography - a technique of concealing information within another object, e.g., text or image (Morkel et al., 2005). Depending on the encoding information, the attacker can embed one bit per parameter for the sign encoding approach, and *n* bits per parameter in the LSB encoding scheme.

- **Black-box access,** i.e., only input-output (query-response) access to the model. In this scenario, the attacker only obtains the predictions from the model, but can arbitrarily choose the queries sent to the model. Instead of encoding the data to be exfiltrated directly into the parameters of the model, the attacker can only encode it into the responses of the model. One approach to utilise is thus by letting the model encode this information into specifically prepared queries. To this end, during training, the attacker synthesises these queries (data points), and each of these data points are labelled with (a part of) the secret. Once the adversary wants to gain access to the secret (i.e., the confidential training data), they

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

synthesise the same data points and return the learned prediction containing the information. The attacker can thus encode bits depending on the number of query-response pairs, depending on the type of output available. In case of a binary classification problem, and only receiving the final label, this allows one bit of encoding; a multi-class problem with just the label predicted allows the square root of the number of classes to be encoded per query-response pair. If besides the label also the complete prediction vector (with e.g. the class probabilities) is available, more information can be encoded.
The black-box access attack is to some extent similar to a data poisoning (backdoor) attack (Gu et al., 2017), as the attacker inserts specifically crafter images into the training set, with a desired specific outcome label (the bit encoding).

In federated learning, the threat of data exfiltration can come from a malicious third-party algorithm used by the participants. The participants might use malicious algorithms to train a machine learning model on the private data. This algorithm encodes the training data into the local models. An adversary having access to the model can decode the information from the model and eventually access a participant's private data. We can consider in principle two further access types:

- Access to the local models (e.g., by a malicious central aggregator, or by other means of accessing a local model e.g., by eavesdropping on the transfer), which renders this setting the same as in a centralised learning (Song et al., 2017), and
- Access to the global model, i.e., after aggregation, which also includes in principle every legitimate user of the model.

The latter setting, i.e., data exfiltration via a global model in a federated setting, is still novel, and the scope of the threat and specific circumstances are investigated by the FeatureCloud project. Naive approaches are expected to fail, at least when the model that contains the embedded data represents just a small fraction of the overall federation, as aggregation of the local models will likely overwrite the embedded data. Methods that are also used in data poisoning (backdoor) attacks against federated learning, where specific clients "boost" their model updates so that they persist after the aggregation (Bagdasaryan et al., 2020) might however improve an attacker's chance. Further, related work so far has only considered image and textual data, but not tabular (structured) data.

5.3.2.2 Defences against ML based Data Exfiltration

As a first aspect to fully understand the potential threat, the capacity and accuracy of data exfiltration attacks utilising the learned model as a channel will be assessed. The effects that this encoding has on the performance (effectiveness) of the original model will be assessed as well. The attacker's goal is to keep that performance as close as possible to an untampered model, as a too large decrease in performance might be an indicator for malicious behaviour. It is expected that utilising higher capacity (e.g., using more or even all parameters to encode data in the white-box setting, utilising more than just one LSB, and more query-response pairs) will have an adverse effect on the performance of the model, and thus be easier to detect.

As mitigation mechanism against this specific form of data exfiltration via learned models, the FeatureCloud platform will provide generic methods to post-process models before they are further made available. The white-box attack using LSB encoding can be defended against by perturbing (or deleting) a number of LSBs. Against sign encoding, flipping the sign of some parameters might sufficiently remove the information from the parameters. For black-box based attacks, due to their nature, defences similar to those frequently employed to spot backdoored models might be viable (Tran et al., 2018) (K. Liu et al., 2018) (B. Chen et al., 2018). These defences can be applied either on the local or the global model - or on both. The expected impact of the defences on the

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

performance of the models needs to be investigated, to allow an informed decision on the desired strength of the defence.

Data Exfiltration via learned models can be called to be based on intentional memorization. It is important to differentiate between intentional memorization and **unintended memorization**. The latter causes models to retain certain characteristic training data without specific intervention of an attacker. E.g., (Carlini et al., 2018) claim that generative sequence models including language models tend to memorise infrequent or unique sequences in the training data. The attacker in this case does not have any influence on the algorithm, but tries to extract data that was memorised by chance. An important point to consider is the relation to overtraining (Yeom et al., 2018). (Thakkar et al., 2020) further suggest that federated learning reduces unintended memorization through user-based data clustering and through the use of the averaging algorithms. It has to be investigated whether similar observations also hold true for intentional memorization.

5.3.3 Summary and Integration in FeatureCloud

Defending against data exfiltration attacks in a setting as complex as FeatureCloud is not trivial. Adversaries focus not only on exfiltrating private information, but also on concealing this process. Thus, a multitude of defences needs to be available. To summarise, the following mitigation mechanisms will be deployed:

- Code inspection of apps from the app store, which will allow discovery of (some of) the apps that use malicious code used for exfiltration
- Hardening of local machine setup (restricting external connections, ….) according to best-practises
- Monitoring for anomalous behaviour, e.g., the KPIs discussed in Deliverable D2.2
- Defending against utilising the learned models as exfiltration channel, e.g., modifying least significant bits before sharing models' parameters (Song et al., 2017).

5.4 Model IP protection by watermarking & fingerprinting

Besides attacks on data confidentiality, also the learned models themselves might be targets of attacks, as they may represent a significant value, given that the resources (input training data, compute resources, domain and expert knowledge, …) required to obtain these models are themselves rare and valuable. Thus, the learned models constitute intellectual property, and parties who create and train the model might want it to be protected from misuse.

5.4.1 Threat Model

To define a threat model, we first need to understand the motives of an attacker (or adversary or malicious user). The model owner, i.e., the person that invested resources to obtain a ML model for a specific task, wants to offer the model to some target audience for use.

Attackers might want to obtain these models from legitimate sources, to use them for their own purposes, or offer services based on the models to third parties, e.g., for monetary reasons. The most prominent reasons for an attacker to re-distribute a model would be (i) no (or not enough) training data, expertise, time or computational power to train such a model themselves, and/or (ii) the unwillingness to agree with the licence terms of the obtained model or the fees for using Machine Learning as a Service (MLaaS).

As a threat model, we consider the following situations:

- **Legal copy**: The model owner distributes the model publicly, either for free, but with a restrictive licence, or for a fee. The attacker then obtains this model, and re-distributes it via a lucrative API service.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

- **Illegal copy**: The model owner distributes the model as a pay-per-query API service. The attacker then performs a Model Extraction Attack and provides his own lucrative API service.

Regardless of how the attacker obtained the model, in both cases, the IP of the model owner is illegally utilised. Later on, for both cases, we will discuss methods to protect the IP of the model owner. It is important to differentiate between those two cases, as this has a large impact for selecting potential defence mechanisms.

5.4.2 Watermarking, Fingerprinting and other information hiding techniques

One option to deter misuse is the reactive defence mechanism of model watermarking and fingerprinting. If an adversary uses an unauthorised model, the owner of this model can prove the ownership of the model by watermark.

Digital watermarking is a procedure of embedding a piece of signature in the data, e.g. multimedia intellectual property (Kahng et al., 1998) or relational database (Kamran and Farooq, 2018), to deter malicious usage and claim ownership of intellectual property. While also perceptible watermarks exist, often to make illicit use undesirable, such as superimposing logos or copyright notices in stock photo images, or video, often this signature is intended to be unnoticeable. The watermark should further be robust, i.e., it should be embedded in such a way that no algorithm can remove or overwrite the watermark, at least not without a significant cost. More recent digital watermarking techniques, e.g. for images, make use of deep learning techniques in the embedding process (Zhong et al., 2020); similarly, attacks targeted to remove such watermarks are increasingly using deep learning (Sharma and Chandrasekaran, 2020).

We consider fingerprinting as an extension of watermarking. While watermarking has the purpose to verify the *owner* of a digital resource, fingerprinting wants to trace back to the *(malicious) recipient* of the resource. Therefore, fingerprinting techniques should be capable of embedding multiple, but unique, marks to identify the recipient. Similar to watermarking, fingerprinting is already widely introduced in multimedia areas like images, audio and video (Lach et al., 1998), relational databases (Yingjiu Li et al., 2005), and other digital data types.

Digital watermarking and fingerprinting are a form of steganography or information hiding, i.e., the practice of concealing a message within another message, but we want to point out that there are several other connotations of watermarking, and generally information hiding, along the machine learning process, as depicted in Figure 4.

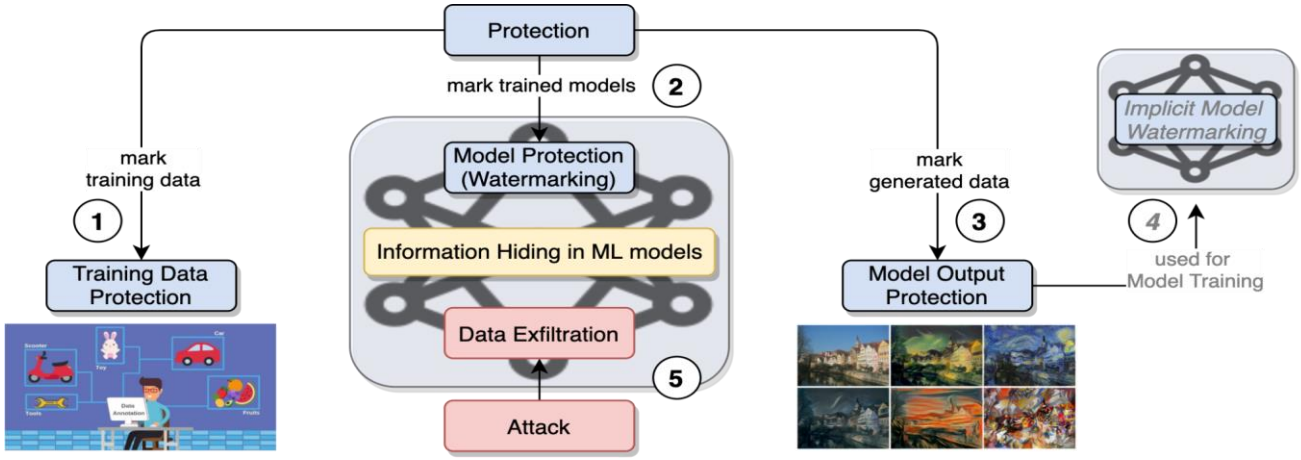


Figure 5: Different notions of information hiding along a model learning process

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

Sablayrolles et al. (Sablayrolles et al., 2020) propose a technique that *traces data usage*; it marks (training) data in a special way, so that a model trained on that data will bear a watermark that can be identified (cf. (1) in Figure 5). Watermarking instead considers models as the objects that need protection and in which the watermark is embedded (cf. (2) in Figure 5).
(Abdelnabi and Fritz, 2020) propose a special form of watermarking. Their scheme is not watermarking a *model*, but the *output* of a (text) generating model (cf. (3) in Figure 5)). They assume that an attacker could use the model for generating whole articles. In such a case, the watermark can be extracted from the generated text and prove illegitimate usage of the model.
In some settings, it is further considered that a marked output (prediction, or data) is generated with the explicit goal to trace usage of this data, e.g., for training by an attacker (cf. (4) in Figure 5). This is a special form of (1), as the data origin is different, and of (2), as the attacker's model is *implicitly marked*.
Model-based data exfiltration methods discussed in the previous section are also based on information hiding. The difference to watermarking however is in the purpose: for data exfiltration, information hiding conceals the malicious activity; for watermarking, the hidden information allows for ownership verification, and thus deter malicious activity (cf. (5) in Figure 5).

Almost all works on watermarking have focused on neural networks, especially on deep learning approaches such as Convolutional Neural Networks (CNN) or Recurrent Neural Networks (RNN). This can be explained by the fact that these generally are among the most complex models to learn, and thus require the most data, compute and other resources to be effectively trained - they thus constitute the most valuable models.

5.4.3 Attack model

Let us assume that the attacker obtains a legal copy of the target model, and either knows or assumes that the model has a watermark in place. We consider the following cases.

- **Watermark detection**: The attacker wants to detect the watermark in order to check for the existence of a watermark, in order to perform a targeted watermark removal or overwriting. If the watermark is not secured with additional mechanisms (e.g., a private key for extraction), the attacker could also claim ownership.
- **Watermark overwriting**: The attacker wants to overwrite the existing watermark by placing an own watermark and making the model owner's watermark useless.
- **Watermark invalidation**: The attacker wants to disable the watermark function, without actually removing it from the model, so that it cannot be verified.
- **Watermark removal**: The attacker wants to modify the model in a way such that the model owner's watermark extraction algorithm will no longer result in proving correct ownership.

Most of these attacks are also valid against *fingerprinting*.

5.4.4 Evaluating of Watermarking & Fingerprinting

A watermarking scheme should fulfil a couple of requirements. Literature is not coherent in the naming of these requirements of watermarking (and fingerprinting) algorithms, and we therefore aim at providing a common nomenclature. To this end, we collect all the requirements that were proposed in the papers included in our literature review and list them in Table 1, identifying also terms used as synonyms in literature.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

Property	Description	Other terms used in literature
Effectiveness	The model owner should be able to prove ownership anytime and multiple times if needed	Authentication, Functionality
Fidelity	The accuracy of the model should not be degraded after embedding the watermark	Functionality-preserving, Loyalty, Utility[2]
Robustness	The embedded watermark should resist a designated class of transformations	Unremovability
Security	The watermark should be secure against brute-force or specifically crafted evasion attacks	Secrecy, Unforgeability
Legality	An adversary cannot produce a watermark for a model that was already watermarked by the model owner	Ownership piracy resilient, non-ownership piracy
Integrity	The watermark verification process should have a negligible false positive rate	Low false positive rate, non-trivial ownership, Uniqueness
Reliability	The watermark verification process should have a negligible false negative rate	Credibility
Efficiency	The watermarking embedding and verification process should be fast	
Capacity	The watermarking scheme should be capable of embedding a large amount of information	Payload

Table 1: Requirements for Watermarking techniques. The notation is not consistent throughout the papers, but the terms in the left column are the most prominent ones. These requirements mostly apply also to Fingerprinting methods

The most important requirements are:

- **Effectiveness**: the watermark shall be embedded in a way that the model owner can prove ownership anytime
- **Fidelity**: the model's accuracy shall not be degraded because of the watermark embedding, and
- **Robustness**: the watermark embedding should be robust against several kinds of attacks, including fine-tuning, model compression and other, specifically crafted attacks.

Note that non-trivial ownership is sometimes used as a synonym for integrity, meaning that innocent models are not being accused of ownership piracy, but also as a requirement that an attacker cannot easily claim ownership without knowing the watermarking scheme and embedded watermark. Moreover, authentication is more a subset of effectiveness than a real synonym since it only requires that there is a provable association between an owner and their watermark. **Feasibility** is used as a combination of robustness and effectiveness (Li et al., 2019), and **correctness** as a combination of effectiveness, reliability, and integrity (Lukas et al., 2020). Fingerprinting should fulfil two more requirements: **uniqueness** - the fingerprint can be uniquely identified with the user, and **scalability** - the fingerprinting scheme should be able to embed multiple fingerprints.

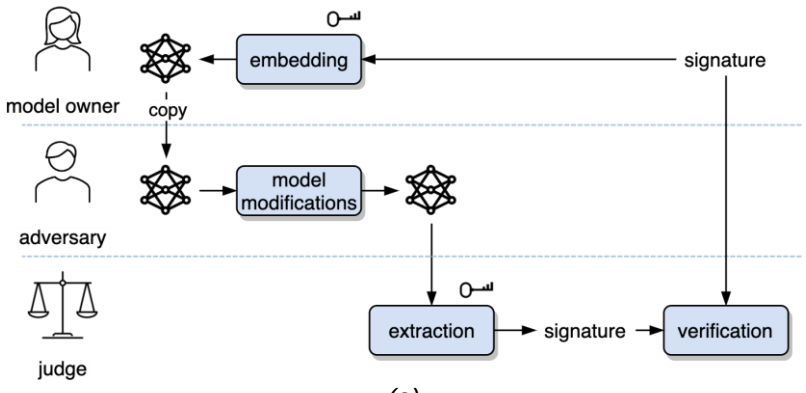
[2]In Image watermarking: Transparency. In Relational Data watermarking: Usability

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

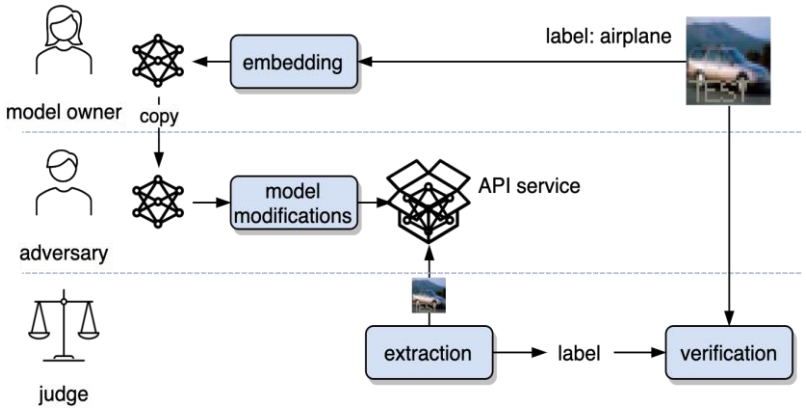
We note that all currently known schemes fulfil the above-identified most important requirements of fidelity, effectiveness and robustness, except for (Guan et al., 2020), which on purpose gives up on robustness in favour of reversibility: the authors point out that the application of their scheme is not IPP, but integrity authentication, and that all existing watermarking methods are irreversible - once the watermark is embedded, it cannot be removed to restore the original model without degrading the model's performance.
They argue that irreversible watermarking schemes alter the signature of a model, which could have severe consequences especially in applications for e.g., the medical or defence domain. To make the scheme reversible they sacrifice on the robustness requirement, inspired by traditional image integrity. For the method of (J. Zhang et al., 2020), the fidelity requirement does not apply, since it is not well-defined for image processing. To determine for a model that outputs an image (or other complex data) whether a watermarked version of such a model is comparable to the original one, one would need to define a similarity measure to compare if the two outputs are equivalent.

5.4.5 Watermarking process

The process of watermarking starts with **watermark embedding,** when the watermark is placed into the digital object, e.g., the machine learning model. This can be performed, e.g., via fine-tuning. Then follows watermark extraction to identify if any, and which watermark is placed.
Watermark extraction is the process in which the embedded watermark is extracted from the model, but neither in a permanent (which is called watermark removal) nor in a malicious way (which is called watermark detection).
Finally, during **watermark verification**, the extracted watermark is compared to the model owner's watermark in order to prove ownership. Following certain rules (e.g., thresholding the watermark accuracy or (bit) error rate), it is then decided if the watermarks are the same.



(a)



(b)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

Figure 6: Typical workflows for (a) white-box watermarking and (b) black-box watermarking

Watermarking methods vary depending on the model access, as shown in Figure 6:

- **White-box** approaches embed the watermark in the model parameters or other model characteristics. White-box means that the model owner needs access to the adversary model parameters or other model characteristics, in either step of the IPP method process, i.e., also during watermark extraction and verification. With that in mind, the model owner would need to get the stolen copy from the attacker for the watermark extraction process. This scenario seems unrealistic in most settings (e.g., an attacker offering an API service based on the model never discloses the model itself).
- **Black-box** mechanisms generally only need access to the model's prediction, e.g., via an API service, to observe matching input and output from the model, using it in a similar fashion as an *oracle*. Black-box watermarking tends to be more popular, as the model owner can verify ownership with as little as a set of trigger inputs and the corresponding responses of the adversary model. Most black-box models rely on a trigger set, which contains selected inputs and the expected response pairs.

5.4.6 White-Box Model Watermarking

The first framework for embedding a watermark into a DNN was proposed by (Uchida et al., 2017). They follow the idea of embedding a signature into the model, particularly in the DNN's weights. While it would be possible to directly alter the model's parameters, as it would be done for watermarking relational data, this would degrade the model's performance. Thus, the model is trained with a regularized term, which balances learning an effective model and embedding the signature.

(Rouhani et al., 2019) proposed a watermarking framework which proves to be more robust against watermark removal, model modifications and watermark overwriting. The method is as well regularized based, but encodes the signature in the probability density function (PDF) of activation maps obtained at different DNN layers, by one additional regularisation term that ensures that selected activations are isolated from other activations, to avoid creating a detectable pattern of alterations. The authors say that the scheme is both white-box and black-box, depending on which layers are available for watermark extraction.

(Wang and Kerschbaum, 2020) generalised both of the above presented algorithms into a white-box scheme. They show that the previous schemes are vulnerable to watermark detection, as the weight distribution deviated from those of non-watermarked models. The authors claim that this arises from the additive regularisation loss function(s). Therefore, they propose a new scheme that is particularly robust against detection attacks. Inspired by the training of GANs, they train a watermarked *target* DNN, which is competing against a *detector* DNN that aims to discover whether a watermark is embedded. While the above mechanisms were developed for models operating on image data (and thus mostly CNNs), Automatic Speech Recognition (ASR) has been considered as another modality (Chen et al., 2020).

5.4.7 Black-Box Model Watermarking

Black-box watermarking methods need only querying access to the model during watermark extraction and verification. Two proposed black-box mechanisms (Jia et al., 2020) (Szyller et al., 2020) address the second threat model case (illegal copy), the other primarily address the first case (legal copy); the latter all use backdoors embedded via data poisoning as the watermark. The backdoor consists of a so-called trigger set of input-output pairs, which are only known to the backdoor creator (in most cases, the model owner), and triggers a behaviour that is not predictable by others.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

Existing black-box watermarking methods concentrate on either creating suitable trigger images (inputs) or the output for the trigger image. Depending on the scheme, different triggers are used for watermarking:

- **Out-of-distribution** (OOD) triggers are completely unrelated to the dataset, for example abstract images in a handwritten digit dataset.
- **In-distribution** triggers are taken from the original training dataset and re-labelled wrongly.
- **Pattern based** triggers originate from the training dataset, but are marked with a pattern, e.g., logo, text or other designed pattern on an image - comparable to patterns embedded in images for "conventional" data poisoning attacks (e.g., (Gu et al., 2019)).
- **Noise** based triggers are taken from the training dataset, but with added noise (i.e., no systematic pattern). For image or video data, they can be either visible or invisible to the human eye.
- **Perturbatio**n based triggers are slightly perturbed images and lie near the classification boundary, thus when re-labelled, they force the model to slightly shift its classification boundary.

Similar to embedding backdoors as an attack to reduce the availability or integrity of a model, the main objective in watermarking is that the model will accurately behave on the main classification task, but will fail the classification on the trigger images in the way the model owner has designated. As there is a wealth of available techniques, we discuss only a relevant subset here. The interested reader is referred to e.g., (Lederer et al., n.d.) for a current survey on methods.

(Zhang et al., 2018) proposed the first black-box watermarking scheme in 2018 and introduced three types of trigger images: unrelated (OOD), content (pattern) and noise. Their work forms the basis for many following papers. Similar to and shortly afterwards, (Adi et al., 2018) proposed to include abstract images as triggers in the training dataset. Those abstract images are completely unrelated to the main classification task of the trained model; thus, it is highly unlikely that a model that has not seen this data point (i.e. one not watermarked) will label it as the designated class, and cause a false positive.
An improved pattern-based technique was proposed by (Li et al., 2020). They show that the schemes by Zhang et al. and Adi et al. are vulnerable to ownership piracy (i.e., an **overwriting** attack, in which an attacker aims to embed his own watermark into an already watermarked model. They propose a watermarking scheme that is especially robust against such attacks using a so-called *dual embedding*: the model is trained to classify (i) data with a pre-defined binary pattern correctly (*null embedding*), and (ii) data with an inverted pattern (binary bits are switched) incorrectly (*true embedding*). They observe that null embedding does not degrade the model's accuracy if the number of pixels in the pattern is sufficiently small, and that once a model is trained and null embedded, an adversary cannot null embed a pirate pattern without largely degrading the model's classification accuracy. Furthermore, they evaluate the robustness against Model Extraction Attacks, and conclude that, with enough (at least the same amount of) in-distribution data, the attacker is able to make a copy of the model without the watermark. For out-of-distribution data, the attacker would need 12.75 times more input data to reach similar accuracy.

As an in-distribution mechanism, (Namba and Sakuma, 2019) proposed an attack, called *query modification*, that investigates the query for trigger images in order to invalidate the watermark. With that in mind, they propose a scheme that is more robust especially against query modification but also model modifications like fine-tuning and model compression (e.g., pruning). The query modification as an attack exploits the fact that trigger images differ from original training images. Therefore, they propose to use trigger images that are selected from the training sample distribution. The trigger images are thus undetectable; however, the model is more likely to overfit to the (on purpose) wrongly labelled triggers, and thus more susceptible to removal attacks via e.g., pruning. The authors want to counter pruning by ensuring that the predictions do not depend on a large number of small model parameters that would likely be pruned. Thus, the model is first trained as

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

usual with the original training set. Then, the watermark is embedded by exponentially weighting the parameters and training the model on the union of the original dataset with the trigger set, which enforces the predictions to depend on a small number of large parameters instead.

A perturbation-based black-box scheme was proposed by (Merrer et al., 2019). The goal is to slightly shift the decision boundary of the model, by generating adversarial examples (Szegedy et al., 2014) for images close to the boundary, and changing the class for those adversaries to the neighbouring class. After fine-tuning the model, the decision boundary is adapted.

(Guo and Potkonjak, 2018) proposed to embed a pattern into the trigger images that can be clearly associated with the model owner's signature, e.g., a logo. The pattern should be embedded with little visibility so that the original model would still classify the trigger images to its original labels. The signature is used as a key to determine the pattern and then embedded in the image.

5.4.8 Evaluation of watermarking schemes

(H. Chen et al., 2018) performed an evaluation of fidelity of the models, as well as estimating the robustness against three attacks (model fine-tuning, parameter pruning, and watermark overwriting). It is worth noting that this is not an independent comparison, since the authors are also the authors of one of the considered watermarking methods, DeepSigns (Rouhani et al., 2019), which performs best in most of the results. As mentioned above, DeepSigns can be implemented as both white-box and black-box and therefore they are comparing it to one other white-box and three black-box methods. Even though the method *can* be considered as black-box, it is not backdoor-based and relies on the prediction vector for watermark verification, which might not be available in all settings.

In general, black-box watermarking schemes allow a wider application, as they are not limited to a specific access setting (i.e., they can be used in both black-box and white-box access). Thus, we selected and compared six different black-box watermarking schemes, namely (Zhang et al., 2018), (Adi et al., 2018), (Li et al., 2020), (Namba and Sakuma, 2019), (Merrer et al., 2019), and (Guo and Potkonjak, 2018). We use four different datasets: two datasets (MNIST and CIFAR-10) for training the models, and another two datasets (EMNIST and CINIC-10) for carrying out the fine-tuning attacks. We evaluate a range of parameters, including the size of the trigger set, which is rarely done in literature:

- Complexity of the model architecture: we choose eight state-of-the-art CNN architectures, inspired by the experimental setup in the papers proposing the mechanisms, namely SimpleNet and LeNet-1/3/5 for MNIST and DenseNet, ResNet-18/34/50 for CIFAR-10.
- Dataset characteristics: size of training set and images, and representation properties. We have chosen two datasets which both have a similar size of training set and a similar (small) size of images; we choose one RGB and one greyscale dataset.
- Size of trigger set: we vary between three trigger set sizes, i.e., 0.04%, 0.2% and 1% of the training set.
- Embedding type: we use two embedding types. Embedding from scratch - training the model on the union of the training dataset and the trigger set from the very beginning of training; and embedding on a pre-trained model - embedding the watermark as a fine-tuning step after training the model only on the original data.
- Complexity of attacks: we choose two common attacks, i.e., parameter pruning and fine-tuning, to test the robustness. The EMNIST and CINIC-10 datasets are used to simulate an attacker with a similar dataset.

Our evaluation shows a number of interesting trends, and allows us to recommend fitting schemes.

We tested fidelity for all watermarking methods and architectures and can clearly say, at least for MNIST, a more complex model is able to hold more watermark information without compromising

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

test accuracy, as we see on SimpleNet compared to the other architectures. SimpleNet is a very complex model for classifying grayscale images. But also comparing LeNet-1 and LeNet-3/5, we can see that a more complex model is less affected by a bigger trigger set.

For models trained on CIFAR-10, however, we do not see a clear trend. The test accuracy difference on ResNet-18/34/50 across the trigger set sizes is more or less the same. However, perturbation based and in-distribution methods perform worse on DenseNet, which could be related to DenseNet's small size.

The trigger set size does not influence the effectiveness of the watermarking methods. All watermarking methods reach 100%, or almost 100%, watermark accuracy, except for the perturbation-based method on the LeNets. Regarding fidelity, only smaller models trained on MNIST are influenced by the trigger set size.

Regarding robustness against pruning, the trigger set size has some influence. For most of the architectures and most of the methods, a bigger trigger set size is less robust. We speculate that a larger trigger set size increases the weights responsible for the original task in order to still be able to classify correctly on the original task, i.e., to cope with the influence of the trigger images.

For fine-tuning with a larger learning rate, no watermarking method could survive this type of attack. The models have a watermark accuracy mostly below 20% after the fine-tuning attack with a larger learning rate. With a smaller learning rate, however, we can again say, as with pruning, that a larger trigger set size leads to a higher watermark accuracy drop. The exception to this trend is SimpleNet, where a higher trigger set size leads to more robustness against fine-tuning, which could be related to SimpleNet's high complexity compared to the classification task.

However, we would have rather expected larger trigger sizes to have a positive effect on the effectiveness, as the model is trained on more data regarding the watermark. As we observe this odd behaviour only on (Merrer et al., 2019), this could be related to the perturbation-based trigger images and the magnitude of the perturbation. As perturbation-based trigger images are very much dependent on the model, this might imply that fine-tuning this hyperparameter needs to be done for each model type.

5.4.9 Summary and Integration in FeatureCloud

Watermarking of machine learning models can serve as a reactive method, to prove illicit use of a trained model once such a suspicion has surfaced. Several state-of-the art methods were analysed regarding effectiveness, fidelity and robustness based on several experiments with well-known Deep Learning (DL) architectures on widely used benchmark datasets. Large-scale evaluation shows that this defence mitigation mechanism comes at a low cost of effectiveness of the model on the original task (around 1%), and most methods are relatively robust against attacks, especially if the attacker is not in possession of a similar enough data set.

For the FeatureCloud platform, model watermarking is available as an app that can be utilised to post-process a model after training to embed a watermark into. It thus choses the approach to fine-tune already trained models, instead of performing embedding during the regular training. This allows for a relatively easy integration without setting restrictions on the original training algorithm. However, as the fine-tuning embedding is in most cases a bit less effective, also templates for integrating the embedding in the initial training are available.

The main goal of the watermarking app is to provide IP protection for the global model. However, local models can also be watermarked, before being sent to the aggregation.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

Several watermarking methods are made available within the FeatureCloud app. Depending on the exact setting, some might be more optimal to choose than others.

OOD methods ((Adi et al., 2018) and (Zhang et al., 2018) in the OOD variant) need a dedicated trigger set, not derived from the training data; this means that for potentially each watermarking setting, a specific trigger set needs to be compiled. Thus, methods based on in-distribution data or those that derive the trigger images from training data (via perturbation, etc.) have a certain convenience advantage.

For those model owners that are interested in fingerprinting, we believe that the most fitting methods would be (Zhang et al., 2018) with their pattern approach, (Guo and Potkonjak, 2018) and (Li et al., 2020), as the watermark generation can be easily customised, in order to embed unique watermarks for multiple users.

Future work regarding watermarking will include adapting and developing methods for other model types besides neural networks, as well as adapting and evaluating methods for fingerprinting.

6 Conclusion

This deliverable discussed mitigation mechanisms that are considered for the FeatureCloud platform to address the threats specific to machine learning and federated learning, originating both from the risk assessment performed in Deliverable D2.1 “Risk assessment methodology” as well as the objectives set in the FeatureCloud research proposal. The mitigations primarily address four different threats to confidentiality to data and information along the federated learning process performed in FeatureCloud. These include:

- The confidentiality and **privacy of inputs** to the federated learning resp. the model averaging. While this data is already an abstraction of the original, raw data held locally, recent studies have shown that it is still possible to infer information about the training data from these models, such as membership inference. Thus, to rule out misuse by the coordinator averaging the models, Secure Multiparty Computation (SMPC) has been selected as a strategy to protect the inputs from a global coordinator.
- The output of the aggregation process, i.e., the global model, can suffer from the same issues as local models, i.e., it might be that information on the training data can still be inferred therefrom. While SMPC protects the inputs, which subsequently do not need to be published anymore, the result of this aggregation is then shared with multiple parties - the participants of the learning process, but also users that want to employ the model in e.g., a predictive task. Thus, the **output of the learning process** needs to be protected as well. Attacks are often very specific to the data, models and other parameters, as a generic solution to this. Thus, the platform will thus provide differential privacy that can be integrated in workflows running within FeatureCloud.
- **Theft of input data** can happen where the apps get access to the data in the local execution environments. Here, the platform will build on a series of defences, built on (i) recommendation to isolate and sandbox the apps, to (ii) monitoring of the amount of data transferred, and (iii) monitoring of the content of the data transferred to the global model, to detect attacks that use the model as channel for exfiltration.
- **Theft of intellectual property** can occur if attackers try to make illicit use of the models trained collaboratively. Especially if complex models are trained, or the training data is rare, the resulting (global) model constitutes a valuable asset. Illicit use should be deterred, for which we provide a generic app within the FeatureCloud environment to **watermark** models. This reactive method then allows to gather evidence when illicit use is suspected.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

This document, and the implemented mechanisms, will be constantly monitored within the corresponding work package for new and novel vulnerabilities and attacks. If required, they will be updated to the current threat landscape, and extended by novel mitigation mechanisms. Further, currently provided mechanisms will be extended to novel types of analysis mechanisms not yet considered, e.g., different types of models.

7 References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L., 2016. Deep Learning with Differential Privacy, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, CCS '16. ACM, New York, NY, USA, pp. 308–318. https://doi.org/10.1145/2976749.2978318

Abdelnabi, S., Fritz, M., 2020. Adversarial Watermarking Transformer: Towards Tracing Text Provenance with Data Hiding.

Adi, Y., Baum, C., Cisse, M., Pinkas, B., Keshet, J., 2018. Turning Your Weakness Into a Strength: Watermarking Deep Neural Networks by Backdooring, in: USENIX Security Symposium. USENIX Association, pp. 1615–1631.

Aïvodji, U., Gambs, S., Ther, T., 2019. GAMIN: An Adversarial Approach to Black-Box Model Inversion.

Ateniese, G., Mancini, L.V., Spognardi, A., Villani, A., Vitali, D., Felici, G., 2015. Hacking Smart Machines with Smarter Ones: How to Extract Meaningful Data from Machine Learning Classifiers. Int. J. Secur. Netw. 10, 137–150. https://doi.org/10.1504/IJSN.2015.071829

Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., Shmatikov, V., 2020. How To Backdoor Federated Learning, in: Proceedings of the 23nd International Conference on Artificial Intelligence and Statistics (AISTATS). Palermo, Sicily, Italy.

Banerjee, M., Karimi Adl, R., Wu, L., Barker, K., 2011. Quantifying Privacy Violations, in: Jonker, W., Petković, M. (Eds.), Secure Data Management, Lecture Notes in Computer Science. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1–17. https://doi.org/10.1007/978-3-642-23556-6_1

Beimel, A., Chor, B., 1994. Universally ideal secret-sharing schemes. IEEE Trans. Inf. Theory 40, 786–794. https://doi.org/10.1109/18.335890

Benhamouda, F., Herranz, J., Joye, M., Libert, B., 2017. Efficient Cryptosystems From $$\mathbf{2}^{{\varvec{k}}}$$ 2 k -th Power Residue Symbols. J. Cryptol. 30, 519–549. https://doi.org/10.1007/s00145-016-9229-5

Blakley, G.R., 1979. Safeguarding cryptographic keys, in: Managing Requirements Knowledge, International Workshop On. IEEE Computer Society, Los Alamitos, CA, USA, p. 313. https://doi.org/10.1109/AFIPS.1979.98

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H.B., Patel, S., Ramage, D., Segal, A., Seth, K., 2017. Practical Secure Aggregation for Privacy-Preserving Machine Learning, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS '17. Association for Computing Machinery, New York, NY, USA, pp. 1175–1191. https://doi.org/10.1145/3133956.3133982

Carlini, N., Liu, C., Erlingsson, Ú., Kos, J., Song, D., 2019. The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks, in: Proceedings of the 28th {USENIX} Security Symposium. Presented at the 28th {USENIX} Security Symposium, pp. 267–284.

Carlini, N., Liu, C., Kos, J., Erlingsson, Ú., Song, D., 2018. The secret sharer: Measuring unintended neural network memorization & extracting secrets.

Chase, M., Gilad-Bachrach, R., Laine, K., Lauter, K., Rindal, P., 2017. Private Collaborative Neural Network Learning. IACR Cryptol EPrint Arch.

Chaudhuri, K., Monteleoni, C., Sarwate, A.D., 2011. Differentially Private Empirical Risk

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

Minimization. J. Mach. Learn. Res. 12, 1069–1109.
Chen, B., Carvalho, W., Baracaldo, N., Ludwig, H., Edwards, B., Lee, T., Molloy, I., Srivastava, B., 2018. Detecting Backdoor Attacks on Deep Neural Networks by Activation Clustering. CoRR abs/1811.03728.
Chen, H., Darvish, B., Koushanfar, F., 2020. SpecMark: A Spectral Watermarking Framework for IP Protection of Speech Recognition Systems, in: Interspeech. Presented at the Interspeech, ISCA, pp. 2312–2316. https://doi.org/10.21437/Interspeech.2020-2787
Chen, H., Rouhani, B.D., Fan, X., Kilinc, O.C., Koushanfar, F., 2018. Performance Comparison of Contemporary DNN Watermarking Techniques.
Dandekar, A., Basu, D., Bressan, S., 2021. Differential Privacy at Risk: Bridging Randomness and Privacy Budget. Proc. Priv. Enhancing Technol. 2021, 64–84. https://doi.org/10.2478/popets-2021-0005
Dankar, F.K., El Emam, K., 2010. A method for evaluating marketer re-identification risk, in: Proceedings of the 1st International Workshop on Data Semantics - DataSem '10. Presented at the the 1st International Workshop, ACM Press, Lausanne, Switzerland, p. 1. https://doi.org/10.1145/1754239.1754271
D'Orazio, C.J., Choo, K.-K.R., Yang, L.T., 2017. Data Exfiltration From Internet of Things Devices: iOS Devices as Case Studies. IEEE Internet Things J. 4, 524–535. https://doi.org/10.1109/JIOT.2016.2569094
Dwork, C., 2008. Differential Privacy: A Survey of Results, in: Agrawal, M., Du, D., Duan, Z., Li, A. (Eds.), Theory and Applications of Models of Computation, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 1–19. https://doi.org/10.1007/978-3-540-79228-4_1
Dwork, C., 2006. Differential Privacy, in: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (Eds.), Automata, Languages and Programming, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 1–12. https://doi.org/10.1007/11787006_1
Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., Naor, M., 2006a. Our Data, Ourselves: Privacy Via Distributed Noise Generation, in: Vaudenay, S. (Ed.), Advances in Cryptology - EUROCRYPT 2006, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 486–503. https://doi.org/10.1007/11761679_29
Dwork, C., McSherry, F., Nissim, K., Smith, A., 2006b. Calibrating Noise to Sensitivity in Private Data Analysis, in: Halevi, S., Rabin, T. (Eds.), Theory of Cryptography, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 265–284. https://doi.org/10.1007/11681878_14
Dwork, C., Roth, A., 2014. The Algorithmic Foundations of Differential Privacy. Found. Trends® Theor. Comput. Sci. 9, 211–407. https://doi.org/10.1561/0400000042
El Emam, K., 2011. Methods for the de-identification of electronic health records for genomic research. Genome Med. 3, 25. https://doi.org/10.1186/gm239
Elliot, M., 2005. Statistical Disclosure Control, in: Encyclopedia of Social Measurement. Elsevier. https://doi.org/10.1016/B0-12-369398-5/00378-9
Elliot, M.J., Dale, A., 1999. Scenarios of attack: the data intruder's perspective on statistical disclosure risk. Neth. Off. Stat. 14.
Evans, D., Kolesnikov, V., Rosulek, M., 2018. A Pragmatic Introduction to Secure Multi-Party Computation. Found. Trends® Priv. Secur. 2, 70–246. https://doi.org/10.1561/3300000019
Fredrikson, M., Jha, S., Ristenpart, T., 2015. Model Inversion Attacks That Exploit Confidence Information and Basic Countermeasures, in: Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security (CCS), CCS '15. ACM, New York, NY, USA, pp. 1322–1333. https://doi.org/10.1145/2810103.2813677
Fredrikson, M., Lantz, E., Jha, S., Lin, S., Page, D., Ristenpart, T., 2014. Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing, in: Proceedings of the 23rd {USENIX} Security Symposium. Presented at the 23rd {USENIX} Security Symposium, pp. 17–32.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

Ganju, K., Wang, Q., Yang, W., Gunter, C.A., Borisov, N., 2018. Property Inference Attacks on Fully Connected Neural Networks using Permutation Invariant Representations. p. 15.

Garfinkel, S.L., 2015. De-identification of personal information (No. NIST IR 8053). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.IR.8053

Gascón, A., Schoppmann, P., Balle, B., Raykova, M., Doerner, J., Zahur, S., Evans, D., 2017. Privacy-Preserving Distributed Linear Regression on High-Dimensional Data. Proc. Priv. Enhancing Technol. 2017, 345–364. https://doi.org/10.1515/popets-2017-0053

Geng, Q., Viswanath, P., 2016. The Optimal Noise-Adding Mechanism in Differential Privacy. IEEE Trans. Inf. Theory 62, 925–951. https://doi.org/10.1109/TIT.2015.2504967

Gentry, C., 2009. Fully homomorphic encryption using ideal lattices, in: Proceedings of the 41st Annual ACM Symposium on Symposium on Theory of Computing - STOC '09. Presented at the the 41st annual ACM symposium, ACM Press, Bethesda, MD, USA, p. 169. https://doi.org/10.1145/1536414.1536440

Geyer, R., Klein, T., Nabi, M., 2017. Differentially Private Federated Learning: A Client Level Perspective.

Goldreich, O., Micali, S., Wigderson, A., 1987. How to Play ANY Mental Game, in: Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing, STOC '87. Association for Computing Machinery, New York, NY, USA, pp. 218–229. https://doi.org/10.1145/28395.28420

Gong, M., Xie, Y., Pan, K., Feng, K., Qin, A.K., 2020. A Survey on Differentially Private Machine Learning [Review Article]. IEEE Comput. Intell. Mag. 15, 49–64. https://doi.org/10.1109/MCI.2020.2976185

Gu, T., Dolan-Gavitt, B., Garg, S., 2017. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain, in: Proceedings of the Machine Learning and Computer Security Workshop. Long Beach, CA, USA.

Gu, T., Liu, K., Dolan-Gavitt, B., Garg, S., 2019. BadNets: Evaluating Backdooring Attacks on Deep Neural Networks. IEEE Access 7, 47230–47244. https://doi.org/10.1109/ACCESS.2019.2909068

Guan, X., Feng, H., Zhang, W., Zhou, H., Zhang, J., Yu, N., 2020. Reversible Watermarking in Deep Convolutional Neural Networks for Integrity Authentication, in: International Conference on Multimedia, MM '20. ACM, Seattle, USA, pp. 2273–2280. https://doi.org/10.1145/3394171.3413729

Guo, J., Potkonjak, M., 2018. Watermarking deep neural networks for embedded systems, in: International Conference on Computer-Aided Design, ICCAD '18. ACM, San Diego, USA, pp. 1–8. https://doi.org/10.1145/3240765.3240862

Hu, H., Salcic, Z., Sun, L., Dobbie, G., Yu, P.S., Zhang, X., 2021. Membership Inference Attacks on Machine Learning: A Survey.

Jagannathan, G., Pillaipakkamnatt, K., Wright, R.N., 2009. A Practical Differentially Private Random Decision Tree Classifier, in: 2009 IEEE International Conference on Data Mining Workshops. Presented at the 2009 IEEE International Conference on Data Mining Workshops, pp. 114–121. https://doi.org/10.1109/ICDMW.2009.93

Jia, H., Choquette-Choo, C.A., Papernot, N., 2020. Entangled Watermarks as a Defense against Model Extraction.

Kahng, A.B., Lach, J., Mangione-Smith, W.H., Mantik, S., Markov, I.L., Potkonjak, M., Tucker, P., Wang, H., Wolfe, G., 1998. Watermarking techniques for intellectual property protection, in: Proceedings of the 35th Annual Design Automation Conference, DAC '98. Association for Computing Machinery, New York, NY, USA, pp. 776–781. https://doi.org/10.1145/277044.277240

Kamran, M., Farooq, M., 2018. A Comprehensive Survey of Watermarking Relational Databases Research. ArXiv180108271 Cs.

Lach, J., Mangione-Smith, W.H., Potkonjak, M., 1998. FPGA fingerprinting techniques for protecting intellectual property, in: Proceedings of the IEEE 1998 Custom Integrated

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

Circuits Conference (Cat. No.98CH36143). Presented at the IEEE 1998 Custom Integrated Circuits Conference, IEEE, Santa Clara, CA, USA, pp. 299–302. https://doi.org/10.1109/CICC.1998.694986

Lederer, I., Mayer, R., Rauber, A., n.d. Identifying Appropriate Intellectual Property Protection Mechanisms for Machine Learning Models: A Systematization of Watermarking, Fingerprinting, Model Access, and Attacks.

Li, H., Wenger, E., Zhao, B.Y., Zheng, H., 2020. Piracy Resistant Watermarks for Deep Neural Networks.

Li, Z., Hu, C., Zhang, Y., Guo, S., 2019. How to prove your model belongs to you: a blind-watermark based framework to protect intellectual property of DNN, in: Annual Computer Security Applications Conference, ACSAC '19. ACM, San Juan, USA, pp. 126–137. https://doi.org/10.1145/3359789.3359801

Lindell, Y., Pinkas, B., Smart, N.P., 2008. Implementing Two-Party Computation Efficiently with Security Against Malicious Adversaries, in: Ostrovsky, R., De Prisco, R., Visconti, I. (Eds.), Security and Cryptography for Networks, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 2–20. https://doi.org/10.1007/978-3-540-85855-3_2

Liu, K., Dolan-Gavitt, B., Garg, S., 2018. Fine-Pruning: Defending Against Backdooring Attacks on Deep Neural Networks, in: Bailey, M., Holz, T., Stamatogiannakis, M., Ioannidis, S. (Eds.), Research in Attacks, Intrusions, and Defenses - 21st International Symposium, RAID 2018, Heraklion, Crete, Greece, September 10-12, 2018, Proceedings, Lecture Notes in Computer Science. Springer, pp. 273–294. https://doi.org/10.1007/978-3-030-00470-5_13

Liu, X., Li, Q., Li, T., Chen, D., 2018. Differentially private classification with decision tree ensemble. Appl. Soft Comput. 62, 807–816. https://doi.org/10.1016/j.asoc.2017.09.010

Lukas, N., Zhang, Y., Kerschbaum, F., 2020. Deep Neural Network Fingerprinting by Conferrable Adversarial Examples.

McMahan, H., Ramage, D., Talwar, K., Zhang, L., 2017. Learning Differentially Private Language Models Without Losing Accuracy.

Merrer, E.L., Perez, P., Trédan, G., 2019. Adversarial Frontier Stitching for Remote Neural Network Watermarking. Neural Comput. Appl. 32, 9233–9244. https://doi.org/10.1007/s00521-019-04434-z

Mohassel, P., Zhang, Y., 2017. SecureML: A System for Scalable Privacy-Preserving Machine Learning, in: 2017 IEEE Symposium on Security and Privacy (SP). Presented at the 2017 IEEE Symposium on Security and Privacy (SP), IEEE, San Jose, CA, USA, pp. 19–38. https://doi.org/10.1109/SP.2017.12

Morkel, T., Eloff, J., Olivier, M., 2005. An overview of image steganography.

Namba, R., Sakuma, J., 2019. Robust Watermarking of Neural Network with Exponential Weighting, in: Asia Conference on Computer and Communications Security, Asia CCS '19. ACM, Auckland, New Zealand, pp. 228–240. https://doi.org/10.1145/3321705.3329808

Park, M., Foulds, J.R., Choudhary, K., Welling, M., 2017. DP-EM: Differentially Private Expectation Maximization, in: AISTATS.

Phan, N.H., Wu, X., Hu, H., Dou, D., 2017. Adaptive Laplace Mechanism: Differential Privacy Preservation in Deep Learning.

Por, L.Y., Delina, B., 2008. Information hiding: A new approach in text steganography, in: WSEAS International Conference. Proceedings. Mathematics and Computers in Science and Engineering. World Scientific and Engineering Academy and Society.

Prasser, F., Kohlmayer, F., Lautenschläger, R., Kuhn, K.A., 2014. ARX--A Comprehensive Tool for Anonymizing Biomedical Data. AMIA Annu. Symp. Proc. AMIA Symp. 2014, 984–993.

Rivest, R.L., Adleman, L., Dertouzos, M.L., 1978. On Data Banks and Privacy Homomorphisms. Found. Secure Comput. Acad. Press 169–179.

Rouhani, B.D., Chen, H., Koushanfar, F., 2019. DeepSigns: An End-to-End Watermarking Framework for Ownership Protection of Deep Neural Networks, in: International Conference on Architectural Support for Programming Languages and Operating Systems,

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

ASPLOS '19. ACM, Providence, USA, pp. 485–497. https://doi.org/10.1145/3297858.3304051
Sablayrolles, A., Douze, M., Schmid, C., Jégou, H., 2020. Radioactive data: tracing through training.
Salem, A., Zhang, Y., Humbert, M., Berrang, P., Fritz, M., Backes, M., 2019. ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models, in: Proceedings 2019 Network and Distributed System Security Symposium. Presented at the Network and Distributed System Security Symposium, Internet Society, San Diego, CA. https://doi.org/10.14722/ndss.2019.23119
Sallee, P., 2003. Model-based steganography, in: International Workshop on Digital Watermarking. Springer, pp. 154–167.
Shamir, A., 1979. How to share a secret. Commun. ACM 22, 612–613. https://doi.org/10.1145/359168.359176
Sharma, S.S., Chandrasekaran, V., 2020. A robust hybrid digital watermarking technique against a powerful CNN-based adversarial attack. Multimed. Tools Appl. 79, 32769–32790. https://doi.org/10.1007/s11042-020-09555-5
Shokri, R., Shmatikov, V., 2015. Privacy-Preserving Deep Learning, in: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security - CCS '15. Presented at the the 22nd ACM SIGSAC Conference, ACM Press, Denver, Colorado, USA, pp. 1310–1321. https://doi.org/10.1145/2810103.2813687
Shokri, R., Stronati, M., Song, C., Shmatikov, V., 2017. Membership Inference Attacks Against Machine Learning Models, in: 2017 IEEE Symposium on Security and Privacy (SP). Presented at the 2017 IEEE Symposium on Security and Privacy (SP), IEEE, San Jose, CA, USA, pp. 3–18. https://doi.org/10.1109/SP.2017.41
Snyder, P., 2014. Yao ' s Garbled Circuits : Recent Directions and Implementations [WWW Document]. URL https://www.semanticscholar.org/paper/Yao-%E2%80%99-s-Garbled-Circuits-%3A-Recent-Directions-and-Snyder/7008813dba1057d62ee863f2383ba2914d330b72 (accessed 12.3.21).
So, J., Güler, B., Avestimehr, A.S., 2021. Turbo-Aggregate: Breaking the Quadratic Aggregation Barrier in Secure Federated Learning. IEEE J. Sel. Areas Inf. Theory 2, 479–489. https://doi.org/10.1109/JSAIT.2021.3054610
Song, C., Ristenpart, T., Shmatikov, V., 2017. Machine Learning Models That Remember Too Much, in: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS), CCS '17. ACM, New York, NY, USA, Dallas, Texas, USA, pp. 587–601. https://doi.org/10.1145/3133956.3134077
Sweeney, L., 2013. Matching known patients to health records in Washington State data. Available SSRN 2289850.
Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I.J., Fergus, R., 2014. Intriguing properties of neural networks, in: Proceedings of the International Conference on Learning Representations (ICLR). Presented at the International Conference on Learning Representations, Banff, AB, Canada.
Szyller, S., Atli, B.G., Marchal, S., Asokan, N., 2020. DAWN: Dynamic Adversarial Watermarking of Neural Networks.
Thakkar, O., Ramaswamy, S., Mathews, R., Beaufays, F., 2020. Understanding unintended memorization in federated learning. ArXiv Prepr. ArXiv200607490.
Tran, B., Li, J., Madry, A., 2018. Spectral Signatures in Backdoor Attacks. CoRR abs/1811.00636.
Truex, S., Baracaldo, N., Anwar, A., Steinke, T., Ludwig, H., Zhang, R., Zhou, Y., 2019a. A Hybrid Approach to Privacy-Preserving Federated Learning, in: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, AISec'19. Association for Computing Machinery, New York, NY, USA, pp. 1–11. https://doi.org/10.1145/3338501.3357370
Truex, S., Liu, L., Gursoy, M.E., Wei, W., Yu, L., 2019b. Effects of Differential Privacy and Data Skewness on Membership Inference Vulnerability, in: 2019 First IEEE International

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA). Presented at the 2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), IEEE, Los Angeles, CA, USA, pp. 82–91. https://doi.org/10.1109/TPS-ISA48467.2019.00019

Truex, S., Liu, L., Gursoy, M.E., Yu, L., Wei, W., 2019c. Demystifying Membership Inference Attacks in Machine Learning as a Service. IEEE Trans. Serv. Comput. 1–1. https://doi.org/10.1109/TSC.2019.2897554

Uchida, Y., Nagai, Y., Sakazawa, S., Satoh, S., 2017. Embedding Watermarks into Deep Neural Networks, in: International Conference on Multimedia Retrieval, ICMR '17. ACM, Bucharest, Romania, pp. 269–277. https://doi.org/10.1145/3078971.3078974

Vaidya, J., Shafiq, B., Basu, A., Hong, Y., 2013. Differentially Private Naive Bayes Classification, in: 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT). Presented at the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), IEEE, Atlanta, GA, USA, pp. 571–576. https://doi.org/10.1109/WI-IAT.2013.80

Van Antwerp, R., 2011. Exfiltration techniques: an examination and emulation (Thesis). University of Delaware.

Veale, M., Binns, R., Edwards, L., 2018. Algorithms that remember: model inversion attacks and data protection law. Philos. Trans. R. Soc. Math. Phys. Eng. Sci. 376, 20180083. https://doi.org/10.1098/rsta.2018.0083

Wang, T., Kerschbaum, F., 2020. Robust and Undetectable White-Box Watermarks for Deep Neural Networks.

Wei, K., Li, J., Ding, M., Ma, C., Yang, H.H., Farokhi, F., Jin, S., Quek, T.Q.S., Poor, H.V., 2020. Federated Learning With Differential Privacy: Algorithms and Performance Analysis. IEEE Trans. Inf. Forensics Secur. 15, 3454–3469. https://doi.org/10.1109/TIFS.2020.2988575

Xiong, L., Gardner, J., Jurczyk, P., Lu, J.J., 2009. Privacy-Preserving Information Discovery on EHRs, in: Information Discovery on Electronic Health Records. Chapman and Hall/CRC, pp. 215–244.

Yeom, S., Giacomelli, I., Fredrikson, M., Jha, S., 2018. Privacy risk in machine learning: Analyzing the connection to overfitting, in: 2018 IEEE 31st Computer Security Foundations Symposium (CSF). IEEE, pp. 268–282.

Yingjiu Li, Swarup, V., Jajodia, S., 2005. Fingerprinting relational databases: schemes and specialties. IEEE Trans. Dependable Secure Comput. 2, 34–45. https://doi.org/10.1109/TDSC.2005.12

Younes, M.A.B., Jantan, A., others, 2008. A new steganography approach for images encryption exchange by using the least significant bit insertion. Int. J. Comput. Sci. Netw. Secur. 8, 247–257.

Zhang, C., Li, S., Xia, J., Wang, W., Yan, F., Liu, Y., 2020. BatchCrypt: efficient homomorphic encryption for cross-silo federated learning, in: Proceedings of the 2020 USENIX Conference on Usenix Annual Technical Conference. USENIX Association, USA, pp. 493–506.

Zhang, J., Chen, D., Liao, J., Fang, H., Zhang, W., Zhou, W., Cui, H., Yu, N., 2020. Model Watermarking for Image Processing Networks, in: AAAI 2020. pp. 12805–12812.

Zhang, J., Gu, Z., Jang, J., Wu, H., Stoecklin, M.Ph., Huang, H., Molloy, I., 2018. Protecting Intellectual Property of Deep Neural Networks with Watermarking, in: Asia Conference on Computer and Communications Security, ASIACCS '18. ACM Press, Incheon, Republic of Korea, pp. 159–172. https://doi.org/10.1145/3196494.3196550

Zhang, J., Zhang, Z., Xiao, X., Yang, Y., Winslett, M., 2012. Functional mechanism: regression analysis under differential privacy. Proc. VLDB Endow. 5, 1364–1375. https://doi.org/10.14778/2350229.2350253

Zhang, X., Ji, S., Wang, H., Wang, T., 2017. Private, Yet Practical, Multiparty Deep Learning, in: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

Presented at the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), IEEE, Atlanta, GA, USA, pp. 1442–1452. https://doi.org/10.1109/ICDCS.2017.215

Zhang, Y., Jia, R., Pei, H., Wang, W., Li, B., Song, D., 2020. The Secret Revealer: Generative Model-Inversion Attacks Against Deep Neural Networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

Zhong, X., Huang, P.-C., Mastorakis, S., Shih, F.Y., 2020. An Automated and Robust Image Watermarking Scheme Based on Deep Neural Networks. IEEE Trans. Multimed. 1–1. https://doi.org/10.1109/TMM.2020.3006415

Zhu, H., Mong Goh, R.S., Ng, W.-K., 2020. Privacy-Preserving Weighted Federated Learning Within the Secret Sharing Framework. IEEE Access 8, 198275–198284. https://doi.org/10.1109/ACCESS.2020.3034602

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

8 Table of acronyms and definitions

AI	Artificial Intelligence
BB	Black Box
CIA	Confidentiality-Integrity-Availability
concentris	concentris research management GmbH
DP	Differential Privacy
FP	Fingerprint
GND	Gnome Design SRL
KPI	Key Performance Indicator
ML	Machine Learning
MS	Milestone
MUG	Medizinische Universitaet Graz
Patients	In this deliverable, we use the term “patients” for all research subjects. In FeatureCloud, we will focus on patients, as this is already the most vulnerable case scenario and this is where most primary data is available to us. Admittedly, some research subjects participate in clinical trials but not as patients but as healthy individuals, usually on a voluntary basis and are therefore not dependent on the physicians who care for them. Thus, to increase readability, we simply refer to them as “patients”.
RI	Research Institute AG & Co. KG
SBA	SBA Research Gemeinnutzige GmbH
SDU	Syddansk Universitet
SMPC	Secure Multiparty Computation
TUM	Technische Universitaet Muenchen
UHAM	University of Hamburg
UM	Universiteit Maastricht
UMR	Philipps Universitaet Marburg
WB	White Box
WM	Watermark
WP	Work package

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.