Project no. 826278

SERUMS

Research & Innovation Action (RIA)
SECURING MEDICAL DATA IN SMART PATIENT-CENTRIC HEALTHCARE SYSTEMS

Report on Refined Privacy-Preserving Distributed Deep Learning and Initial Work on Privacy-Preserving Transfer and Multi-Task Learning
D3.2

Due date of deliverable: 30th September 2020

Start date of project: January 1st, 2019

Type: Deliverable
WP number: WP3

Responsible institution: Software Competence Center Hagenberg
Editor and editor's address: Michael Rossbory, Software Competence Center Hagenberg

Version 1.0

Project co-funded by the European Commission within the Horizon 2020 Programme		
Dissemination Level		
PU	Public	√
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Change Log

Rev.	Date	Who	Site	What
1	01/09/20	Michael Rossbory	SCCH	Document Structure
2	10/09/20	Mohit Kumar	SCCH	Initial Version
3	28/09/20	Eduard Baranov	UCL	Update Task 3.4
4	28/09/20	Thomas Given-Wilson	UCL	Update Task 3.4

Executive Summary

This deliverable describes that differentially private distributed deep learning framework reported in D3.1 can be complemented by an analytical framework to evaluate informational privacy. We suggest an information theoretic approach to quantify privacy-leakage in-terms of mutual information between private/sensitive data and the publically released data. At the core of the method lies a variational Bayesian fuzzy model approximating the uncertain mapping between released noise added data and private data such that the model is employed for variational approximation of informational privacy. The deliverable further presents a novel differentially private semi-supervised transfer learning framework that 1) is able to handle high-dimensional data and heterogeneity of domains; 2) uses an optimal noise adding mechanism achieving a given level of privacy-loss bound with the minimum perturbation of the data; 3) does not require an access to private/sensitive source domain training data for the learning of target domain model; 4) employs deep models to use data features at different abstraction levels for transferring knowledge across domains; 5) provides a robustness of the target model towards the perturbations in source data caused by the privacy requirements demanded by source data owner.

Contents

1. Privacy-Preserving Distributed Deep Learning (T3.1)

1.1 Summary of Work Reported in D3.1 on "Differentially Private Distributed Deep Learning"

The work reported in D3.1 introduced a privacy-preserving framework for distributed deep fuzzy learning. Specifically, sufficient conditions for $(\epsilon, \delta)-$ differential privacy of the learning algorithm were derived. Following the entropy based approach of [1], the optimal noise distribution that minimizes the expected noise magnitude together with satisfying the sufficient conditions for $(\epsilon, \delta)-$differential privacy was derived. A comparison of the derived noise adding mechanism with the classical Gaussian mechanism was made and a multi-fold reduction (e.g. by more than 4 times in the high privacy regime) in the magnitude of noise over the Gaussian mechanism was observed. To study distributed deep learning problem in private setting, a deep model, formed by a composition of a finite number of Takagi-Sugeno fuzzy filters, was considered. Variational Bayes, a widely used Bayesian inference method, was applied for the learning of deep fuzzy model. We considered a particular configuration of the deep fuzzy model, referred to as *deep fuzzy autoencoder*, for data representation learning. The flexibility and robustness features offered by fuzzy sets and fuzzy rules were next leveraged to facilitate a distributed learning from the data locally owned by different participant. A fuzzy set in the multi-dimensional real space was associated to each local deep fuzzy model. The post-processing invariance property of differential privacy allowed to build a global fuzzy rule-based classifier that *aggregates* the fuzzy sets associated to local private deep models using a logical operator.

The motivation of the approach was derived from the idea of employing fuzzy sets and rules based systems for an optimal differentially private data representation learning at varying abstraction levels using variational Bayesian deep fuzzy models in a distributed setting. We hypothesize that a privacy-preserving framework for distributed learning of deep models could be benefitted from

1. an analytical optimization of differentially private noise adding mechanism to optimize the privacy-utility trade-off;

2. an incorporation of a statistical noise model in variational Bayesian inference of deep models for a robustness towards noise;

3. flexible and robust combination of local private deep models by means of fuzzy sets and fuzzy rules.

The work provided an $(\epsilon, \delta)-$differentially private noise adding mechanism that results in multi-fold reduction in noise magnitude over the classical Gaussian mechanism and thus leads to an increased utility for a given level of privacy. An architecture for distributed form of differentially private learning was presented where a privacy wall separates the private local training data from the globally shared data, and fuzzy sets and fuzzy rules are used to aggregate robustly the local deep models for building the global model.

The proposed methodology was evaluated via performing experiments. The aim of the experiments was to

1. study the effect of privacy level on the classification accuracy of the proposed method,

2. compare the proposed noise adding mechanism with the classical Gaussian mechanism in terms of classification accuracy,

3. compare the non-private version of the proposed distributed deep fuzzy models based classifier with the classical machine learning methods in classifying high-dimensional data.

The experiments on "MNIST", "Freiburg Groceries", "Caltech-101", and "Caltech-256" datasets have validated the competitive performance of the proposed method.

1.2 Limitations of Differential Privacy and Motivation of Informational Privacy

Differential privacy guarantees that an adversary, by virtue of presence or absence of an individual's data in the dataset, can't draw any conclusions about an individual from the released output of the analysis algorithm. Differential privacy, however, doesn't always adequately limit inference about participation of a single record in the database [2]. Differential privacy requirement does not necessarily constrain the information leakage from a data set [3]. Correlation among records of a dataset would degrade the expected privacy guarantees of differential privacy mechanism [4]. These limitations of differential privacy motivate an information theoretic approach to privacy where privacy is quantified by the mutual information between sensitive information and the released data [3, 5–8]. A data release mechanism aims to provide useful data available while simultaneously limiting any reveled sensitive information. The data perturbation approach uses a random noise adding mechanism to preserve privacy, however, results in distortion of useful data

and thus utility of any subsequent machine learning and data analytics algorithm is adversely affected. There remains the challenge of studying and optimizing privacy-utility tradeoff especially in the case when statistical distributions of data are unknown. Information theoretic privacy can be optimized theoretically using a prior knowledge about data statistics. However, in practice, a prior knowledge (such as joint distributions of public and private variables) is missing and therefore a data-driven approach based on generative adversarial networks has been suggested [9]. The data-driven approach of [9] leverages recent advancements in generative adversarial networks to allow learning the parameters of the privatization mechanism. However, the framework of [9] is limited to only binary type of sensitive variables. A similar approach [10] applicable to arbitrary distributions (discrete, continuous, and/or multivariate) of variables employs adversarial training to perform a variational approximation of mutual information privacy. The approach of approximating mutual information via a variational lower bound was also used in [11].

1.3 A Novel Information Theoretic Approach to Evaluation of Privacy-Leakage in Distributed Deep Learning

A novel approach is introduced to study the informational privacy of a data release mechanism. The privacy-leakage problem is mathematically formulated where a sample of sensitive or private data x ($x \in \mathcal{X} \subseteq \mathbb{R}^n$), corresponding observed data vector y ($y \in \mathcal{Y} \subseteq \mathbb{R}^p$), and the released data vector z ($z \in \mathcal{Z} \subseteq \mathbb{R}^p$) are modeled as random variables. A privacy-preserving mechanism to release data vector z will add random noise $v \in \mathbb{R}^p$ (sampled from a density function, say $q(v)$) to the observed data vector y, i.e.,

$$z(v; y) = y + v. \tag{1.1}$$

A relevant problem here is to evaluate the privacy-leakage in-terms of mutual information $I(x; z)$. The privacy-leakage $I(x; z)$ can be analytically derived for a known data distribution $P_{X,Y}(x, y)$ over the space $\mathcal{X} \times \mathcal{Y}$. The framework proposed in this study, however without knowing data distribution, allows to evaluate privacy-leakage. This is done as follows:

1. The privacy of sensitive data is preserved via adding random noise (sampled from derived optimal distribution) to the data observations, i.e., eq. (1.1). Only the noise added data observations are meant to be publicly released.

2. Given a finite set of private-public data pairs $\{(x^i, z^i) \mid i \in \{1, \cdots, N\}\}$, a stochastic fuzzy model $\mathcal{G}$ is built using variational Bayesian methodology such that $x^i = \mathcal{G}(z^i) + v^i$, where $v^i \in \mathbb{R}^n$ is the disturbance vector affecting the data model.

3. A lower bound on privacy-leakage is derived as a functional of distributions characterizing the data model, i.e., $I(x;z) \geq I_L(q(\alpha,\beta))$, where $q(\alpha,\beta)$ is an arbitrary probability density function on parameters α and β which characterizes the distributions related to data model: $x = \mathcal{G}(z) + v$.

4. An approximation to $I(x;z)$ is provided via maximizing I_L w.r.t. $q(\alpha,\beta)$, i.e., $\hat{I}(x;z) = \max_{q(\alpha,\beta)} I_L(q(\alpha,\beta))$.

The significant feature of the proposed framework is its generality for any unknown data distribution $P_{X,Y}$ and privacy-leakage is computed analytically without relying on the training of black-box models (e.g. adversarial networks [10]) for approximating distributions.

1.3.1 Variational Bayesian Fuzzy Data Modeling

A Takagi-Sugeno fuzzy filter ($\mathcal{F} : \mathbb{R}^q \to \mathbb{R}$) is considered that maps $q-$dimensional real-space to $1-$dimensional real-line. The fuzzy filter consists of M number of rules of following type:

$$\text{If } s \text{ is } \mathbf{A}_m, \text{ then } \mathcal{F}(s) = c_m,\ m \in \{1, \cdots, M\}$$

where $s \in \mathbb{R}^q$, $c_m \in \mathbb{R}$, and the fuzzy set $\mathbf{A}_m$ is defined, without loss of generality, with the following Gaussian membership function

$$\mu_{\mathbf{A}_m}(s) = \exp\left(-0.5 \left\| s - a^m \right\|_W^2\right) \tag{1.2}$$

where $a^m \in \mathbb{R}^q$ is the mean of $\mathbf{A}_m$, $W \in \mathbb{R}^{q\times q}(W > 0)$, and $\|s\|_P^2 \stackrel{\text{def}}{=} s^T P s$. For a given input $s \in \mathbb{R}^q$, the *degree of fulfillment* of the $m-$th rule is given by $\mu_{\mathbf{A}_m}(s)$. The output of the filter to input vector s is computed by taking the weighted average of the output provided by each rule, i.e.,

$$\mathcal{F}(s) = \frac{\sum_{m=1}^M \mu_{\mathbf{A}_m}(s) c_m}{\sum_{m=1}^M \mu_{\mathbf{A}_m}(s)}. \tag{1.3}$$

Definition 1 (A Stochastic Fuzzy Model (FM)) *A stochastic fuzzy model, $\mathcal{G} : \mathbb{R}^p \to \mathbb{R}^n$, maps an input vector $z \in \mathbb{R}^p$ to the output vector $\mathcal{G}(z) \in \mathbb{R}^n$ given as*

$$\mathcal{G}(z) = \left[\, \mathcal{F}_1(V^T z) \ \cdots \ \mathcal{F}_n(V^T z) \,\right]^T \in \mathbb{R}^n \tag{1.4}$$

where $V \in \mathbb{R}^{p\times q}$ (with $q \leq p$) is a matrix, $\mathcal{F}_k$ ($k \in \{1, 2, \cdots, n\}$) is a Takagi-Sugeno fuzzy filter (1.3), with consequent parameters being considered as random variables and being represented by $\alpha_k = \left[\, c_{k,1} \ \cdots \ c_{k,M} \,\right]^T \in \mathbb{R}^M$, such that

$$\mathcal{F}_k(s) = \frac{\sum_{m=1}^M \mu_{\mathbf{A}_m}(s) c_{k,m}}{\sum_{m=1}^M \mu_{\mathbf{A}_m}(s)}. \tag{1.5}$$

Given a finite set of input-output pairs $\mathcal{D} = \left\{(x^i, z^i) \mid i \in \{1, \cdots, N\}\right\}$, the data is modeled through a stochastic fuzzy model $\mathcal{G}$ as

$$x^i = \mathcal{G}(z^i) + v^i \tag{1.6}$$
$$= \left[\, \mathcal{F}_1(V^T z^i) \;\; \cdots \;\; \mathcal{F}_n(V^T z^i) \,\right]^T + v^i. \tag{1.7}$$

The following notation is introduced:

$$\mathbf{z} = \left\{z^i \mid z^i \in \mathbb{R}^p, i \in \{1, \cdots, N\}\right\} \tag{1.8}$$
$$\mathbf{a} = \{a^m \mid a^m \in \mathbb{R}^q, m \in \{1, \cdots, M\}\} \tag{1.9}$$
$$\mathrm{f}_k = \left[\, \mathcal{F}_k(V^T z^1) \;\; \cdots \;\; \mathcal{F}_k(V^T z^N) \,\right]^T \in \mathbb{R}^N \tag{1.10}$$
$$\alpha_k = \left[\, c_{k,1} \;\; \cdots \;\; c_{k,M} \,\right]^T \in \mathbb{R}^M \tag{1.11}$$
$$\mathrm{x}_k = \left[\, x_k^1 \;\; \cdots \;\; x_k^N \,\right]^T \in \mathbb{R}^N \tag{1.12}$$
$$\mathrm{v}_k = \left[\, v_k^1 \;\; \cdots \;\; v_k^N \,\right]^T \in \mathbb{R}^N \tag{1.13}$$

where $k \in \{1, \cdots, n\}$, and x_k^i and v_k^i denote the $k-$th element of x^i and v^i respectively. Let $K_{\mathbf{za}} \in \mathbb{R}^{N \times M}$ be a matrix whose $(i, m)-$th element is given as

$$\left(K_{\mathbf{za}}(V, W)\right)_{i,m} = \frac{\exp\left(-0.5 \left\|V^T z^i - a^m\right\|_W^2\right)}{\sum_{m=1}^{M} \exp\left(-0.5 \left\|z^i - a^m\right\|_W^2\right)}. \tag{1.14}$$

It follows from (1.5), (1.14), and (1.11) that

$$\mathrm{f}_k = K_{\mathbf{za}} \alpha_k. \tag{1.15}$$

Also, it can be observed that

$$\mathrm{x}_k = K_{\mathbf{za}} \alpha_k + \mathrm{v}_k. \tag{1.16}$$

The disturbance vector v_k is priori assumed to be Gaussian with mean zero and a precision of β, i.e.,

$$p(\mathrm{v}_k|\beta) = \left(1/\sqrt{(2\pi)^N (\beta)^{-N}}\right) \exp\left(-0.5\beta \|\mathrm{v}_k\|^2\right) \tag{1.17}$$

where $\beta > 0$ is priori assumed to be Gamma distributed:

$$p(\beta; a, b) = \left(b^a / \Gamma(a)\right) (\beta)^{a-1} \exp(-b\beta) \tag{1.18}$$

where $a, b > 0$. The Gaussian prior is taken over parameter vector α_k:

$$p(\alpha_k; \mathrm{m}_k, \Lambda_k) = \left(1/\sqrt{(2\pi)^M |(\Lambda_k)^{-1}|}\right) \exp\left(-0.5(\alpha_k - \mathrm{m}_k)^T \Lambda_k (\alpha_k - \mathrm{m}_k)\right) \tag{1.19}$$

where $\mathrm{m}_k \in \mathbb{R}^M$ and $\Lambda_k \in \mathbb{R}^{M \times M} (\Lambda_k > 0)$. Define sets

$$\mathbf{X} \stackrel{\text{def}}{=} \{\mathrm{x}_1, \cdots, \mathrm{x}_n\} \tag{1.20}$$
$$\alpha \stackrel{\text{def}}{=} \{\alpha_1, \cdots, \alpha_n\} \tag{1.21}$$

and consider the marginal probability of data $\mathbf{X}$ which is given as

$$p(\mathbf{X}) = \int \mathrm{d}\alpha \, \mathrm{d}\beta \, p(\mathbf{X}, \alpha, \beta). \tag{1.22}$$

Let $q(\alpha, \beta)$ be an arbitrary distribution. The log marginal probability of $\mathbf{X}$ can be expressed as

$$\begin{aligned} \log(p(\mathbf{X})) = & \int \mathrm{d}\alpha \, \mathrm{d}\beta \, q(\alpha, \beta) \log \left(\frac{p(\mathbf{X}, \alpha, \beta)}{q(\alpha, \beta)} \right) \\ & + \int \mathrm{d}\alpha \, \mathrm{d}\beta \, q(\alpha, \beta) \log \left(\frac{q(\alpha, \beta)}{p(\alpha, \beta | \mathbf{X})} \right). \end{aligned} \tag{1.23}$$

Define

$$F(q(\alpha, \beta), \mathbf{X}) \stackrel{\text{def}}{=} \int \mathrm{d}\alpha \, \mathrm{d}\beta \, q(\alpha, \beta) \log \left(p(\mathbf{X}, \alpha, \beta) / q(\alpha, \beta) \right) \tag{1.24}$$

to express (1.23) as

$$\log(p(\mathbf{X})) = F(q(\alpha, \beta), \mathbf{X}) + \mathrm{KL}(q(\alpha, \beta) \| p(\alpha, \beta | \mathbf{X})) \tag{1.25}$$

where KL is the Kullback-Leibler divergence of $p(\alpha, \beta | \mathbf{X})$ from $q(\alpha, \beta)$ and F, referred to as negative free energy, provides a lower bound on the the logarithmic evidence for the data.

The variational Bayesian approach minimizes the difference (in term of KL divergence) between variational and true posteriors via analytically maximizing negative free energy F over variational distributions. However, the analytical derivation requires the following widely used mean-field approximation:

$$q(\alpha, \beta) = q(\alpha) q(\beta) \tag{1.26}$$

$$= q(\alpha_1) \cdots q(\alpha_n) q(\beta). \tag{1.27}$$

Applying the standard variational optimization technique (as in [12–16]), it can be verified that the optimal variational distributions maximizing F are as follows:

$$q^*(\alpha_k) = \left(1 / \sqrt{(2\pi)^M |(\hat{\Lambda}_k)^{-1}|} \right) \exp \left(-0.5 (\alpha_k - \hat{\mathrm{m}}_k)^T \hat{\Lambda}_k (\alpha_k - \hat{\mathrm{m}}_k) \right) \tag{1.28}$$

$$q^*(\beta) = \left((\hat{b})^{\hat{a}} / \Gamma(\hat{a}) \right) (\beta)^{\hat{a}-1} \exp(-\hat{b}\beta) \tag{1.29}$$

where the parameters $(\hat{\Lambda}_k, \hat{\mathrm{m}}_k, \hat{a}, \hat{b})$ satisfy the following:

$$\hat{\Lambda}_k = \Lambda_k + \left(\hat{a} / \hat{b} \right) (K_{\mathbf{za}})^T K_{\mathbf{za}} \tag{1.30}$$

$$\hat{\mathrm{m}}_k = (\hat{\Lambda}_k)^{-1} \left(\Lambda_k \mathrm{m}_k + \left(\hat{a} / \hat{b} \right) (K_{\mathbf{za}})^T \mathrm{x}_k \right) \tag{1.31}$$

$$\hat{a} = a + 0.5nN \tag{1.32}$$

$$\hat{b} = b + 0.5 \sum_{k=1}^{n} \left\{ \| \mathrm{x}_k - K_{\mathbf{za}} \hat{\mathrm{m}}_k \|^2 + Tr \left((\hat{\Lambda}_k)^{-1} (K_{\mathbf{za}})^T K_{\mathbf{za}} \right) \right\} \tag{1.33}$$

Algorithm 1 An algorithm for variational Bayesian inference of data model.

Require: Data set $\mathcal{D} = \{(x^i, z^i) \mid x^i \in \mathbb{R}^n, z^i \in \mathbb{R}^p, i \in \{1, \cdots, N\}\}$, the number of rules in a fuzzy filter $M \in \mathbb{Z}_+$, the subspace dimension $q \in \mathbb{Z}_+$ with $q \leq p$.

1: Define $V \in \mathbb{R}^{p \times q}$ such that $j-$th column of V is equal to eigenvector corresponding to $j-$th largest eigenvalue of sample covariance matrix of $\{z^i \mid i \in \{1, \cdots, N\}\}$.
2: Compute $s^i = V^T z^i$, for $i \in \{1, \cdots, N\}$.
3: The fuzzy sets' mean values, $\mathbf{a} = \{a^m \mid m \in \{1, \cdots, M\}\}$, are defined as

$$\{a^m \mid m \in \{1, \cdots, M\}\} = \text{ClusterCentroid}(\{s^i \mid i \in \{1, \cdots, N\}\}, M) \tag{1.34}$$

ClusterCentroid$(\cdot)$ represents the k-means clustering to return M cluster centroids.
4: Define W to be a diagonal matrix such that $j-$th diagonal element is equal to the inverse of squared-distance between two most-distant points in the set $\{s^i_j \mid i \in \{1, \cdots, N\}\}$, where s^i_j is $j-$th element of s^i.
5: Compute $K_{\mathbf{za}}(V, W)$ using (1.14).
6: Choose $a = 10^{-6}, b = 10^{-6}, \mathrm{m}_j = 0, \Lambda_j = 10^{-6} I$.
7: Initialise $\hat{a}/\hat{b} = 1$.
8: **repeat**
9: Update $\{\hat{\Lambda}_k, \hat{\mathrm{m}}_k \mid k \in \{1, \cdots, n\}\}, \hat{a}, \hat{b}$ using (1.30), (1.31), (1.32), (1.33).
10: **until** (convergence **or** iterations = 1000)
11: **return** $\mathcal{M} = \{\hat{a}, \hat{b}, \{\hat{\mathrm{m}}_k, \hat{\Lambda}_k \mid k \in \{1, \cdots, n\}\}, \mathbf{a}, V, W\}$.

where Tr$(\cdot)$ denotes the trace operator.

Variational Bayesian inference lends itself to a data modeling algorithm formally stated as Algorithm 1. The optimal distributions $q^*(\alpha_k)$ and $q^*(\beta)$ determined using Algorithm 1 define a model as stated in Remark 1.

Remark 1 (Model) *The model built using Algorithm 1 relates sensitive data vector* $x = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^T \in \mathbb{R}^n$ *to released data vector* $z \in \mathbb{R}^p$ *as*

$$x_k = k(z)\alpha_k + v_k, \tag{1.35}$$

$$p(v_k|\beta) = \left(1/\sqrt{(2\pi)(\beta)^{-1}}\right) \exp\left(-0.5\beta|v_k|^2\right), \tag{1.36}$$

$$p(\beta; \hat{a}, \hat{b}) = \left(\hat{b}^{\hat{a}}/\Gamma(\hat{a})\right) (\beta)^{\hat{a}-1} \exp(-\hat{b}\beta), \tag{1.37}$$

$$p(\alpha_k; \hat{\mathrm{m}}_k, \hat{\Lambda}_k) = \left(1/\sqrt{(2\pi)^M |(\Lambda_k)^{-1}|}\right) \exp\left(-0.5(\alpha_k - \hat{\mathrm{m}}_k)^T \hat{\Lambda}_k (\alpha_k - \hat{\mathrm{m}}_k)\right) \tag{1.38}$$

where $k(z) \in \mathbb{R}^{1 \times M}$ *is a vector-valued function whose* $m-$*th element is given as*

$$(k(z))_m = \frac{\exp\left(-0.5 \left\|V^T z - a^m\right\|^2_W\right)}{\sum_{m=1}^{M} \exp\left(-0.5 \left\|V^T z - a^m\right\|^2_W\right)}. \tag{1.39}$$

Here, $\{\hat{a}, \hat{b}, \{\hat{\mathrm{m}}_k, \hat{\Lambda}_k \mid k \in \{1, \cdots, n\}\}, \mathbf{a}, V, W\}$ *are returned by Algorithm 1.*

1.3.2 Variational Approximation of Informational Privacy

The mutual information between sensitive data vector x and released data vector z is given as

$$I(x;z) = H(x) - H(x|z) \tag{1.40}$$
$$= H(x) + \int_{\mathcal{X},\mathcal{Z}} p(x,z)\log\left(p(x|z)\right)\mathrm{d}x\,\mathrm{d}z \tag{1.41}$$
$$= H(x) + \left\langle \log\left(p(x|z)\right)\right\rangle_{p(x,z)} \tag{1.42}$$

where $H(x)$, $H(x|z)$ are marginal, conditional entropies respectively and the averaging operator $< \cdot >_{\cdot}$ is defined as

$$< f(x) >_{p(x)} = \int \mathrm{d}x\, p(x) f(x). \tag{1.43}$$

Result 1 (Variational Approximation of $I(x;z)$) *Assuming the data model as stated in Remark 1, a variational approximation of $I(x;z)$ is given as*

$$\hat{I}(x;z) = \tag{1.44}$$
$$H(x) - 0.5n\log(2\pi) + 0.5n\left\{\Psi(\bar{a}) - \log(\bar{b})\right\}$$
$$- \frac{1}{2}\left(\bar{a}/\bar{b}\right)\sum_{k=1}^{n}\left\langle |x_k - k(z)\bar{\mathrm{m}}_k|^2\right\rangle_{p(x,z)} - \frac{1}{2}\left(\bar{a}/\bar{b}\right)\sum_{k=1}^{n}\left\langle Tr\left((\bar{\Lambda}_k)^{-1}(k(z))^T k(z)\right)\right\rangle_{p(z)}$$
$$- 0.5\sum_{k=1}^{n}\left\{(\hat{\mathrm{m}}_k - \bar{\mathrm{m}}_k)^T\hat{\Lambda}_k(\hat{\mathrm{m}}_k - \bar{\mathrm{m}}_k) + Tr\left(\hat{\Lambda}_k(\bar{\Lambda}_k)^{-1}\right) - \log\left(\frac{|(\bar{\Lambda}_k)^{-1}|}{|(\hat{\Lambda}_k)^{-1}|}\right)\right\}$$
$$+ 0.5nM - \hat{a}\log\left(\bar{b}/\hat{b}\right) + \log\left(\Gamma(\bar{a})/\Gamma(\hat{a})\right) - (\bar{a}-\hat{a})\Psi(\bar{a}) + (\bar{b}-\hat{b})\left(\bar{a}/\bar{b}\right)$$

where $\Psi(\cdot)$ is the digamma function and the parameters $(\bar{\Lambda}_k, \bar{\mathrm{m}}_k, \bar{a}, \bar{b})$ satisfy followings:

$$\bar{\Lambda}_k = \hat{\Lambda}_k + \left(\bar{a}/\bar{b}\right)\left\langle (k(z))^T k(z)\right\rangle_{p(z)} \tag{1.45}$$
$$\bar{\mathrm{m}}_k = (\bar{\Lambda}_k)^{-1}\left(\hat{\Lambda}_k\hat{\mathrm{m}}_k + \left(\bar{a}/\bar{b}\right)\left\langle (k(z))^T x_k\right\rangle_{p(x,z)}\right) \tag{1.46}$$
$$\bar{a} = \hat{a} + 0.5n \tag{1.47}$$
$$\bar{b} = \hat{b} + \frac{1}{2}\sum_{k=1}^{n}\left\langle |x_k - k(z)\bar{\mathrm{m}}_k|^2\right\rangle_{p(x,z)} + \frac{1}{2}\sum_{k=1}^{n}\left\langle Tr\left((\bar{\Lambda}_k)^{-1}(k(z))^T k(z)\right)\right\rangle_{p(z)} \tag{1.48}$$

Proof: Consider the conditional probability of x which is given as

$$p(x|z) = \int \mathrm{d}\alpha\,\mathrm{d}\beta\, p(\alpha,\beta,x|z) \tag{1.49}$$

where α is a set defined as in (1.21). Let $q(\alpha, \beta)$ be an arbitrary distribution. The log conditional probability of x can be expressed as

$$\log(p(x|z)) = \int \mathrm{d}\alpha \, \mathrm{d}\beta \; q(\alpha, \beta) \log\left(p(x|z)\right) \tag{1.50}$$

$$= \int \mathrm{d}\alpha \, \mathrm{d}\beta \; q(\alpha, \beta) \log\left(p(\alpha, \beta, x|z)/p(\alpha, \beta|x, z)\right) \tag{1.51}$$

$$= \int \mathrm{d}\alpha \, \mathrm{d}\beta \; q(\alpha, \beta) \log\left(p(\alpha, \beta, x|z)/q(\alpha, \beta)\right) + \int \mathrm{d}\alpha \, \mathrm{d}\beta \; q(\alpha, \beta) \log\left(q(\alpha, \beta)/p(\alpha, \beta|x, z)\right). \tag{1.52}$$

Define

$$F(q(\alpha, \beta), x, z) \stackrel{\text{def}}{=} \int \mathrm{d}\alpha \, \mathrm{d}\beta \; q(\alpha, \beta) \log\left(p(\alpha, \beta, x|z)/q(\alpha, \beta)\right) \tag{1.53}$$

to express (1.52) as

$$\log(p(x|z)) = F(q(\alpha, \beta), x, z) + \mathrm{KL}(q(\alpha, \beta) \| p(\alpha, \beta|x, z)) \tag{1.54}$$

where KL is Kullback-Leibler divergence of $p(\alpha, \beta|x, z)$ from $q(\alpha, \beta)$. Using (1.42),

$$I(x; z) = H(x) + \langle F(q(\alpha, \beta), x, z) \rangle_{p(x,z)} + \langle \mathrm{KL}(q(\alpha, \beta) \| p(\alpha, \beta|x, z)) \rangle_{p(x,z)}. \tag{1.55}$$

Since KullbackâĂŞLeibler divergence is always non-zero, it follows from (1.55) that $H(x) + \langle F \rangle_{p(x,z)}$ provides a lower bound on $I(x; z)$ i.e.

$$I(x; z) \geq H(x) + \langle F(q(\alpha, \beta), x, z) \rangle_{p(x,z)}. \tag{1.56}$$

Our approach to approximate $I(x; z)$ is to maximize its lower bound with respect to variational distribution $q(\alpha, \beta)$. That is, we solve

$$\hat{I}(x; z) = \max_{q(\alpha,\beta)} \left(H(x) + \langle F(q(\alpha, \beta), x, z) \rangle_{p(x,z)} \right) \tag{1.57}$$

$$= H(x) + \max_{q(\alpha,\beta)} \langle F(q(\alpha, \beta), x, z) \rangle_{p(x,z)}. \tag{1.58}$$

For this, consider

$$F(q(\alpha, \beta), x, z) = \langle \log(p(x|\alpha, \beta, z)) \rangle_{q(\alpha,\beta)} + \langle \log\left(p(\alpha, \beta)/q(\alpha, \beta)\right) \rangle_{q(\alpha,\beta)} \tag{1.59}$$

Assuming that $x_1, \cdots, x_n$ are independent,

$$\log(p(x|z, \alpha, \beta)) = \sum_{k=1}^{n} \log(p(x_k|z, \alpha_k, \beta)). \tag{1.60}$$

It follows from (1.36) and (1.35) that

$$\log(p(x_k|z, \alpha_k, \beta)) = -0.5 \log(2\pi) + 0.5 \log(\beta) - 0.5\beta |x_k - k(z)\alpha_k|^2 \tag{1.61}$$

Using (1.60), (1.61), and (1.26-1.27) in (1.59), we have

$$F = -0.5n\log(2\pi) + 0.5n\left\langle \log(\beta)\right\rangle_{q(\beta)} - 0.5\left\langle \beta\right\rangle_{q(\beta)} \sum_{k=1}^{n} \left\langle |x_k - k(z)\alpha_k|^2 \right\rangle_{q(\alpha_k)} + \sum_{k=1}^{n} \left\langle \log\left(p(\alpha_k;\hat{\mathrm{m}}_k,\hat{\Lambda}_k)/q(\alpha_k)\right)\right\rangle_{q(\alpha_k)} + \left\langle \log\left(p(\beta;\hat{a},\hat{b})/q(\beta)\right)\right\rangle_{q(\beta)}. \quad (1.62)$$

Thus,

$$\begin{aligned}\left\langle F\right\rangle_{p(x,z)} = & -0.5n\log(2\pi) + 0.5n\left\langle \log(\beta)\right\rangle_{q(\beta)} - 0.5\left\langle \beta\right\rangle_{q(\beta)} \sum_{k=1}^{n} \left\langle |x_k|^2 \right\rangle_{p(x)} \\ & - 0.5\left\langle \beta\right\rangle_{q(\beta)} \sum_{k=1}^{n} \left\langle (\alpha_k)^T \left\langle (k(z))^T k(z)\right\rangle_{p(z)} \alpha_k \right\rangle_{q(\alpha_k)} \\ & + \left\langle \beta\right\rangle_{q(\beta)} \sum_{k=1}^{n} \left\langle (\alpha_k)^T \left\langle (k(z))^T x_k\right\rangle_{p(x,z)} \right\rangle_{q(\alpha_k)} \\ & + \sum_{k=1}^{n} \left\langle \log\left(p(\alpha_k;\hat{\mathrm{m}}_k,\hat{\Lambda}_k)/q(\alpha_k)\right)\right\rangle_{q(\alpha_k)} \\ & + \left\langle \log\left(p(\beta;\hat{a},\hat{b})/q(\beta)\right)\right\rangle_{q(\beta)}. \end{aligned} \quad (1.63)$$

Now, $\left\langle F\right\rangle_{p(x,z)}$ can be maximized w.r.t. $q(\alpha_k)$ and $q(\beta)$ using variational optimization. It can be seen that optimal distributions maximizing $\left\langle F\right\rangle_{p(x,z)}$ are given as

$$q^*(\alpha_k) = \left(1/\sqrt{(2\pi)^M |(\bar{\Lambda}_k)^{-1}|}\right) \exp\left(-0.5(\alpha_k - \bar{\mathrm{m}}_k)^T \bar{\Lambda}_k (\alpha_k - \bar{\mathrm{m}}_k)\right) \quad (1.64)$$

$$q^*(\beta) = \left((\bar{b})^{\bar{a}}/\Gamma(\bar{a})\right)(\beta)^{\bar{a}-1}\exp(-\bar{b}\beta) \quad (1.65)$$

where the parameters $(\bar{\Lambda}_j^l, \bar{\mathrm{m}}_j^l, \bar{a}_l, \bar{b}_l)$ satisfy (1.45), (1.46), (1.47), (1.48). The maximum attained value of $\left\langle F\right\rangle_{p(x,z)}$ is given as

$$\begin{aligned} & \left\langle F(q^*(\alpha,\beta),x,z)\right\rangle_{p(x,z)} = \quad (1.66) \\ & -0.5n\log(2\pi) + 0.5n\left\{\Psi(\bar{a}) - \log(\bar{b})\right\} - 0.5\left(\bar{a}/\bar{b}\right)\sum_{k=1}^{n}\left\langle |x_k - k(z)\bar{\mathrm{m}}_k|^2\right\rangle_{p(x,z)} \\ & -0.5\left(\bar{a}/\bar{b}\right)\sum_{k=1}^{n}\left\langle Tr\left((\bar{\Lambda}_k)^{-1}(k(z))^T k(z)\right)\right\rangle_{p(z)} - \sum_{k=1}^{n}\mathrm{KL}(q^*(\alpha_k)\|p(\alpha_k;\hat{\mathrm{m}}_k,\hat{\Lambda}_k)) \\ & -\mathrm{KL}(q^*(\beta)\|p(\beta;\hat{a},\hat{b})) \end{aligned}$$

where $\Psi(\cdot)$ is the digamma function. After substituting the maximum value of $\left\langle F\right\rangle_{p(x,z)}$ in (1.58) and calculating Kullback-Leibler divergences, we get (1.44). ■

2. Privacy-Preserving Semi-Supervised Transfer and Multi-Task Approaches (T3.2)

The availability of high quality labelled data is crucial for the success of machine learning methods. While a single entity may not own massive amount of data, a collaboration among data-owners regarding sharing of knowledge extracted locally from their private data can be beneficial. The data privacy concerns and the legal requirements may not allow a centralization of the data from multiple sources. Thus, an interest in privacy-preserving machine learning with distributed training datasets arises. We consider the privacy-preserving distributed machine learning problem under a scenario that the knowledge extracted from a labelled training dataset (referred to as "source domain") is intended to improve the learning of a classifier trained using a dataset with both unlabelled and very few labelled samples (referred to as "target domain") such that source and target domains are allowed to be heterogeneous. That is, source and target data samples are allowed to differ in their dimensions and no assumptions are made regarding statistical distributions of source and target data. The problem of "privacy-preserving semi-supervised transfer learning" has previously been addressed in the literature from different prospectives. We focus on the development of a method able to simultaneously deal with high-dimensional data and heterogeneous domains.

2.1 Background

A lot of research has been carried out in the area of transfer learning. The heterogeneous data from source and target domain (i.e. source and target domains have different feature space and dimensions) can be transformed to a common subspace by using two different projection matrices. Existing supervised learning methods (e.g., SVM) can be then employed to learn the projection matrices and the target domain classifier [17]. It is possible to learn a transformation that maps feature points from one domain to another using cross-domain constraints formed by requiring that the transformation maps points from the same category (but different domain) near each other [18]. A study [19] learns projections from each domain

to a latent space via simultaneously minimizing a notion of domain variance while maximizing a measure of discriminatory power where Riemannian optimization techniques are used to match statistical properties between samples projected into the latent space from different domains. Another study [20] proposes a regularized unsupervised optimal transportation model to perform the alignment of the representations in the source and target domains. The method in [21] uses geodesic flow to construct an infinite-dimensional feature space that assembles information on the source domain, on the target domain, and on "phantom" domains interpolating between source and target domains. Inner products in infinite-dimensional feature space give rise to a kernel function facilitating the construction of any kernelized classifiers. Another approach is of an adaptation of source model to the target domain via iteratively deleting source-domain samples and adapting the model gradually to the target-domain instances [22]. Boosting-based learning algorithms can be also used to adaptively assign the training weights to source and target samples based on their relevance in the training of the classifier [23]. Bayesian learning can be a framework to study transfer learning through modeling of a joint prior probability density function for feature-label distributions of the source and target domains [24]. Deep learning framework is another promising research direction explored for transfer learning [25–27].

Federated learning techniques allow multiple decentralized actors to build a common robust machine learning model without sharing their data. However, the local datasets may contain sensitive information that need to be protected from *model inversion* attack [28] and from adversaries with an access to model parameters and knowledge of the training procedure. This goal has been addressed within the framework of differential privacy [29,30]. Differential Privacy [31,32] is a formalism to quantify the degree to which the privacy for each individual in the dataset is preserved while releasing the output of a data analysis algorithm. Differential privacy provides a guarantee that an adversary, by virtue of presence or absence of an individual's data in the dataset, would not be able to draw any conclusions about an individual from the released output of the analysis algorithm. This guarantee is achieved by means of a randomization of the data analysis process. In the context of machine learning, randomization is carried out via either adding random noise to the input or output of the machine learning algorithm or modifying the machine learning algorithm itself. A limited number of studies exist on differentially private semi-supervised transfer learning. The authors in [33] suggest an importance weighting mechanism to preserve the differential privacy of a private dataset via computing and releasing a weight for each record in an existing public dataset such that computations on public dataset with weights is approximately equivalent to computations on private dataset. The importance weighting mechanism is adapted in [34] to determine the weight of a source hypothesis in the process of constructing informative Bayesian prior for logistic regression based target model. This method does not allow transferring knowledge across heterogeneous domains and is limited to binary logistic regression. [35] introduces "private aggregation of teacher ensembles" approach where an ensemble of "teacher" models is trained on

disjoint subsets of the sensitive data and a “student” model learns to predict an output chosen by noisy voting among all of the teachers. However, this method doesn't consider the heterogeneous domains. Another approach [36–38] is to construct a differentially private unsupervised generative model for generating a synthetic version of the private data, and then releases the synthetic data for a non-private learning. This technique is capable of effectively handling high-dimensional data in differential privacy setting, however, can not handle heterogeneous domains. The study in [39] uses a large public dataset to learn a dimension-reducing representation mapping which is then applied on private data to obtain a low-dimensional representation of the private data followed by the learning of a differentially private predictor. Again, this method is not capable of handling heterogeneous domains.

2.2 Requirements

There is a need of developing a “differentially private semi-supervised transfer learning" framework that

R1: is capable of handling high-dimensional data and heterogeneity of domains;

R2: optimizes the differential private noise adding mechanism such that for a given level of privacy, the perturbation in the data is as small as possible;

R3: facilitates a transfer of knowledge from source to target domain without requiring the availability of source domain private training data;

R4: allows employing deep models in source and target domains so that data features at different abstraction levels can be used to transfer knowledge across domains, however, without the requirement of the availability of large amount of data;

R5: provides a robustness of target model towards the perturbations in source data caused by the privacy requirements demanded by source data owner.

To the best knowledge of authors, there does not exist any study in the literature addressing sufficiently simultaneously all of the aforementioned five requirements (i.e. R1-R5). Thus, we present in this study a novel approach to differentially private semi-supervised transfer learning that fulfills all of the requirements.

2.3 Proposed Methodology

The basic idea of the proposed approach is stated in Fig. 2.1. The salient features of our approach are following:

- An optimal differentially private noise adding mechanism is used to perturb the source dataset for preserving its privacy.

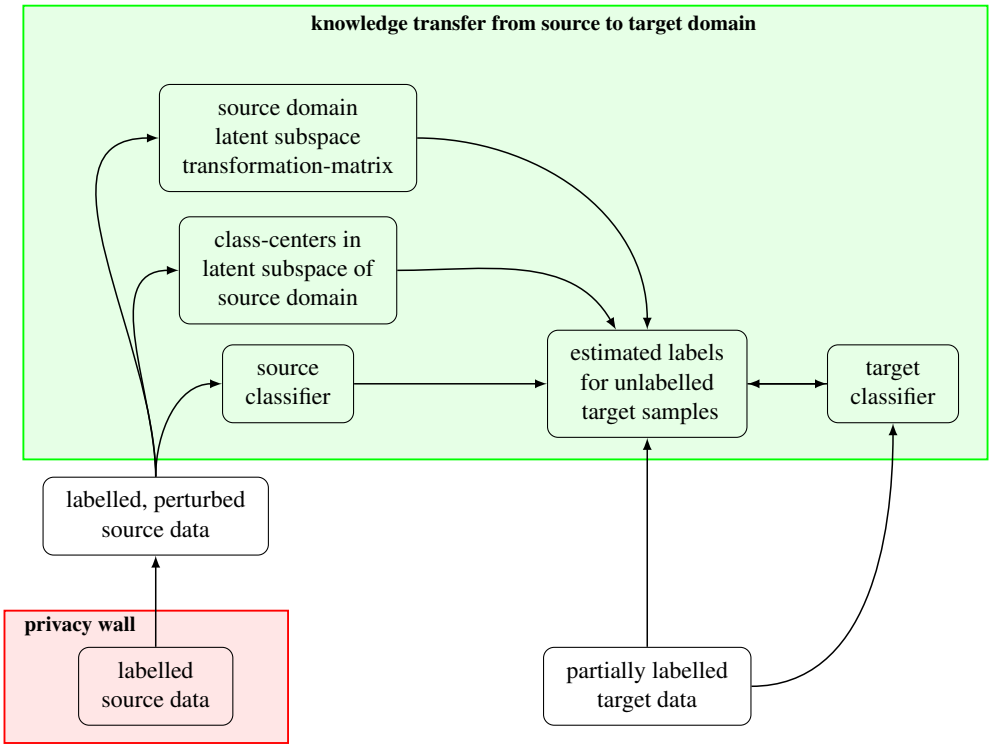

Figure 2.1: The proposed approach to privacy-preserving semi-supervised transfer learning.

- Both source and target classifiers consist of autoencoders based compositions. A multi-class classifier is presented that employs a wide-deep autoencoder for each class to learn data representation. A wide-deep autoencoder consists of a parallel composition of deep autoencoders and each deep autoencoder consists of a nested composition of mappings. An analytical approach is presented for the learning of the deep autoencoder.

- Since differential privacy will remain immune to any post-processing of noise added data samples, the perturbed source dataset is used to

  - build a differentially private source domain classifier,
  - compute a differentially private source domain latent subspace transformation-matrix,
  - define differentially private class-centers in latent subspace of source domain.

- The class-labels for unlabelled target data samples are estimated via

  - representing target data samples in source-data-space using a transformation that maps a target sample close to center of $c-$th labelled target data samples to a point in source-data-space that is close to center of $c-$th labelled source data samples,
  - combining source and target domain classifiers for predicting class-labels.

- The target domain classifier is learned adaptively in a manner that higher-level data features are used during initial iterations for updating the classifier parameters and as the number of iterations increases more and more lower-level data features are intended to be included in the process of updating the classifier parameters.

3. Verification of Differential Privacy Deep Learning Models (T3.4)

The important part of the work package is to verify the developed framework for differentially private distributed deep learning. Theoretical guarantees have already been provided in D3.1 and in Chapter 1: generated noise distribution satisfies the sufficient conditions for $(\epsilon, \delta)-$differential privacy. However the theoretical guarantees cannot prevent bugs and vulnerabilities in the developed software. Within this task we need to ensure the correctness of the implementation and its integration in the Serums Smart Health Center system.

3.1 Proposed Methodology

In this task we propose two approaches for checking correctness of the developed software. The first part will check the differential privacy preservation by the software component while in the second part will validate the lack of vulnerabilities during the interactions between the privacy-preserving machine learning component and the rest of the Serums Smart Health Center system.

Checking Differential Privacy Preservation

For the component implementation we are going to use a black-box testing checking the satisfaction of $(\epsilon, \delta)-$differential privacy based on the ideas from adversarial machine learning [40]. One of the commonly used approaches is based on checking the presence of adversarial perturbations that are (minimal) changes to the input data causing misclassifications. The definition of differential privacy allows an adaptation of these techniques: search for the adversarial perturbation that would break $(\epsilon, \delta)-$differential privacy guarantees. There exists many works on automated search of perturbations, e.g. [41–46]. We would employ the approach based on a list of transformations (e.g. [45,46]) that could be applied to the original input.

One key component to this part is the available of suitable data. Since using

real medical data is very complex and has many ethical and privacy concerns, in the Serums project we use fabricated data (for details see Work Package 4 and Deliverables D) instead since this allows for large amounts of data to be generated that can be used for testing and experiments. Here this fabricated data will be used in the experiments, and also the use of fabricated data allows us to perform many different experiments with small (or differential) peturbations of the inputs to test the outputs upon directly. This allows for results to be validated as behaving in a correct and expected manner, as well as the above aspects using adversarial machine learning.

Overall, the methodology here will include both targetted attempts using adversarial machine learning to find edge cases, as well as broader black-box testing of the implementation to ensure correct and expected behavior.

Vulnerability and Integration Analysis

The necessity of verification of machine learning on the global system level is argued in [47]. For the verification of the integration we are building a model in a formal language incorporating all components and their interactions. The model is being built in Uppaal tool[1] that provides a modelling language and a model checker and which have been used in multiple projects, e.g. [48, 49]. Formal model of the complete system would allow us to check that the behaviour meets the specification. The model will also be used in verification activities of other work packages (WP5, WP6). Within this task we will focus on interactions with the privacy preserving distributed deep learning component. In particular, we will validate that all communications with the component are following the specifications, absence of unspecified interactions and the contents of the data flow inside and outside of the components. Attack models would be used to explore how to violate the correct behavior of the system and to find potential vulnerabilities such as in [50]. An example of property to be verified is impossibility to impersonate the distributed deep learning component in order to receive private data.

[1] http://www.uppaal.org/

Bibliography

[1] M. Kumar, M. Rossbory, B. A. Moser, and B. Freudenthaler, "Deriving an optimal noise adding mechanism for privacy-preserving machine learning," in *Database and Expert Systems Applications*, G. Anderst-Kotsis, A. M. Tjoa, I. Khalil, M. Elloumi, A. Mashkoor, J. Sametinger, X. Larrucea, A. Fensel, J. Martinez-Gil, B. Moser, C. Seifert, B. Stein, and M. Granitzer, Eds. Cham: Springer International Publishing, 2019, pp. 108–118.

[2] D. Kifer and A. Machanavajjhala, "No free lunch in data privacy," in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD'11. New York, NY, USA: Association for Computing Machinery, 2011, pp. 193–204. [Online]. Available: https://doi.org/10.1145/1989323.1989345

[3] F. D. P. Calmon and N. Fawaz, "Privacy against statistical inference," in *in Proceedings of the 50th Annual Allerton Conference on Communication, Control, and Computing, Allerton 2012, 2012. [Online]. Available: http://arxiv.org/abs/1210.2123*.

[4] C. Liu, S. Chakraborty, and P. Mittal, "Dependence makes you vulnberable: Differential privacy under dependent tuples," in *23rd Annual Network and Distributed System Security Symposium, NDSS 2016, San Diego, California, USA, February 21-24, 2016*. The Internet Society, 2016. [Online]. Available: http://wp.internetsociety.org/ndss/wp-content/uploads/sites/25/2017/09/dependence-makes-you-vulnerable-differential-privacy-under-dependent-tuples.pdf

[5] D. Rebollo-Monedero, J. Forné, and J. Domingo-Ferrer, "From t-closeness-like privacy to postrandomization via information theory," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 11, pp. 1623–1636, Nov 2010.

[6] L. Sankar, S. R. Rajagopalan, and H. V. Poor, "Utility-privacy tradeoffs in databases: An information-theoretic approach," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 6, pp. 838–852, June 2013.

[7] Y. O. Basciftci, Y. Wang, and P. Ishwar, "On privacy-utility tradeoffs for constrained data release mechanisms," in *2016 Information Theory and Applications Workshop (ITA)*, Jan 2016, pp. 1–6.

[8] Y. Wang, Y. O. Basciftci, and P. Ishwar, "Privacy-utility tradeoffs under constrained data release mechanisms," *CoRR*, vol. abs/1710.09295, 2017. [Online]. Available: http://arxiv.org/abs/1710.09295

[9] C. Huang, P. Kairouz, X. Chen, L. Sankar, and R. Rajagopal, "Context-aware generative adversarial privacy," *Entropy*, vol. 19, no. 12, p. 656, Dec 2017. [Online]. Available: http://dx.doi.org/10.3390/e19120656

[10] A. Tripathy, Y. Wang, and P. Ishwar, "Privacy-preserving adversarial networks," in *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Sep. 2019, pp. 495–505.

[11] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 2172–2180.

[12] M. Kumar, N. Stoll, and R. Stoll, "Variational bayes for a mixed stochastic/deterministic fuzzy filter," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 4, pp. 787–801, Aug 2010.

[13] M. Kumar, N. Stoll, R. Stoll, and K. Thurow, "A Stochastic Framework for Robust Fuzzy Filtering and Analysis of Signals-Part I," *IEEE Transactions on Cybernetics*, vol. 46, no. 5, pp. 1118–1131, May 2016.

[14] M. Kumar, N. Stoll, and R. Stoll, "Stationary Fuzzy Fokker-Planck Learning and Stochastic Fuzzy Filtering," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 5, pp. 873–889, Oct 2011.

[15] M. Kumar, S. Neubert, S. Behrendt, A. Rieger, M. Weippert, N. Stoll, K. Thurow, and R. Stoll, "Stress monitoring based on stochastic fuzzy analysis of heartbeat intervals," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 4, pp. 746–759, Aug 2012.

[16] M. Kumar, A. Insan, N. Stoll, K. Thurow, and R. Stoll, "Stochastic fuzzy modeling for ear imaging based child identification," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 9, pp. 1265–1278, Sep. 2016.

[17] W. Li, L. Duan, D. Xu, and I. W. Tsang, "Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 6, pp. 1134–1148, 2014.

[18] J. Hoffman, E. Rodner, J. Donahue, B. Kulis, and K. Saenko, "Asymmetric and category invariant feature transformations for domain adaptation," *International Journal of Computer Vision*, vol. 109, no. 1, pp. 28–41, 2014.

[19] S. Herath, M. Harandi, and F. Porikli, "Learning an invariant hilbert space for domain adaptation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[20] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, "Optimal transport for domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 9, pp. 1853–1865, 2017.

[21] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2066–2073.

[22] L. Bruzzone and M. Marconcini, "Domain adaptation problems: A dasvm classification technique and a circular validation strategy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 770–787, May 2010.

[23] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proceedings of the 24th International Conference on Machine Learning*, ser. ICML'07. New York, NY, USA: Association for Computing Machinery, 2007, pp. 193–200.

[24] A. Karbalayghareh, X. Qian, and E. R. Dougherty, "Optimal bayesian transfer learning," *IEEE Transactions on Signal Processing*, vol. 66, no. 14, pp. 3724–3739, 2018.

[25] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 97–105.

[26] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. Red Hook, NY, USA: Curran Associates Inc., 2016, pp. 136–144.

[27] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2096–2030, Jan. 2016.

[28] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications*

Security, ser. CCS '15. New York, NY, USA: ACM, 2015, pp. 1322–1333. [Online]. Available: http://doi.acm.org/10.1145/2810103.2813677

[29] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: ACM, 2016, pp. 308–318. [Online]. Available: http://doi.acm.org/10.1145/2976749.2978318

[30] N. Phan, Y. Wang, X. Wu, and D. Dou, "Differential privacy preservation for deep auto-encoders: An application of human behavior prediction," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, ser. AAAI'16. AAAI Press, 2016, pp. 1309–1316. [Online]. Available: http://dl.acm.org/citation.cfm?id=3015812.3016005

[31] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Advances in Cryptology - EUROCRYPT 2006*, S. Vaudenay, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 486–503.

[32] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211–407, 2014. [Online]. Available: https://doi.org/10.1561/0400000042

[33] Z. Ji and C. Elkan, "Differential privacy based on importance weighting," *Machine Learning*, vol. 93, no. 1, pp. 163–183, 2013.

[34] Y. Wang, Q. Gu, and D. E. Brown, "Differentially private hypothesis transfer learning," in *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2018, Dublin, Ireland, September 10-14, 2018, Proceedings, Part II*, ser. Lecture Notes in Computer Science, M. Berlingerio, F. Bonchi, T. Gärtner, N. Hurley, and G. Ifrim, Eds., vol. 11052. Springer, 2018, pp. 811–826.

[35] N. Papernot, M. Abadi, U. Erlingsson, I. J. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data." in *ICLR*. OpenReview.net, 2017. [Online]. Available: http://dblp.uni-trier.de/db/conf/iclr/iclr2017.html#PapernotAEGT17

[36] G. Acs, L. Melis, C. Castelluccia, and E. De Cristofaro, "Differentially private mixture of generative neural networks," in *2017 IEEE International Conference on Data Mining (ICDM)*, 2017, pp. 715–720.

[37] L. Xie, K. Lin, S. Wang, F. Wang, and J. Zhou, "Differentially private generative adversarial network," *ArXiv*, vol. abs/1802.06739, 2018.

[38] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao, "Privbayes: Private data release via bayesian networks," *ACM Trans. Database Syst.*, vol. 42, no. 4, Oct. 2017.

[39] T. Niinimäki, M. A. Heikkilä, A. Honkela, and S. Kaski, "Representation transfer for differentially private drug sensitivity prediction," *Bioinformatics*, vol. 35, no. 14, pp. i218–i224, 07 2019.

[40] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar, "Adversarial machine learning," in *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, 2011, pp. 43–58.

[41] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 427–436.

[42] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE, 2016, pp. 372–387.

[43] K. Pei, Y. Cao, J. Yang, and S. Jana, "Towards practical verification of machine learning: The case of computer vision systems," *arXiv preprint arXiv:1712.01785*, 2017.

[44] C.-H. Cheng, G. Nührenberg, and H. Ruess, "Maximum resilience of artificial neural networks," in *International Symposium on Automated Technology for Verification and Analysis*. Springer, 2017, pp. 251–268.

[45] X. Huang, M. Kwiatkowska, S. Wang, and M. Wu, "Safety verification of deep neural networks," in *International Conference on Computer Aided Verification*. Springer, 2017, pp. 3–29.

[46] M. Wicker, X. Huang, and M. Kwiatkowska, "Feature-guided black-box safety testing of deep neural networks," in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2018, pp. 408–426.

[47] T. Dreossi, S. Jha, and S. A. Seshia, "Semantic adversarial deep learning," in *International Conference on Computer Aided Verification*. Springer, 2018, pp. 3–26.

[48] F. Mercaldo, F. Martinelli, and A. Santone, "Real-time scada attack detection by means of formal methods," in *2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*. IEEE, 2019, pp. 231–236.

[49] D. Basile, M. H. ter Beek, and A. Legay, "Strategy synthesis for autonomous driving in a moving block railway system with uppaal stratego," in *International Conference on Formal Techniques for Distributed Objects, Components, and Systems*. Springer, 2020, pp. 3–21.

[50] N. Ben Henda, "Generic and efficient attacker models in spin," in *Proceedings of the 2014 International SPIN Symposium on Model Checking of Software*, 2014, pp. 77–86.