DYSAN: Dynamically sanitizing motion sensor data against sensitive inferences through adversarial networks

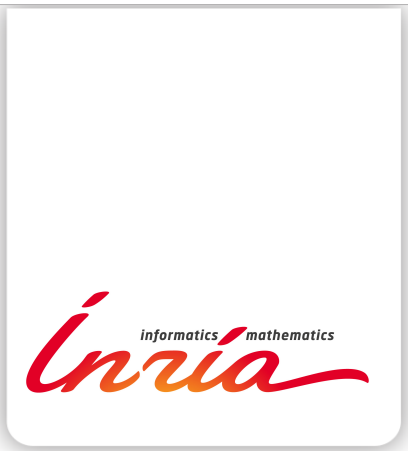
Antoine Boutet, Carole Frindel, Sébastien Gambs, Théo Jourdan, Claude Rosin Ngueveu

DySan: Dynamically sanitizing motion sensor data against sensitive inferences through adversarial networks

Antoine Boutet, Carole Frindel, Sébastien Gambs, Théo Jourdan, Rosin Claude Ngueveu

RESEARCH REPORT
N° 9325
February 2020
Project-Teams Privatics

ISSN 0249-6399 ISRN INRIA/RR--9325--FR+ENG



DySan: Dynamically sanitizing motion sensor data against sensitive inferences through adversarial networks

Antoine Boutet*, Carole Frindel†, Sébastien Gambs‡, Théo Jourdan*†, Rosin Claude Ngueveu‡

Project-Teams Privatics

Research Report n° 9325 — February 2020 — 17 pages

Abstract: With the widespread adoption of the quantified self movement, an increasing number of users rely on mobile applications to monitor their physical activity through their smartphones. Granting to applications a direct access to sensor data expose users to privacy risks. Indeed, usually these motion sensor data are transmitted to analytics applications hosted on the cloud leveraging machine learning models to provide feedback on their health to users. However, nothing prevents the service provider to infer private and sensitive information about a user such as health or demographic attributes.

In this paper, we present DySan, a privacy-preserving framework to sanitize motion sensor data against unwanted sensitive inferences (i.e., improving privacy) while limiting the loss of accuracy on the physical activity monitoring (i.e., maintaining data utility). To ensure a good trade-off between utility and privacy, DySan leverages on the framework of Generative Adversarial Network (GAN) to sanitize the sensor data. More precisely, by learning in a competitive manner several networks, DySan is able to build models that sanitize motion data against inferences on a specified sensitive attribute (e.g., gender) while maintaining a high accuracy on activity recognition. In addition, DySan dynamically selects the sanitizing model which maximize the privacy according to the incoming data. Experiments conducted on real datasets demonstrate that DySan can drastically limit the gender inference to 47% while only reducing the accuracy of activity recognition by 3%.

Key-words: privacy, articifial intelligence, transparency, health data, confidentiality

* Univ Lyon, INSA Lyon, Inria, CITI, F-69621 VILLEURBANNE, France
† Univ Lyon, INSA Lyon, CNRS, Inserm, CREATIS UMR 5220, U1206, F-69621 VILLEURBANNE, France
‡ Université du Québec à Montréal, Montréal, Québec, Canada

RESEARCH CENTRE GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

DySan: Assainissement dynamique des données de capteur de mouvement contre l'inférence d'informations sensibles à partir de réseaux adversariaux

Résumé : Avec l'adoption généralisée du suivi d'activité, un nombre croissant d'utilisateurs s'appuient sur des applications mobiles pour surveiller leur activité physique par le biais de leur smartphone. Le fait d'accorder aux applications un accès direct aux données des capteurs expose les utilisateurs à des risques pour leur vie privée. En effet, ces données de capteurs de mouvement sont généralement transmises à des applications d'analyse hébergées sur le cloud, qui exploitent des modèles d'apprentissage machine pour fournir aux utilisateurs un retour d'information sur leur santé. Cependant, rien n'empêche le fournisseur de services d'inférer des informations privées et potentiellement sensibles sur un utilisateur, telles que des attributs démographiques ou de santé.

Dans cet article, nous présentons DySan, un système de préservation de la vie privée pour assainir les données provenant de capteurs de mouvement contre les inférence non désirées d'informations sensibles (c'est-à-dire améliorer la vie privée) tout en limitant la perte de précision sur la surveillance de l'activité physique (c'est-à-dire maintenir une certaine utilité dans les données protégées). Pour garantir un bon compromis entre utilité et respect de la vie privée, DySan s'appuie sur des Réseaux génératifs Adversariaux (GAN) pour assainir les données issues des capteurs. Plus précisément, en apprenant de manière compétitive plusieurs réseaux, DySan est capable de construire des modèles d'apprentissage machine qui assainissent les données de mouvement contre l'inférence d'un attribut sensible spécifié (par exemple, le genre) tout en maintenant une grande précision sur la reconnaissance d'activité. De plus, DySan sélectionne dynamiquement le modèle d'assainissement qui maximise la confidentialité en fonction des données entrantes. Les expériences menées sur des ensembles de données réels montrent que DySan peut limiter considérablement l'inférence du genre jusqu'à 47% tout en n'impactant la précision de la reconnaissance d'activité que de 3%.

Mots-clés : vie privée, intelligence artificielle, transparence, données de santé, confidentialité

1 Introduction

The integration of motion sensors in smartphones and wearables has been accompanied by the growth of the quantified self movement. For instance nowadays, users increasingly use these devices to monitor their physical activity. Usually, the motion sensor data are transmitted to analytics applications hosted on the cloud leveraging machine learning models to compute heath statistics about users. While this analysis can bring many benefits in term of health [1, 2, 3], it can also lead to privacy breaches by exposing personal information regarding the individual concerned. Indeed, a large range of inferences can be done from motion sensor data including sensitive ones such as demographic and health-related attributes [4, 5].

Consider for instance the scenario in which Alice, a woman, use a fitness application on her smartphone to monitor her physical activity. The application performs the activity recognition as well as activity monitoring on the cloud. However, Alice has no guarantee that her data are not processed to infer other information for targeting or marketing purposes. Another example is the new trend of insurance companies that propose discount to clients if they accept to use a connected device to follow their activity along the days. These data can be used to provide a personalized coaching for better health management but also for early detection of a pathology, which can negatively impact the evolution of the insurance cost or lead to other discriminations. The goal of this work is to provide a solution which sanitizes the motion sensor data in such a way that it hides sensitive attributes while still preserving the activity information contained in the data.

To achieve this objective, we design DySan, that leverages on the framework of Generative Adversarial Networks (GANs) to sanitize the sensor data. More precisely, by learning in a competitive manner several networks, DySan is able to build models sanitizing motion data to prevent inferences on a specified sensitive attribute while maintaining a high level of activity recognition. In addition, by limiting the data distortion between the raw and sanitized data, DySan also maintains a high level of information in the sensor data to conduct further analysis (like step counting). Finally, each time DySan processes incoming data, it is able to dynamically selects the sanitizing model which limits as much as possible the risk of inference of the sensitive attribute.

The evaluation of DySan on real datasets, in which the gender is considered as the sensitive information to hide, demonstrates that DySan can drastically limit the gender inference up to 47% while only reducing the accuracy of activity recognition of 3%. In addition, by limiting the data distortion, we show that DySan also better preserves the sensor data utility compared to state-of-the-art approaches with respect to other analytical tasks such as estimating the number of steps. Finally, we show that the dynamic models selection of DySan provides an adaptation of the sanitization according to the incoming data in order to respect the user expectation in terms of utility and privacy trade-off.

Finally, our implementation of DySan as well as the considered datasets to assess its performance are publicly available [1].

The paper is organized as follows. First, the problem definition and the considered system model are described in Section 2. Then, DySan is presented in Section 3 before reviewing the experimental setting as well as the results obtained, respectively in Section 4 and Section 5. Finally, the related work is reviewing in Section 6 before concluding in Section 7.

[1] DySan: `https://github.com/DynamicSanitizer/DySan`

2 Problem definition and system model

We consider a mobile application installed on the user's smartphone aiming to monitor its physical activity. The smartphone of the user is assumed to be trusted by assumption. More precisely, we consider that DySan is deployed in a trust zone of the smartphone to avoid the mobile application to have a direct access to the motion sensor but only from the output of DySan (thus ensuring that the mobile application use only sanitized data). Afterwards, the mobile application sends the motion data to a server hosted on the cloud. This server leverages machine learning models such as classifiers to identify the activity of the user or to estimate other physical activity features (e.g., number of steps). However, we consider this server as untrusted in the sense that it can also try to infer additional sensitive information from the motion data. In this paper, we consider the gender as the sensitive attribute.

We consider raw motion data (denoted by A) captured through accelerometer and gyroscope that sample 3-axial signals with a frequency of 50 Hz. To enable activity recognition over time, the raw motion data are split in t sliding windows, where each sliding window is considered as being a sample of a single activity. Here, we consider 4 dynamic activities (i.e., walking, running, climbing and descending stairs) so that the length of sliding windows is chosen to match a walking cycle of two steps. Knowing that in average the cadence range of walking is not less than 1.5 steps per second [6], the window length is chosen to be 2.5 seconds with an overlap of 50 %.

We assume a population of N users and we call D the dataset containing all users data. This dataset includes the raw motion data as well as the label associated to the activity performed by the user (denoted by a multivalued attribute Y) and the binary sensitive attribute (denoted by S). Each measurement is timestamped. The dataset is therefore $D = (X_1, \ldots, X_t)$, in which $X \in \{A, Y, S\}$. We do not consider any correlation between the sensitive attribute S and the activity performed by the user Y.

The objective of DySan is to protect the user motion sensor data against inferences on sensitive attribute while maintaining as much as utility as possible. More formally, we aim at learning a set of transformation functions $S_{an_{\alpha,\lambda,\beta}}$ for various hyperparameters values α, λ, and β in which each function will transform the original dataset D into $\bar{D} = S_{an_{\alpha,\lambda,\beta}}(D) = \{\bar{X}_{1,\ldots,t}\}$. This transformation functions set is learned so that any model D_{isc} trained to predict S from the sanitized data $\bar{A}$ will fail while an activity predictor P_{red} trained on the same data $\bar{A}$ is able to maintain an accuracy close to the original data. To further preserve the utility of $\bar{D}$, we also constrain the data sanitization by ensuring that the distortion between the original and sanitized data is minimized.

Lastly, as the set of hyperparameters α, λ, β which provides the best utility and privacy trade-off is different for every user and can changes according to the incoming raw data, the choice of the considered transformation function $S_{an_{\alpha,\lambda,\beta}}$ is dynamically chosen locally on the smartphone of the user every time incoming data are processed.

3 DySan: Dynamic sanitizer

In this section, we start by providing an overview of our solution (Section 3.1) before developing in more detail how DySan builds sanitizer models (Section 3.2) in the training phase (Section 3.3) as well as during the online phase (Section 3.4).

3.1 Overview

An overview of DySan is depicted in Figure 1.To avoid an unwanted exploitation of the motion sensor data, these data are firstly sanitized by DySan before being transmitted to the mobile

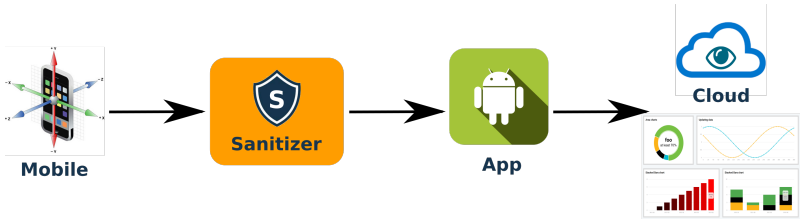


Figure 1: Overview of dynamic sanitizer.

application. More precisely, this sanitization removes the correlations with the gender in the raw data while preserving the information necessary to detect the activity performed by users. In addition, this sanitization limits the distortion between the raw and sanitized data to preserve the utility for other analytical tasks. Then, the resulting protected data are sent to an analytics application hosted on the cloud exploiting machine learning models to classify the activity of the user and compute statistics related to their physical activity.

Before exploit DySan, multiple sanitizing models corresponding to various utility-privacy trade-offs are built during the training phase. Once these multiple sanitizing models are deployed on the smartphone, DySan dynamically selects the model corresponding to the incoming processed data. Both phases, the training and the online phase are summarized in Figure 2 and explained in the following subsections.

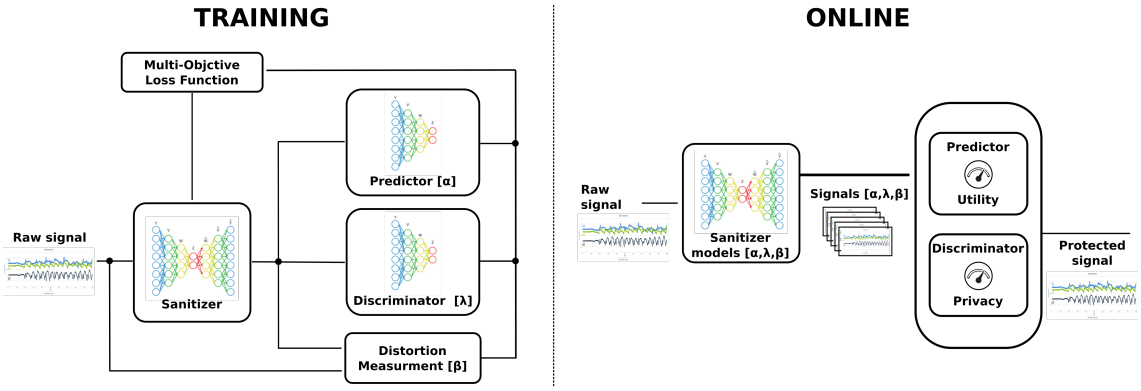


Figure 2: Dynamic Sanitizer framework during the training phase (left) and the online phase (right).

3.2 Building multiple sanitizing models

The training phase is performed only once and aims to prepare multiple sanitizing models. This training is performed with a reference dataset used for activity recognition (i.e., MotionSense dataset described in Section 4.1). As shown in Figure 2, DySan is composed of multiple building blocks: 1) a sanitizer, 2) a discriminator, 3) a predictor, 4) a distortion measurement, and 5) a multi-objective loss function.

3.2.1 Discriminator (D_{isc})

The discriminator guides the sanitizer through the process of removing information related to the sensitive attribute $S \in \{0, 1\}$. We used a Convolutional Neural Network (CNN), which is among the most suited neural networks to capture time-invariant features in time series [7]. The architecture of this CNN is presented in Appendix A.1. The discriminator is trained to infer the sensitive information from the output of the sanitizer (Section 3.2.4). The training of the discriminator is based on a loss function measuring the Balanced Error Rate (BER) [8] defined as :

$$BER(D_{isc}(\bar{A}, \bar{Y}), s) = \frac{1}{2}(\sum_{s=0}^{1} P(Disc(\bar{A}, \bar{Y}) \neq s | S = s)), \quad (1)$$

Note that BER ranges between 0 and 0.5, where a value close to 0 corresponds to a perfect accuracy for the prediction of the sensitive attribute. This loss will be called $Loss_{Sensitive}$ thereafter.

3.2.2 Predictor (P_{red})

The predictor aims at helping the sanitizer in preserving as much as possible information about the activities. We also use a CNN that has been optimized for predicting the user activity from the sanitized data. The architecture of this CNN is presented in Appendix A.2 The predictor is trained to maximise the accuracy in predicting activities from the output of the sanitizer (Section 3.2.4). To do so, we used again the balanced error rate as the loss function that should be minimized: $BER(P_{red}(\bar{A}, \bar{Y}), y)$. This loss will be called $Loss_{Activities}$ thereafter.

3.2.3 Distortion measurement

The last constraint on the sanitizer is the minimization of data distortion between the raw data and sanitized data. Specifically, this distortion should be limited to keep enough information in the motion data for subsequent analytical tasks. The distortion is measured through the L_1 loss function (denoted *l1*) applied independently on each data-dimension. This loss function for two vectors A and $\bar{A}$ is defined as follows:

$$l1(A, \bar{A}) = \frac{1}{N_A} \sum_{j=1}^{N_A} |a_j - \bar{a}_j|, \quad (2)$$

where $a_j \in A$ and N_A the number of dimensions in A.

3.2.4 Sanitizer (S_{an})

The sanitizer modifies the raw data to remove information correlated to the sensitive attribute while maintaining useful information for activity detection. Since the raw and sanitized data live in the same space, we implemented the sanitizer as an auto-encoder. An auto-encoder is a neural network performing a dimension reduction of the signal to delete irrelevant sensitive information before trying to reconstruct the input based on an multi-objective function. The sanitizer takes into account the feedback of the discriminator, predictor and distortion measurement to output the sanitized version of the input raw data. More precisely, these feedbacks are combined into a multi-objective loss function that should be minimized. The architecture of this neural network is presented in Appendix A.3.

3.2.5 Multi-objective loss function

The multi-objective loss function J^{San} drives the transformation performed by the auto-encoder to generated the sanitized data $\bar{A}$. This loss function takes into account three components, the capacity to detect the activity of the user (i.e., the output of the predictor), the capacity to detect the sensitive attribute (i.e., the output of the discriminator), and the level of distortion introduced in the sanitized signal compared to the raw signal. More formally, the multi-objective is defined as follows:

$$\begin{aligned} J^{San}(D,\ San,\ D_{isc},\ P_{red}) = \{&\alpha * d_s(S, D_{isc}(San(D))),\\ &\lambda * d_p(Y, P_{red}(San(D))),\\ &\beta * d_r(D, San(D))\}, \end{aligned}$$

where $d_s(x) = \frac{1}{2} - Loss_{Sensitive}$, $d_p = Loss_{Activities}$ and $d_r = \{l1(A_i, \bar{A}_i), \ldots\}$ with i representing each data dimension. The $\frac{1}{2}$ term in $d_s(x)$ comes from the objective of maximizing the error of the discriminator, since the sanitizer aims at transforming the data so that the discriminator is no more able to infer sensitive information.

The gradient descent is applied on J^{San} to minimize each of its component. Note that each loss term is weighted with a hyperparameter: d_s, d_p and d_r are weighted with α, λ and β, respectively. As we impose the constraint $\alpha+\lambda+\beta = 1$, we only control α and λ hyperparameters, leaving $\beta = 1-(\alpha+\lambda)$. α represents the relative importance given to the privacy while λ controls the utility (i.e., the quality of activity detection). We believe this setup is more realistic, as end users can control the utility and privacy trade-off.

3.3 Training Phase

During the training phase, we build a model for each set of possible value for α and λ in order to explore the entire domain of the multi-objective loss function. This exploration will allow DySan to dynamically test and select the best models for the incoming data in the online phase. The training procedure is resumed in Algorithm 1.

In order to optimize the utility and privacy trade-off for a specific set of value for α and λ (line 1, Algorithm 1), the three neural networks which composed DySan are trained in an adversarial manner. This adversarial training can be seen as a game between the sanitizer on one side and the predictor and the discriminator on the other side. Those neural networks will compete each other with opposing objectives until an equilibrium where no one can improve his own objective. In this competition, the sanitizer is trained to fool the discriminator and to maintained a high activity detection by the predictor while limiting the data distortion.

We follow the standard training procedure of GANs consisting in alternating iteratively the training of each individual models with their respective loss function until convergence (i.e., we do not consider Competitive Gradient Descent [9]).

Specifically, after the initialization (line $1-8$) we start by training the sanitizer with the J^{San} while the discriminator and the predictor are frozen (lines $11-12$). Once the training of the sanitizer is converged, the predictor and the discriminator are then trained independently with their respective loss function while the sanitizer is frozen (lines $13-20$). These two neural networks are trained until convergence (i.e., the loss is no longer decreased) or if a maximum number of iterations is reached, K_{pred} and K_{disc}, respectively. This two-steps process is iteratively performed until an equilibrium is reached.

Algorithm 1 DySan training algorithm

Input: D, λ, α, max_epoch, $batch\_size$, K_{pred}, K_{disc}.
Outputs: S_{an}, D_{isc}, P_{red}.
train(M, **trParams): Train the model M using trParams.
freeze(M): Freeze the model M parameters and avoid modifications.
{Initialisation}
S_{an}, D_{isc}, P_{red}, $D_d = \mathsf{shuffle}(D)$, $D_p = \mathsf{shuffle}(D)$
$\mathsf{Iterations} = \frac{|D|}{\mathsf{batch\_size}}$
{Training Procedure}
for $e = 1$ **to** max_epoch **do**
  for $i = 1$ **to** $Iterations$ **do**
    Sample batch B of size batch_size from D
    train(S_{an}, B, $J^{S_{an}}$, α, λ, freeze(S_{an}), freeze(D_{isc}))
    for $k = 1$ **to** K_{pred} **do**
      Sample batch B of size batch_size from D_p
      train(P_{red}, B, $Loss_{Activities}$, freeze(S_{an}))
    end for
    for $k = 1$ **to** K_{disc} **do**
      Sample batch B of size batch_size from D_d
      train(D_{isc}, B, $Loss_{Sensitive}$, freeze(S_{an}))
    end for
  end for
end for

3.4 Online Phase

Once deployed, DySan is composed of three components (Figure 2): 1) the sanitizer, 2) the discriminator, and 3) the predictor. Specifically, DySan contains all the models already trained in the training phase for the sanitized, the predictor and the discriminator. All these models correspond to all possible utility and privacy trade-off (i.e., set of values for α and λ).

DySan then processes all sanitizer models on the raw incoming data and evaluates the associated utility and privacy through both the corresponding discriminator and the predictor. Consequently, given incoming data DySan dynamically tests and selects the model that achieves the best privacy. This selection is done every time the sanitizer processes incoming data, specifically a time window of raw motion data.

4 Experimental setup

We evaluate the capacity of DySan to sanitize motion data from sensitive information while limiting the loss of activity recognition. In this section, we present the datasets used to assess DySan (Section 4.1), the baselines we compared with (Section 4.2), and the considered evaluation metrics (Section 4.3).

4.1 Datasets

We used two real datasets to assess DySan: MotionSense and MobiAct which are both publicly available. These datasets are split in trials with 2/3 of trials for training/validation and 1/3 for testing. MotionSense includes data sensed from accelerometer (i.e., acceleration and gravity) and

gyroscope at a constant frequency of 50Hz collected with an iPhone 6s kept in the participant's front pocket [10]. Overall, a total of 24 participants have performed six activities during 15 trials in the same environment and conditions. The considered activities are going downstairs, going upstairs, walking, jogging, sitting and standing.

MobiAct [11], in turn, includes data from 58 subjects with more than 2500 trials, all captured with a smartphone in a pocket. This dataset comprises data recorded from the accelerometer and gyroscope sensors of a Samsung Galaxy S3 smartphone for subjects performing nine different types of activities of daily living. In our case we only used the trials that have the same activities as MotionSense dataset.

4.2 Baselines

To compare the performance of DySan we consider a set of baselines.

ORF: To limit the exposure of the data, the raw signal in this solution [12] is preprocessed on the user's smartphone and only relevant features are transmitted to the application hosted on the cloud. These relevant features are firstly identified according to the target application (e.g., activity recognition) and selected either in the temporal or in the frequency domain. Originality performed to avoid users re-identification, we however use this approach to avoid to detect the sensitive attribute considered in this paper, the gender. To do that, we firstly identified the most relevant features correlated to the gender and then applied the methodology of the paper: we normalized the features in the frequency domain and removed the features in temporal domain that are not used for the activity classification.

GEN: Similarly to DySan, GEN (Guardian-Estimator-Neutralizer) [13] also uses an adversarial approach to optimize the utility and privacy trade-off. However, this solution does not follow the standard iterative training procedure of GANs (as described in Section 3.3). More precisely, the first network, a classifier, is learned once on the raw data to identify both sensitive and non sensitive information. Then the second one, an auto encoder, is also trained only once through a loss function which does not take into account the data distortion. Additionally, the hyperparameters are static for all users. While this solution uses an architecture of neural networks slightly different to our, we implement GEN by using our solution but with the following differences. Firstly the Predictor and the Discriminator are learned on raw data, secondly the loss function does not take into account the data distortion, and lastly the models used in the online phase is static and correspond to the best hyperparameters identified during the training phase.

Olympus: This approach [14] is similar to GEN but two different neural networks are used to learn the sensitive attributes and to learn sensitive information. In addition, these classifiers are trained using sanitized data by following an iterative process similar to DySan described Section 3.3. However, the loss function does not account data distortion and the models (i.e., the hyperparameters) are static for all users. While this approach is used for a different context (i.e., the sensitive attribute is the user identity), we also implement it by using our solution but the Sanitizer does not take into account the data distortion, and the model used in the online phase is static and correspond to the best hyperparameters identified during the training phase.

MSDA: This solution [15] can be viewed as an evolution of Olympus where the loss function driving the training of the auto encoder accounts data distortion. However, the model used in

the online phase is still static for all users. We also implement this solution by using DySan without dynamic models selection in the online phase.

4.3 Metrics

We evaluated DySan along both utility and privacy metrics.

Utility In our context of physical activity monitoring, the first considered utility metric is the accuracy on the prediction of the activity recognition. To do that, we measure the accuracy of a classifier trained to detect the users activities. More precisely, we use the confusion matrix to measure the number of correct predictions made by the model over all predictions that it makes. Its value ranges from 0 to 1, where 1 corresponds to perfect accuracy.

Analytics applications monitoring physical activity usually compute and present many estimators to users. To evaluate this aspect, we compute the number of steps we can detect from the sanitized signals and compare it with the number of steps in the raw data to estimate the conservation of this information. To do that, we first normalize raw and sanitized data in order to compare them in the same range of values, then we fix a threshold for each window of the raw data corresponding to the mean plus the standard deviation of the signal in the window. Then we count the number of peaks above the threshold and look at the corresponding sanitized window if we count the same number of peaks with the same threshold. We then compute a Peak Acceleration Threshold [16] from the raw data used to compute the number peaks [2] as an estimator of the number of steps detected by the analytic application from the received data.

Privacy To assess the level of privacy of DySan, we rely on the accuracy of inferring the sensitive attribute (i.e., the gender). The accuracy is comprised within 0 and 1. An accuracy of 1 means the classifier has not made any prediction mistake, and an accuracy of 0.5 corresponds to a random guess (in case of balanced dataset), means the inability of the classifier to predict the gender.

5 Evaluation

We carried out an extensive evaluation of DySan. In this section, we present the results by highlighting important features of DySan, namely the utility and privacy trade-off (Section 5.1), and the advantage of dynamic sanitized model selection (Section 5.2). Results reported in this section are the average over 10 repetitions of each experiments. The computation of the different global models (each corresponding to a precise set of hyperparameters) has been parallelized on a hybrid GPU/CPU computing farm.

5.1 Utility and Privacy trade-off

We evaluate here the capacity of an analytics application receiving the sanitized data from the mobile application to infer the gender of the user and its activity. We compare the performance of several classifiers that could be used by the application, namely a gradient boosting classifier (GB), a multi-layer perceptron (MLP), a decision tree (DT), a random forest (RF), a logistic regression (LR) and also two CNNs with the same architectures than the Predictor and the Discriminator (DySan). Figure 3 reports the accuracy for the gender and the activity with the

[2] We used Adaptiv: An Adaptive Jerk Pace Buffer Step Detection Algorithm (`https://github.com/danielmurray/adaptiv`)

different classifiers as well as in case of using the raw data. First of all, results show that without any protection (i.e., on raw data) the application is able to infer the gender with 98% of accuracy, which represent an important privacy risks. Secondly, results show that DySan successfully decreases the privacy risk while limiting the loss of activity detection. Indeed, with the sanitized data, an analytics application is only able to infer the gender up to 60% of accuracy. In term of utility, depending to the classifier, the accuracy of the activity recognition varies between 90% and 86% which represents only a small drop compared to using raw data.

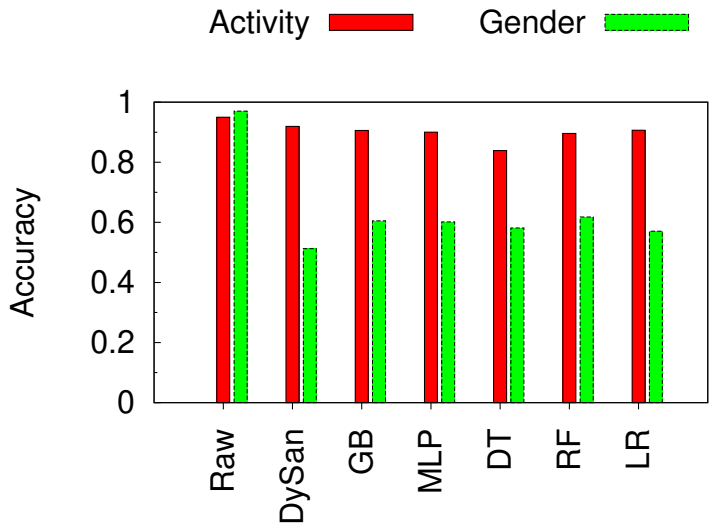


Figure 3: The sanitized data provided by DySan drastically decreases the privacy risk compared to using raw data while limiting the loss of activity detection.

The utility of the sanitized data is not just about the classification of activities but also about more precise information on the activity level. Then to show that DySan allows to keep relevant information in the signal allowing to conduct further analysis, we consider here the computation of the number of steps from the signal. Following the step detection method presented in 4.3, Table 1 shows that with DySan we can find almost all the steps detected from raw data with less than 5% of errors. We compared this result with baselines: the sanitized signals appears to be much more noisy and the step detection is greatly impacted with more than 64% steps over-detected for Olympus, more than 29 % for MSDA and more than 12% of errors for GEN. The method ORF is not considered here because it only extracts features and the signal is not preserved, prohibiting at the same time further analysis.

	Steps
Raw data	14387
DySan	13699 (**-4.88** %)
GEN	12817 (**-12.25**%)
Olympus	23658 (+**64.44**%)
MSDA	18624 (+**29.45**%)

Table 1: The sanitized signal of DySan appears to be much more useful for step detection than comparative approaches.

Lastly, we compare DySan against comparative baselines. As depicted on Figure 4, results show that DySan outperforms other approaches by limiting the gender inference to 47% while only reducing the accuracy of activity recognition by 3% compared to without any protection (i.e., using raw data). In addition, results also show the performance improvement provided by the evolution of the approaches based on adversarial networks. GEN, Olympus, and then MSDA gradually reduce the gender inference according to the considered features (an accuracy from 96% to 63%). DySan then makes this gender inference as precise as a random guess with an accuracy of 51%.

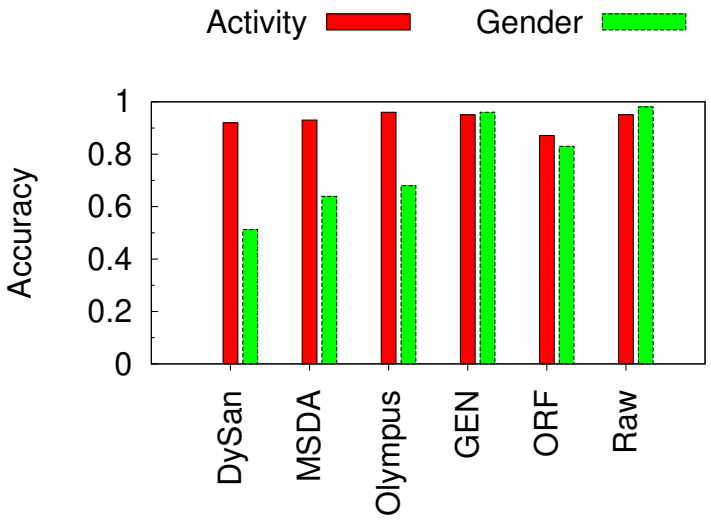


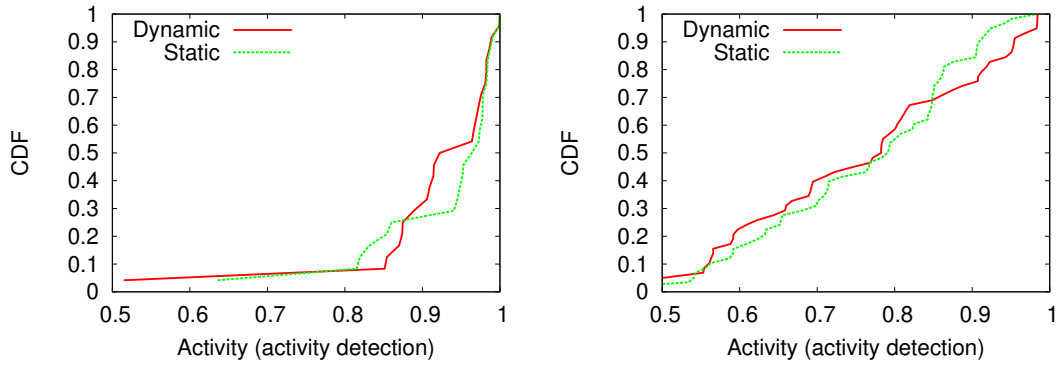
Figure 4: DySan provides the best utility and privacy trade-off compared to comparative baselines.

Interesting enough, we were not able to reproduce the results of GEN reported in [13]. Indeed, this original paper reports an accuracy of 94% for the activity recognition and 64% for the gender inference for the same dataset (MotionSense) compared to 95% and 96%, respectively in our experiments.

5.2 Dynamic Sanitizer Model Selection

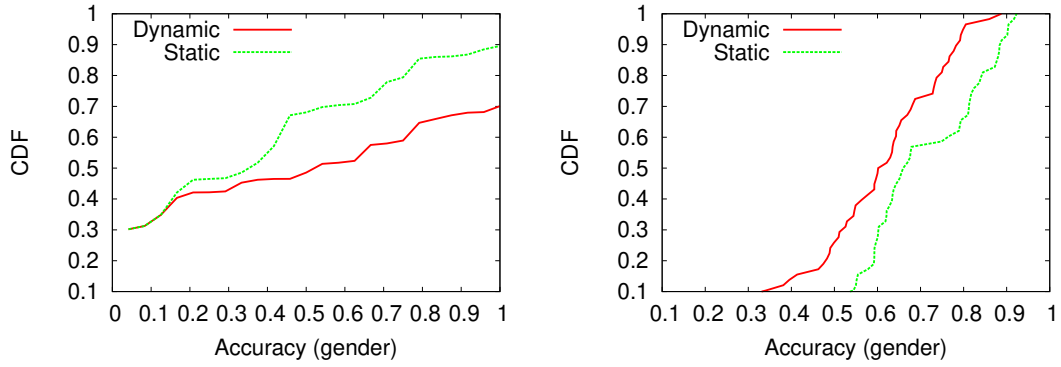
During its training phase, DySan computes the sanitizer models corresponding to all possible utility and privacy trade-off by exploring the range of values for the hyperparameters α and λ. We evaluate here the capacity of DySan in its online phase to dynamically select the best sanitizer model according to the incoming data of the user. Firstly, we compute the accuracy for both the gender inference and the activity recognition when the sanitizer model is fixed for all the users. In this case which represents the behaviors of all comparative baselines, the considered model is the one which provides the best performance in average for all the users. Secondly, we compare this static solution against DySan where the sanitizer model is dynamically selected according to the incoming data. In this case, the selected model is the one which provides the smallest accuracy in term of gender inference according to the current user.

Figure 5 and Figure 6 show for a static and a dynamic sanitizer model selection, the distribution of the accuracy for both the gender inference and the activity recognition, respectively, over the population of users for MotionSense and MobiAct dataset. Firstly, results show that the accuracy in both cases is highly heterogeneous over the population of users (excepting for



(a) MotionSense (b) MobiAct

Figure 5: The dynamic sanitizer model selection does not significantly impact the activity recognition.



(a) MotionSense (b) MobiAct

Figure 6: By dynamically selecting the best sanitizer model over time for each user, DySan greatly improved the protection against gender inference.

the utility for MotionSense dataset where most of the users benefit from an important accuracy). This high heterogeneity reflects the fact that the selected models is not adapted for all users. However, results secondly show that the dynamic model selection significantly reduces the gender inference (around 20% and 5% less accurate in average for MotionSense and MobiAct, respectively). In term of utility, accuracy in both cases is instead similar. Lastly, as DySan was trained on MotionSense dataset and tested on MobiAct, we also show a good transferability to new users unseen by DySan.

6 Related Works

With the availability of wearable and personnal devices, there have been a growing research on the use of collected data for quantifying various aspects of personnal life, such as the number of calories consumed, the blood pressure, etc. A growing litterature concern the use of data for predicting the physical activities performed by users, let it be for either medical, insurance or various other reasons. We refer the interested reader to the surveys [17] and [18] on machine learning and deep learning techniques applied for predicting the type of activities performed.

In this section, we compare our approach with other existing techniques that protect sensitive information in sensor data while retaining data utility. First, we focus on approaches used as baselines previously in the paper. [13] is the only one which focus on the gender as the sensitive information. [12] and [15] in their case, focus on the re-identification only while [14] apply their approach on several applications like object recognition or action recognition with several data types such as images or motion sensors. In the case adversarial approach that use autoencoder, output corresponding to the sensitive information can be extracted from the representation produced by the encoder [19] , the decoder [14] which also correspond to our approach, or both the encoder and the decoder [15] for data sanitization.

Specific to the sensor data generation, SenseGen [20] is a deep learning architecture for protecting users privacy by generating synthetic sensor data. Unfortunately, they did not provide any garantee on the protection.

To enlarge with other applications basing on protecting sensitive informations using adversarial methods, [21] use a VGAN to transform face images in order to hide facial expression of the users that can be used to reveal their identity while preserving generic expressions. Adversarial approaches can also be used to hide sensitive information such as text in images [22] or identity information in the fingerprints [23].

From a broader privacy perspective, [24] proposes an adversarial network technique to minimize the amount of mutual information between a sensitive attribute and useful data while bounding the amount of distortion introduced. They applied their solution on a synthetic and a computer vision dataset. Inspired from [24], authors in [25] have developed a method for learning an optimal privacy protection mechanism also inspired from GAN, which they have applied to location privacy. In [26], authors have proposed an approach called table-GAN, which aim at preserving privacy by generating synthetic data. By suppressing *one-to-one* relationship and limiting the quality of dataset reconstruction re-identification attacks are rendered less performant. They compared their approach with standard privacy techniques such as k-anonymity, t-closeness.

7 Conclusion

We presented DySan, a privacy-preserving framework which sanitizes motion sensor data in order to prevent unwanted inference of sensitive information. At the same time, DySan preserves as

much as possible the useful information for activity recognition and other estimators of physical activity monitoring. Results show that DySan drastically reduces the risk of gender inference without impacting the ability to detect its activity or to monitor the number of steps. We also show that the dynamic sanitized model selection of DySan successfully adapts the protection to each user. Lastly, we also compared our approach with existing approaches and demonstrated that DySan provides better control over privacy-utility trade-off.

References

[1] G. D. Fulk and E. Sazonov, "Using sensors to measure activity in people with stroke," Topics in Stroke Rehabilitation, vol. 18, no. 6, pp. 746–757, 2011.

[2] E. Park, H.-J. Chang, and H. S. Nam, "Use of machine learning classifiers and sensor data to detect neurological deficit in stroke patients," Journal of Medical Internet Research, vol. 19, no. 4, p. e120, 2017.

[3] J. Qi, P. Yang, D. Fan, and Z. Deng, "A survey of physical activity monitoring and assessment using internet of things technology," in International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, 2015, pp. 2353–2358.

[4] J. Han, E. Owusu, L. T. Nguyen, A. Perrig, and J. Zhang, "Accomplice: Location inference using accelerometers on smartphones," in International Conference on Communication Systems and Networks, 2012, pp. 1–9.

[5] J. L. Kröger, P. Raschke, and T. R. Bhuiyan, "Privacy implications of accelerometer data: a review of possible inferences," in International Conference on Cryptography, Security and Privacy, 2019, pp. 81–87.

[6] C. BenAbdelkader, R. Cutler, and L. Davis, "Stride and cadence as a biometric in automatic person identification and verification," in International Conference on Automatic Face Gesture Recognition, 2002, pp. 372–377.

[7] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," Data Mining and Knowledge Discovery, vol. 33, no. 4, pp. 917–963, 2019.

[8] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, "Certifying and removing disparate impact," in International Conference on Knowledge Discovery and Data Mining. ACM, 2015, pp. 259–268.

[9] F. Schäfer and A. Anandkumar, "Competitive gradient descent," in Advances in Neural Information Processing Systems, 2019, pp. 7623–7633.

[10] J. L. Reyes-Ortiz, Smartphone-based human activity recognition. Springer, 2015.

[11] G. Vavoulas, C. Chatzaki, T. Malliotakis, M. Pediaditis, and M. Tsiknakis, "The mobiact dataset: Recognition of activities of daily living using smartphones," in International Conference on Information and Communication Technologies for Ageing Well and e-Health, 2016, pp. 143–151.

[12] T. Jourdan, A. Boutet, and C. Frindel, "Toward privacy in iot mobile devices for activity recognition," in International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, 2018, pp. 155–165.

[13] M. Malekzadeh, R. G. Clegg, A. Cavallaro, and H. Haddadi, "Protecting sensory data against sensitive inferences," in Workshop on Privacy by Design in Distributed Systems, 2018, pp. 2:1–2:6.

[14] N. Raval, A. Machanavajjhala, and J. Pan, "Olympus: Sensor privacy through utility aware obfuscation," Privacy Enhancing Technologies, vol. 2019, no. 1, 2019.

[15] M. Malekzadeh, R. G. Clegg, A. Cavallaro, and H. Haddadi, "Mobile sensor data anonymization," Internet of Things Design and Implementation, 2019.

[16] A. Abadleh, E. Al-Hawari, E. Alkafaween, and H. Al-Sawalqah, "Step detection algorithm for accurate distance estimation using dynamic step length," in International Conference on Mobile Data Management, 2017, pp. 324–327.

[17] H. F. Nweke, Y. W. Teh, M. A. Al-Garadi, and U. R. Alo, "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges," Expert Systems with Applications, vol. 105, pp. 233–261, 2018.

[18] S. Ramasamy Ramamurthy and N. Roy, "Recent trends in machine learning for human activity recognition - a survey," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 8, no. 4, p. e1254, 2018.

[19] C. Liu, S. Chakraborty, and P. Mittal, "Deeprotect: Enabling inference-based access control on mobile sensing applications," 2017.

[20] M. Alzantot, S. Chakraborty, and M. Srivastava, "Sensegen: A deep learning architecture for synthetic sensor data generation," in International Conference on Pervasive Computing and Communications Workshops, 2017, pp. 188–193.

[21] J. Chen, J. Konrad, and P. Ishwar, "Vgan-based image representation learning for privacy-preserving facial expression recognition," 2018.

[22] H. Edwards and A. Storkey, "Censoring representations with an adversary," 2015.

[23] W. Oleszkiewicz, P. Kairouz, K. Piczak, R. Rajagopal, and T. Trzcinski, "Siamese generative adversarial privatizer for biometric data," 2018.

[24] A. Tripathy, Y. Wang, and P. Ishwar, "Privacy-preserving adversarial networks," in Annual Allerton Conference on Communication, Control, and Computing, 2019, pp. 495–505.

[25] M. Romanelli, C. Palamidessi, and K. Chatzikokolakis, "Generating optimal privacy-protection mechanisms via machine learning," arXiv preprint arXiv:1904.01059, 2019.

[26] N. Park, M. Mohammadi, K. Gorde, S. Jajodia, H. Park, and Y. Kim, "Data synthesis based on generative adversarial networks," International Conference on Very Large Data Bases, vol. 11, no. 10, pp. 1071–1083, 2018.

A Neural Network Architecture

We provide in this section details about the underlying neural networks of DySan.

A.1 Discriminator Net

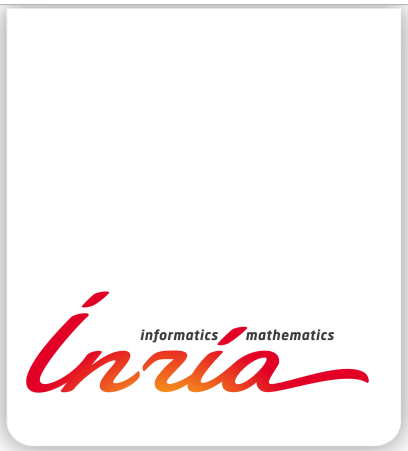
1. Input (125,6)
2. Conv1D (64, kernel_size=6, stride=1, activation=ReLU)
3. AvgPool1D(kernel_size=2, stride=2)
4. BatchNorm1D(100, eps=1e-05, momentum=0.1)
5. Dropout(p=0.5)
6. Dense(64, activation=ReLU)
7. Dense(2, activation=softmax)

A.2 Predictor Net

1. Input (125,6)
2. Conv1D (100, kernel_size=6, stride=1, activation=ReLU)
3. AvgPool1D(kernel_size=2, stride=2)
4. BatchNorm1D(100, eps=1e-05, momentum=0.1)
5. Conv1D(100, kernel_size=5, stride=1, activation=ReLU)
6. AvgPool1d(kernel_size=2, stride=2)
7. Conv1D(160, kernel_size=5, stride=1, activation=ReLU)
8. AvgPool1d(kernel_size=2, stride=2)
9. Conv1D(160, kernel_size=5, stride=1, activation=ReLU)
10. AvgPool1d(kernel_size=2, stride=2)
11. Dropout(p=0.5)
12. Dense(64, activation=ReLU)
13. Dense(4, activation=softmax)

A.3 Sanitizer Net

1. Input (125,6)
2. Conv1D (64, kernel_size=6, stride=1,)
3. Conv1D (128, kernel_size=5, stride=1)
4. Dense(128)
5. Dense(64, activation=LeakyReLU(0.01))
6. Dense(64)
7. Dense(128)
8. Deconv1D (128, kernel_size=5, stride=1)
9. Deconv1D (64, kernel_size=5, stride=1, activation=softmax)



RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399