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

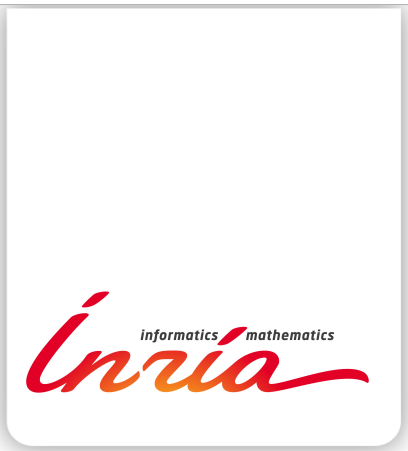
Rosin Claude Ngueveu, Antoine Boutet, Carole Frindel, Sébastien Gambs, Théo Jourdan, Claude Rosin

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
Octobre 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 — Octobre 2020 — 26 pages

* 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

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. However, granting applications a direct access to sensor data expose users to privacy risks. In particular, motion sensor data are usually transmitted to analytics applications hosted on the cloud, which leverages on machine learning models to provide feedback on their activity status to users. In this setting, nothing prevents the service provider to infer private and sensitive information about a user such as health or demographic attributes. To address this issue, we propose 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). Our approach is inspired from the framework of Generative Adversarial Networks to sanitize the sensor data for the purpose of ensuring a good trade-off between utility and privacy. 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 an accurate activity recognition. DySan builds various sanitizing models, characterized by different sets of hyperparameters in the global loss function, to propose a transfer learning scheme over time by dynamically selecting the model which provides the best utility and privacy trade-off according to the incoming data. Experiments conducted on real datasets demonstrate that DySan can drastically limit the gender inference up to 41% (from 98% with raw data to 57% with sanitized data) while only reducing the accuracy of activity recognition by 3% (from 95% with raw data to 92% with sanitized data).

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

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 construit divers modèles d'assainissement, caractérisés par différents ensembles d'hyperparamètres dans la fonction de perte globale, pour proposer un schéma d'apprentissage du transfert dans le temps en sélectionnant dynamiquement le modèle qui offre le meilleur compromis entre utilité et respect de la vie privée en fonction des données entrantes.

Les expériences menées sur des ensembles de données réels montrent que DySan peut limiter considérablement l'inférence de genre jusqu'à 41% (de 98% avec des données brutes à 57% avec des données assainies) tout en ne réduisant la précision de la reconnaissance d'activité que de 3% (de 95% avec des données brutes à 92% avec des données assainies)

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 [36]. For instance nowadays, users increasingly exploit these devices to monitor their physical activities. Usually, the motion sensor data are not analyzed directly on the device but are rather transmitted to analytics applications hosted on the cloud. These analytics applications leverage machine learning models to compute statistical indicators related to the status of users that are send back to them. While these analyses can bring many benefits from the health perspective [11, 24, 26], they 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 [12, 14, 15].

Consider for instance the scenario in which Alice, a woman, uses a fitness application on her smartphone to monitor her physical activity. The application performs the activity recognition as well as the activity monitoring on the cloud. However even if the service provider declares that it will never do it, Alice has no formal guarantees that her data will not be processed to infer other information about her (*e.g.*, for targeting or marketing purposes). Another possible scenario is related to the new trend of insurance companies that propose discount to clients if they accept to use a connected device to follow their daily activity [33]. 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 insurance cost or lead to other type of discrimination. To address the issues raised by these scenarios, in this work we propose a solution sanitizing the motion sensor data in such a way that it hides sensitive attributes while preserving the activity information contained in the data.

To achieve this objective, we design DySan, inspired from the framework of Generative Adversarial Networks (GANs) [23] 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 distortion between the raw and sanitized data, DySan also maintains a high level of utility with respect to other analysis tasks related to activity monitoring (*e.g.*, steps counting).

Furthermore, our approach aims at addressing the heterogeneous aspect of sensor data, which is inherent to the way each user moves, to the characteristics of the device used for data collection and to the evolution of activity during the day. Thus, as one sanitizing model cannot provide the best utility and privacy trade-off for all users over time, DySan builds a set of diverse sanitizing models by exploring different combination of hyperparameters balancing loss functions of activity recognition, sensitive inference and data distortion terms. By doing so, DySan is able to dynamically select the model which provides the best trade-off over time according to the incoming sensor data.

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 41% while only inducing a drop of 3% on the accuracy of activity recognition. In addition to preserve activity recognition, DySan, by limiting data distortion, also preserves the sensor data utility for other analytical tasks such as estimating the number of steps. Moreover, we show that the dynamic model selection of DySan successfully provides an adaptation of the sanitization according to the incoming user data. This dynamic model selection is specially useful to transfer learning from the dataset used to build the sanitizer models to another dataset with new users with potentially different data distribution. Our dynamic sanitization method overcomes several shortcomings of the state-of-the-art approaches, namely the use of the same sanitization model

for all users over time, which may lead to a poor privacy-utility trade-off for atypical users. Lastly, we evaluate the cost of operating DySan on a smartphone and show that the introduced overhead is compatible with real-time processing and that the energy consumption remains reasonable. Our implementation of DySan as well as the datasets used to assess its performances are publicly available [1]

The outline of the paper is 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 discussed in Section 6 before concluding in Section 7.

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. For instance, we consider that DySan could be deployed in the trusted environment of the smartphone to prevent the mobile application to have a direct access to the sensor data but only from the output of DySan (thus ensuring that the mobile application uses only sanitized data). Afterwards, the mobile application sends the sanitized data to a server hosted on the cloud. This server leverages machine learning models to identify the activity of the user or to estimate other physical activity features (*e.g.*, number of steps). The server is considered to follow the honest-but-curious adversary model in the sense that it may also try to infer additional sensitive information from the sensor data. For the rest of the paper, we consider the *gender* as being the sensitive attribute to protect. Note however that our approach is much more generic and could be applied to protect other sensitive attributes (*e.g.*, handicap). This choice is only motivated by the availability of different datasets with this information. Note also that the gender could be inferred from the list of performed activities and their associated frequencies in case of unbalanced data distribution between men and women (which is not the case in the datasets considered in this paper).

We consider raw motion sensor 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 sensor data are split in sliding windows, in which each sliding window is considered to be a sample of a single activity (*i.e.*, by assumption the user cannot perform two different activities during a single sliding window). The datasets used are composed of four type of dynamic activities (*i.e.*, walking, running, climbing and going down stairs), and we chose the length of sliding window to match a walking cycle of two steps. The choice of the window size is not trivial, especially for an activity recognition task, and has to be well calibrated. Indeed, a small window size could split an activity signal while large window size could contain multiple activity signals. Knowing that in average the walking pace is not less than 1.5 steps per second [4], the window length T is chosen to be 2.5 seconds with an overlap of 50 %.

We assume a population of N users contained in a dataset X storing all users data. This dataset includes the raw sensor data as well as the label associated to the activity performed by the user (denoted by a multi-valued attribute Y), the binary sensitive attribute (denoted by S) and a timestamp. Thus, the dataset $X = \{A, Y, S\}$ in which $A = (A_1, \ldots, A_T)$.

The objective of DySan is to protect the user motion sensor data against sensitive attribute inferences while maintaining data utility. More formally, we aim at learning a set of sanitizers $S_{an_{\alpha,\lambda,\beta}}$ for various values of the hyperparameters α, λ and β. Each sanitizer will transform the original data X into $\bar{X} = S_{an_{\alpha,\lambda,\beta}}(X) = \{\bar{A}, Y, S\}$; $\bar{A} = (\bar{A}_1, \bar{\ldots}, \bar{A}_T)$. This set of sanitizers is learned so that it is difficult to build a discriminator D_{isc} trained to predict S from the sanitized

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

data and activities $\{\bar{A}\, Y\}$ while an activity predictor P_{red} trained on the same sensor data ($\bar{A}$) is able to maintain an accuracy close to the original one. To further preserve the utility of $\bar{X}$, we also constrain the sanitization process to minimize the distortion between the original and sanitized data.

Furthermore, DySan aims to dynamically adapt over time the hyperparameters of the model according to the incoming data of each user. Indeed, while a particular model could provide the best utility/privacy trade-off on average for all users with respect to training dataset, the model leading to the best trade-off can change when testing on new user (*e.g.*, when the new user data does not fit the data distribution of the training dataset). More formally, we aim to find for each window of data the sanitizer $\widehat{S_{an_{\alpha,\lambda,\beta}}}$ providing the best utility/privacy trade-off for the current incoming data. This trade-off is defined by a metric combining the accuracy of the activity recognition and the inference of the sensitive attribute.

3 DYSAN: Dynamic Sanitizer

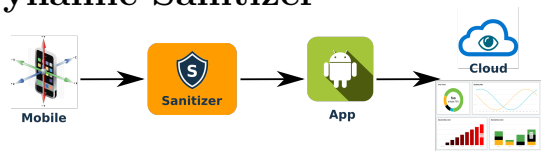


Figure 1: Overview of dynamic sanitizer.

An overview of DySan is shown in Figure 1. To avoid an unwanted exploitation of the motion sensor data, these data are sanitized by DySan before being transmitted to the mobile application. This sanitizing process removes the correlations with S in the sensor data while preserving the information necessary to detect the activity performed by a user. In addition, DySan also aims at limiting the distortion between the raw and sanitized data to preserve the utility for other analytical tasks. Finally, the resulting sanitized data are sent to an analytics application hosted on the cloud, exploiting machine learning models to classify the users activity and compute statistics related to their physical activity. Before exploiting DySan, multiple sanitizers corresponding to various utility and privacy trade-offs are built during the training. These models are then deployed on the smartphone. During the online phase, DySan selects the best sanitizer for the associated user. Both the training and the online phases are summarized in Figure 2 and explained in the following subsections.

3.1 Building multiple sanitizers

The training phase is performed only once and aims at learning multiple sanitizers. This training is performed with a reference dataset used in activity recognition, the MotionSense dataset that we describe in Section 4.1. As shown in Figure 2, DySan is composed of multiple building blocks that we detail hereafter: 1) a sanitizer, 2) a discriminator, 3) a predictor, 4) a distortion measurement and 5) a multi-objective loss function.

- **Discriminator:** The discriminator D_{isc} guides the sanitizer through the process of removing information related to the sensitive attribute $S \in \{0, 1\}$. In practice, we use a Convolutional Neural Network (CNN), which is well-suited to capture time-invariant features in time series [9]. 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. The training of the discriminator is based on a loss function measuring the Balanced Error Rate BER [10] between the output of the discriminator and the ground truth sensitive attribute, which is defined as:

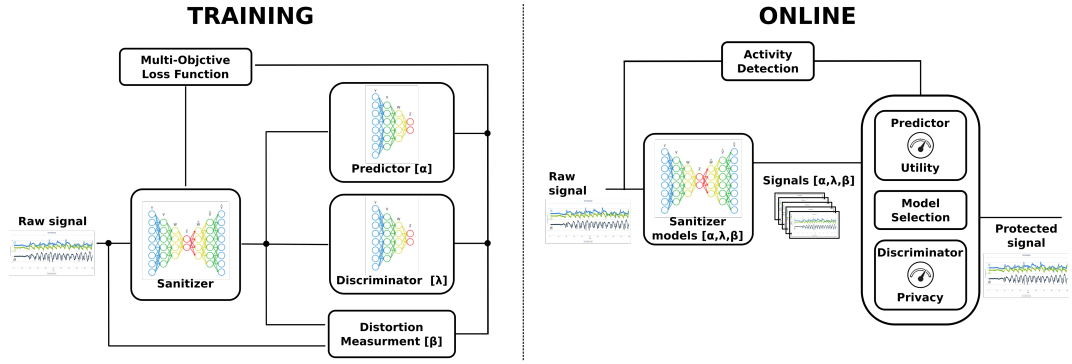


Figure 2: Dynamically sanitizing motion sensor data with DySan framework during the training (left) and the online (right) phases. Training phase allows to build different models that are distinguished by their parameters and online phase allows to choose among these models the most adapted to the final user.

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

The value of BER ranges between 0 and 0.5, in which a value close to 0 corresponds to a perfect accuracy for the prediction of the sensitive attribute while 0.5 means the discriminator is unable to retrieve any information about the sensitive attribute from the sanitized data. Hereafter, we will refer to this loss by $Loss_{Sensitive}$.

- **Predictor:** The predictor P_{red} aims at helping the sanitizer in preserving as much information as possible with respect to the activity recognition task. We also use a CNN for the predictor that has been optimized for predicting the user activity from the sanitized data. The architecture of this CNN is presented in Appendix A.2. Thus, the predictor is trained to maximize the accuracy in inferring activities from the output of the sanitizer. We also use the balanced error rate as the loss function that should minimize the error between the output of the predictor and the ground truth of the activity: $BER(P_{red}(\bar{A}), y)$. For the rest of the paper, we will refer to the predictor loss as $Loss_{Activities}$.

- **Distortion measurement:** The last constraint on the sanitizer is the minimization of data distortion between the raw and sanitized data. Specifically, this distortion should be limited to keep as much information as possible in the sensor data for subsequent analytical tasks. The data distortion is measured through the L_1 loss function denoted *l1*, applied independently on each attribute. For two vectors A_i and $\bar{A}_i$, corresponding respectively to the raw and sanitized sensor data, the loss function is defined as follows:

$$l1(A_i, \bar{A}_i) = \frac{1}{N_A} \sum_{j=1}^{N_A} |a_{ij} - \bar{a}_{ij}|, \tag{2}$$

in which N_A is the number of possible values for a particular attribute (*e.g.*, the number of axes of the accelerometer or the gyroscope), $a_{ij} \in A_i$ and i denotes a single observation in

the window of length T.

- **Sanitizer:** The sanitizer S_{an} modifies the raw data to remove information correlated with the sensitive attribute while maintaining useful information for activity detection. Since the raw and sanitized data belong to the same space, we have implemented the sanitizer as an auto-encoder. In a nutshell, an auto-encoder is a neural network performing a dimension reduction of the signal to compress information before trying to reconstruct the input. 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 different feedbacks are integrated into a multi-objective loss function that should be minimized. The architecture of the auto-encoder is given in Appendix A.3.

- **Multi-objective loss function** The multi-objective loss function $J^{S_{an}}$ drives the transformation performed by the auto-encoder to generated the sanitized data $\bar{X}$. 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 data compared to the original one. More formally, the multi-objective is defined as follows:

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

in which $d_s(x) = \frac{1}{2} - Loss_{Sensitive}$, $d_p = Loss_{Activities}$ and $d_r = \{l1(a_{:,j}, \bar{a}_{:,j}), \ldots\}$ with $a_{:,j}$ representing a dimension of all timesteps of a single sliding window. The term $\frac{1}{2}$ in $d_s(x)$ comes from the objective of maximizing the error of the discriminator, since the sanitizer aims at sanitizing the data so that the discriminator is no more able to infer sensitive information.

A gradient descent is applied on $J^{S_{an}}$ to minimize the global loss function following a similar approach as in [2]. Note that each loss term is weighted with a hyperparameter. More precisely, d_s, d_p and d_r are weighted respectively with α, λ and β. The parameter α represents the relative importance given to the privacy while λ controls the utility (*i.e.*, the quality of activity detection). As we impose the constraint that $\alpha + \lambda + \beta = 1$, we only adjust α and λ hyperparameters, leaving $\beta = 1 - (\alpha + \lambda)$.

3.2 Training Phase

During the training phase, we build a sanitizer for each set of possible values for the hyperparameters α and λ to explore the domain of the multi-objective loss function. This exploration will allow DySan to select the best model for each user during the online phase. The training procedure is summarized in Algorithm 1.

In order to optimize the utility and privacy trade-off for a specific set of α and λ (line 1, Algorithm 1), the three neural networks 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. These neural networks compete against each other with opposing objectives until an equilibrium is reached. More precisely, the sanitizer is trained to fool the

discriminator and maintained a high activity detection quantified with the predictor while limiting the data distortion. We follow the standard training procedure of GANs consisting in alternating in an iterative manner (at each batch of data) the training of each model with their respective loss function until convergence or until a maximum number of epoch (*i.e.*, we do not consider Competitive Gradient Descent [31]).

Specifically, after initialization (lines $1-8$) the training of the sanitizer starts with $J^{S_{an}}$ while the discriminator and the predictor are frozen (lines $11-12$). Once the training of the sanitizer has converged, the predictor and the discriminator are 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.*, until the loss no longer decreased) or if a maximum number of iterations, respectively K_{pred} and K_{disc}, is reached. This two-steps process is performed iteratively until an equilibrium is reached.

Algorithm 1 DySan training algorithm

1: **Input:** X, λ, α, max_epoch, $batch\_size$, K_{pred}, K_{disc}.
2: **Outputs:** S_{an}, D_{isc}, P_{red}.
3: **train(M, **trParams):** Train the model M using trParams.
4: **freeze(M):** Freeze the model M parameters and avoid modifications.
5: {Initialisation}
6: S_{an}, D_{isc}, P_{red}, $X_d = \mathsf{shuffle}(X)$, $X_p = \mathsf{shuffle}(X)$
7: $\mathsf{Iterations} = \frac{|D|}{\mathsf{batch\_size}}$
8: {Training Procedure}
9: **for** $e = 1$ **to** max_epoch **do**
10: **for** $i = 1$ **to** Iterations **do**
11: Sample batch B of size batch_size from X
12: train(S_{an}, B, $J^{S_{an}}$, α, λ, freeze(P_{red}), freeze(D_{isc}))
13: **for** $k = 1$ **to** K_{pred} **do**
14: Sample batch B of size batch_size from X_p
15: train(P_{red}, B, $Loss_{Activities}$, freeze(S_{an}))
16: **end for**
17: **for** $k = 1$ **to** K_{disc} **do**
18: Sample batch B of size batch_size from X_d
19: train(D_{isc}, B, $Loss_{Sensitive}$, freeze(S_{an}))
20: **end for**
21: **end for**
22: **end for**

3.3 Online Phase

Once deployed on the smartphone, DySan is composed of four components as depicted in Figure 2: the sanitizer, the discriminator, the predictor and an activity detection component. Specifically, DySan knows all the sanitizer, predictor and discriminator models built during the training phase. This set of models correspond to the different possible utility and privacy trade-offs (*i.e.*, set of values explored for the α and λ hyperparameters). The selection of the model is performed by maximizing $S(P,U) = xU + yP$, where x and y being positive weight coefficients with $x + y = 1$, U the evaluation of the activity done by the predictor, and P the accuracy in terms of privacy as $P = 1 - |0.5 - p|$, where P is the evaluation of the gender done by the discriminator. Consequently, P is higher when the evaluation of the gender accuracy corresponds to a random guess (i.e., an

accuracy of 0.5). According to the expected utility and privacy trade-off, the coefficients x and y can be tuned (Figure 12).

To find the best sanitizer over time (according to coefficients x and y), DySan evaluates the utility and the privacy of all models to select the best one. This evaluation requires to know the actual activity performed by the user and the sensitive attribute. While the sensitive attribute can be given by the user, the motion sensor data are not labeled with the activities as it is rather the objective of the activity recognition task to perform this inference.

We use the activity detection component (see Figure 2) to annotate some motion sensor data with their activities on the smartphone. More precisely, we ask the user to follow a specific calibration process at the installation of DySan. During this process, the user is asked to perform a series of different activities for short periods to learn a specific supervised classifier to detect his activities. As the quantity of data available to train this classifier is limited, we rely on the use of random forests that are adapted to this context [13]. This random forest (RF) classifier is then used to label the raw data in order to evaluate the utility of all sanitizers. This evaluation is performed on a regular basis (*e.g.*, each period of p windows) and we compute the average accuracy over this period. By following this process, DySan is able to identify over time the sanitizer providing the best utility and privacy trade-off defined as a measure combining the accuracy of the activity recognition and the inference of the sensitive attribute.

4 Experimental setting

4.1 Datasets

We used two real datasets, which are both publicly available and heavily used in the literature: MotionSense and MobiAct.

- **MotionSense** [29] contains data captured from an 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. 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** [35] records the data from 58 subjects during more than 2500 trials, all captured with a smartphone in a pocket. This dataset includes signals recorded from the accelerometer and gyroscope sensors of a Samsung Galaxy S3 smartphone with subjects performing nine different types of activities of daily living. For our experiments, we only used the trials corresponding to the same activities as the MotionSense dataset.

Both datasets are balanced and contains an equal number of males and females. The datasets are split between training and testing, with 2/3 of trials used for training and validation and 1/3 for testing. These two datasets share similar characteristics, which allows to test the transferability of the models from one dataset to the other. More precisely, the models learned on one dataset can be used to sanitize data from the other dataset. This evaluation corresponds to a more realistic use case and to the best of our knowledge was never considered in previous work related to the sanitization of sensor data.

4.2 Baselines

To assess the performance of DySan, we considered a set of baselines that we detail hereafter. One of these baselines is based on a random forest classifier [13] while the others are based on GANs

[17, 28, 18]. Regarding GAN approaches, authors use an architecture of neural networks slightly different to ours. To provide a fair comparison, we propose to implement their functionalities in our architecture (number of layers, type of CNN, ...). This methodology allows to assess the main characteristics adopted in the baselines without depending on their choice of architecture that can also have an impact on performance.

- **ORF:** To limit the exposure of the data, in [13] the raw data is preprocessed on the user's smartphone and only relevant features are transmitted to the application hosted on the cloud. The relevant features are first identified according to the target application (*e.g.*, activity recognition) and selected either in the temporal or the frequency domain. Originally proposed to avoid users re-identification, we adapt this approach to prevent the inference of the sensitive attribute, namely gender. More specifically, we first detect the features that are the most correlated with the gender before normalizing the features in the frequency domain and removing the features in the temporal domain that are not used for the activity classification.

- **GEN:** Similarly to DySan, GEN (Guardian Estimator Neutralizer) [17] also relies on 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 3.2. More precisely, the first network, a classifier, is learned once on the raw data to identify both sensitive (*e.g.*, the gender) and non-sensitive information (*e.g.*, the activity). Then the second network, an auto-encoder, is also trained only once through a loss function that does not take into account the data distortion. Finally, the model used in the online phase is the same for all users and corresponds to the best set of hyperparameters identified during the training phase. While this solution relies on a neural network architecture slightly different from ours, we implement GEN by using our architecture. However, to evaluate the performance of GEN in a context of transfer learning, we also use their original neural networks (learned on MotionSense[2]) to assess its performance on MobiAct.

- **Olympus:** This approach [28] is similar to GEN with the exception that two different neural networks are used to learn the sensitive attributes and to learn non sensitive information. In addition, these classifiers are trained using sanitized data by following an iterative process similar to DySan described in Section 3.2. However, the loss function does not account for data distortion and the model deployed is the same for all users. While this approach is used for a different objective (*i.e.*, to avoid users re-identification), we adapt it by using our architecture.

- **MSDA:** This solution [18] can be viewed as an evolution of Olympus in which the loss function driving the training of the auto-encoder accounts for data distortion. However, the model used in the online phase is still the same for all users. While this approach was originally developed with a different purpose in mind (*i.e.*, to avoid re-identification), we adapt this solution by using our architecture. This baseline is the closest to DySan but without the dynamic sanitizing model selection in the online phase.

4.3 Evaluation metrics

We evaluated DySan along both utility and privacy metrics, and a couple of system-level metrics.

[2] https://github.com/mmalekzadeh/motion-sense/tree/master/codes/gen_paper_codes

- **Utility:** In our context of physical activity monitoring, the first considered utility metric is the accuracy of a classifier for activity recognition. More precisely, we use the confusion matrix derived by this classifier to measure the number of correct predictions made by the classifier over all predictions made. The value of the accuracy ranges from 0 to 1, in which 1 corresponds to perfect accuracy. In addition, analytics applications monitoring physical activity usually compute and present many estimators to users. To evaluate this aspect, we compute the number of steps detected from the sanitized data and compare it with the number of steps in the raw data. To realize this, we first normalize the raw and sanitized data to compare them in the same range of values, and then compute a Peak Acceleration Threshold [1] from the raw data to estimate the number of peaks. More precisely, we used Adaptiv: An Adaptive Jerk Pace Buffer Step Detection Algorithm (`https://github.com/danielmurray/adaptiv`) for estimating the number of steps detected by the analytics 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). In our case, an accuracy of 0.5 corresponds to a random guess as our dataset is balanced.

- **System-level:** To assess the overhead of operating DySan on a smartphone, we measure both the CPU time spent to sanitize the raw data on the smartphone and the energy consumption over time during a real-time processing of DySan.

4.4 Methodology

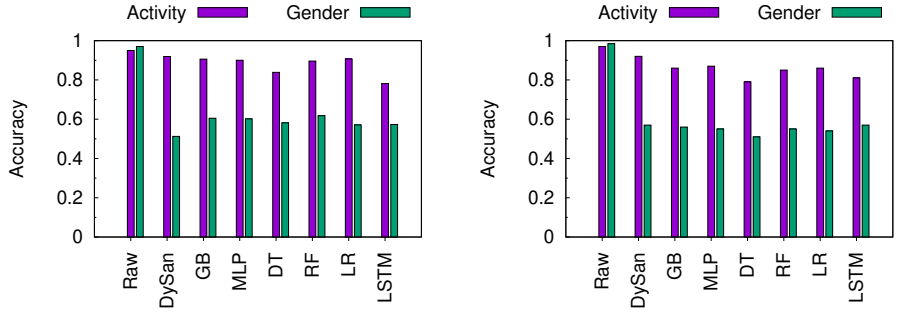
DySan is trained only with the MotionSense dataset while the results reported for MobiAct evaluate the transfer learning (*i.e.*, using sanitizing models trained on MotionSense to sanitize data from MobiAct). In the training phase, we explore a range of values between 0.1 and 0.9 with a 0.1 step for both α and λ, which corresponds to 36 different sanitizing models. The sanitizer models of DySan are trained for 300 epochs and the size of a data batch is set to 256 samples. In the online phase, we select a privacy and utility trade-off focusing primarily on privacy (*i.e.*, ensuring the protection of the gender at the cost of the accuracy). This trade-off is controlled by the parameters x (utility) and y (privacy) (Section 3.3) which are set respectively to 0.1 and 0.9.

The random forest classifier applied during the online phase of DySan uses a feature vector extracted from the raw signal. The choice of these descriptors was made on the basis of an earlier review on effective descriptors for gait recognition [32]. We use 4-fold cross-validation in which the testing set is randomly partitioned into 4 equal sized subsamples.

Reported results correspond to average over 10 repetitions of each experiment. 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 Evaluation

In this section, we report the results obtained for the evaluation of DySan by highlighting important features, namely the good utility and privacy trade-off (Section 5.1), the low distortion of the sanitized data (Section 5.2), the better performances compared to state-of-the-art approaches (Section 5.3), the advantage of dynamically select the best sanitizing model according to the incoming data (Section 5.4), and the limited cost of operating DySan on a mobile (Section 5.5).



(a) MotionSense (b) MobiAct

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

5.1 Utility and privacy trade-off

In this section, we evaluate the capacity of an analytics application to infer the gender of the user and its activity from the sanitized data provided by DySan and sent by the mobile application. We compare the performance of several classifiers that could be used by the analytics application, namely a gradient boosting classifier (GB), a multi-layer perceptron (MLP), a long short-term memory neural network (LSTM), 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 of DySan.

Figure 3 reports the accuracy for both datasets for predicting the gender and the activity with the different classifiers as well when using the raw data. First, the results show that without any protection (*i.e.*, on raw data) the application is able to infer the gender with 98.5% accuracy, In addition, the activity is also inferred from the raw data with 97% of accuracy on average. Secondly, we can observe that DySan successfully decreases the privacy risk with respect to inferring the sensitive attribute while limiting the loss of activity detection. Indeed, with the sanitized data, an analytics application is only able to infer the gender up to 61% and 57% of accuracy, respectively for MotionSense and MobiAct. In term of utility, depending on the classifier, the accuracy of the activity recognition varies between 78% and 92%, which represents only a small drop compared to using the use of the raw data. Remark that the LSTM, a recurrent neural network architecture commonly used for temporal signal, does not provide best results as one could expect.

5.2 Distortion of the sanitized signal

The utility of the sanitized data is not just about the activity recognition but also with respect to more fine-grained information related to the activity. In this section, we demonstrate that DySan keeps relevant information in the signal enabling to conduct further analysis. More precisely, we consider the computation of the number of steps from the signal for MotionSense dataset. Following the step detection method presented in 4.3, Table 1 shows that with DySan the estimation of the number of steps only suffers from a 7% error compared to the raw data. With the different baselines, the sanitized signal appears to be much more noisy and the step detection is greatly impacted with an overestimate number of the steps of more than 64% 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, which prohibits possibility to conduct further analysis.

	Steps	DTW
Raw data	14387	-
DySan	15321 (**+6.49** %)	12.96
GEN	12817 (**-12.25**%)	14.28
Olympus	23658 (**+64.44**%)	156.03
MSDA	18624 (**+29.45**%)	23.37

Table 1: The sanitized signal provided by DySan appears to be less distorted and more useful for step detection than other approaches.

To evaluate the deformation of the signal, we also report the Dynamic Time Warping (DTW) [5] between the raw and the sanitized data from each baseline (Table 1). This metric measures the distortion between two temporal signals. If this metric has a small value then it means that the two signals are quite similar to each other, which is a sign of a small distortion. The results obtained show that the sanitized data produced by DySan is more similar to the raw data compared to other baselines. Similarly to step detection, the sanitization process of Olympus depicts a large data distortion making further analysis of the signal impossible. Other metrics assessing the deformation of the signal (*i.e.*, mean, standard deviation, skewness, kurtosis, and energy) are reported in Appendix B.

5.3 Comparative analysis

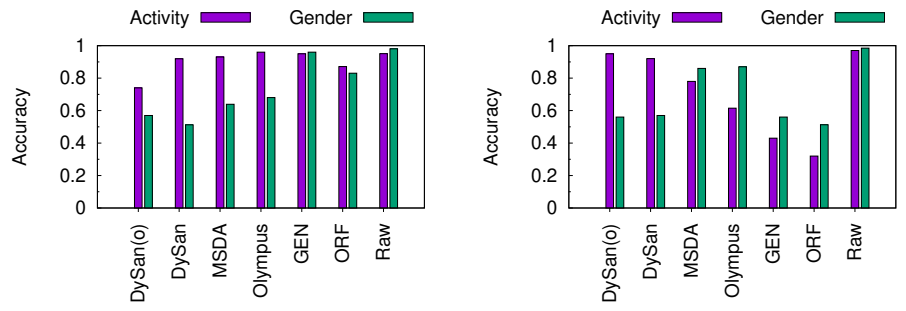
We compare DySan against baseline approaches (Figure 4). Two versions of DySan are given to represent, DySan where the annotations of the activities are known and the online version, DySan(o), where the activities are not given but inferred from the random forest (RF) classifier. The first version has been added for a more fair comparison to state-of-the-art that does not evaluate models as we suggest.

For MotionSense (Figure 4a), the privacy improvement of DySan occurs at the cost of a slightly decrease of utility (gender inference limited to 51% and an activity recognition of 92%). For the online version, which works blindly without annotations, the performance is a little worse, with a gender inference of 57% and accuracy in activity of 75%. This utility mitigation comes from the imperfect accuracy of the random forest classifier used in the online phase to select the best sanitized model. Indeed, to dynamically select the sanitizer model, DySan needs to estimate the model providing the best utility and privacy trade-off with respect to the considered parameters (Section 3.3). To achieve this, DySan relies on a calibration process to build a RF classifier on the raw data used as a reference to predict the current activity performed by the user. This RF classifier provides an average accuracy of respectively 96% and 94% on the activity recognition for MotionSense and MobiAct datasets. While these accuracies are high, an activity wrongly predicted by this classifier leads to a selection of the sanitizer model that does not correspond to the best utility and privacy trade-off.

As depicted on Figure 4b, results for MobiAct show that DySan and DySan(o) outperform other approaches by limiting the gender inference to 55% and 54% while only reducing the accuracy of activity recognition by 2% and 5% compared to using the raw data, respectively. Although GEN and ORF also limit significantly the gender inference, the accuracy of the activity detection is drastically impacted (43% and 32%, respectively).

We detail in Appendix C the accuracy for each activity. From these results, we can observe that the less represented activities are the least well recognized (*i.e.*, the dataset is unbalanced with more data related to the walk).

Results also show the performance improvement provided by each baselines approach based on



(a) MotionSense (b) MobiAct

Figure 4: DySan provides the best privacy protection compared to state-of-the-art approaches at the cost of a slightly smaller accuracy in term of activity detection.

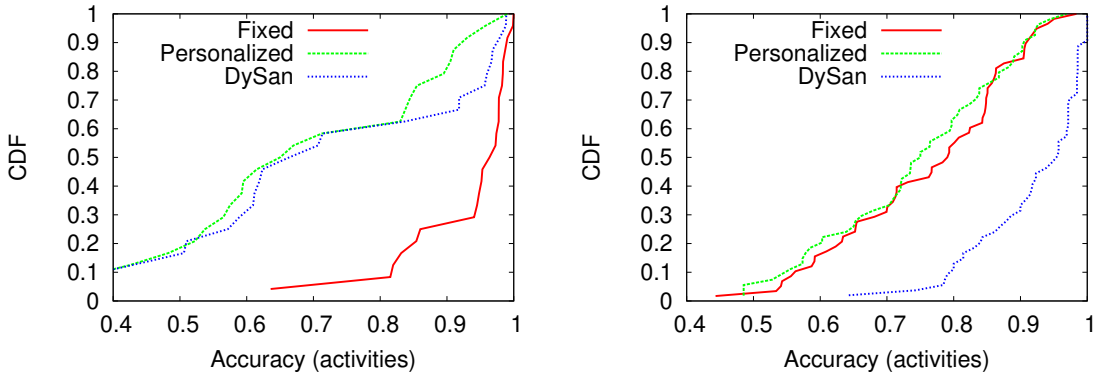
adversarial networks. Specifically, GEN, Olympus and MSDA gradually improve the utility and privacy trade-off. However, our utility analysis (Table 1) shows that the sanitized data is very distorted, which harms the possibility to perform signal processing for further analysis. MSDA integrates the data distortion in its loss function, which leads to less distorted data. This feature improves the quality of signal processing but does not significantly improve the trade-off between utility and privacy compared to Olympus (Figure 4). By dynamically selecting the best sanitizer model for each window of raw data, DySan(o) makes the gender inference close to a random guess while preserving an accurate activity detection.

The results of GEN reported in [17] mention an accuracy of 94% for the activity recognition and 64% for the gender inference for MotionSense dataset compared to 95% and 96%, respectively in our experiments. This difference comes from our implementation that does not used exactly the same neural network setting as the original baseline (only one neural network for both classification tasks versus two neural networks as explained in Section 4.2). However, this difference also tends to assume an over adaptation of the underlying neural network to the considered dataset. This over adaptation is also pointed by the complete different trend for the accuracy provided for MobiAct compared to MotionSense.

As described in Section 3.3, the best sanitizer model is selected according to the definition of the utility and privacy trade-off defined by weight coefficients x and y in the online phase. The reported results correspond to a privacy and utility trade-off controlled by parameters $x = 0.1$ and $y = 0.9$ (Section 4.4). Appendix E depicts the evolution of this trade-off according to these parameters.

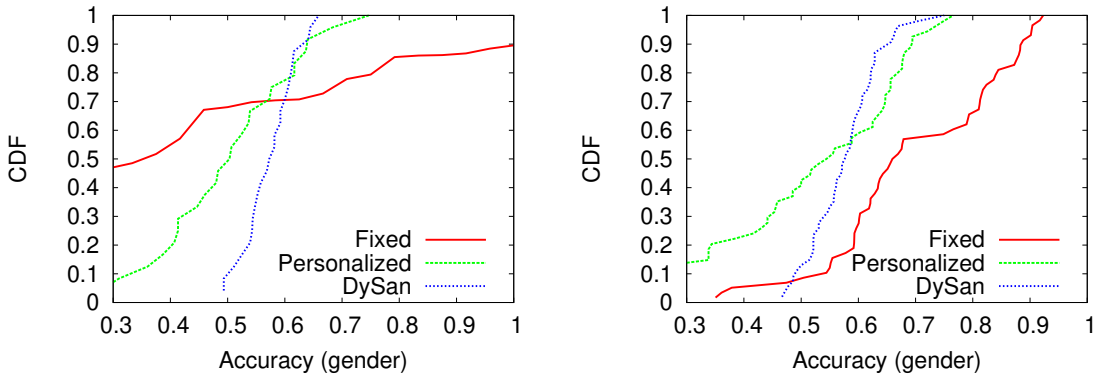
5.4 Dynamic selection of sanitizing model

During the 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 benefit to dynamically adapt the sanitizing model according to the incoming data of each user compared to two static baseline approaches. Firstly, we compute the accuracy for both the gender inference and the activity recognition when the sanitizer model is fixed for all the users. This case represents the behaviors of all comparative baselines where the considered model is the one providing the best performance (*i.e.*, the utility and privacy trade-off) on average for all the users. Secondly, we consider a personalized solution where the sanitizer model is personalized for each user. In this case, the sanitizing model is the one which provides the smallest accuracy in term of gender inference and the best accuracy in term of activity recognition according to the whole models set for a specific user. This solution provides



(a) MotionSense (b) MobiAct

Figure 5: The dynamic sanitizing model selection of DySan significantly improves the activity recognition in case of transfer learning (*i.e.*, MobiAct dataset).



(a) MotionSense (b) MobiAct

Figure 6: By dynamically adapting the sanitizing model for each user according to the incoming data, DySan greatly improved the protection against gender inference (the distribution of the gender accuracy is more centered around 0.5 which corresponds to a random guess).

a sanitizer model personalization but the selected model is static and does not change according to the evolution of the incoming data (and the associated changes in term of performed activity).

We compare these both static solutions against DySan where the considered sanitizer model for each user changes according to the incoming data in order to maximize the utility and privacy trade-off over time. Figures 5 and 6 depict for both datasets the cumulative distribution (*i.e.*, CDF) of the accuracy of the activity recognition and the gender inference respectively, when a fixed, a personalized, and a dynamic sanitizing model is considered. Firstly, results show that the accuracy in both classification tasks is highly heterogeneous over the population of users. This high heterogeneity reflects the fact that a static model is not well adapted for all users or for all activity performed by the user which motivates our dynamic approach.

Specifically, results show that dynamically adapting the sanitizing model significantly improves the activity recognition compared to using a static model in case of transfer learning (*i.e.*, MobiAct dataset, Figure 5b). For MotionSense dataset (Figure 5a), most users benefit from an important accuracy with a static model fixed for all users. This result can be explained by the fact that

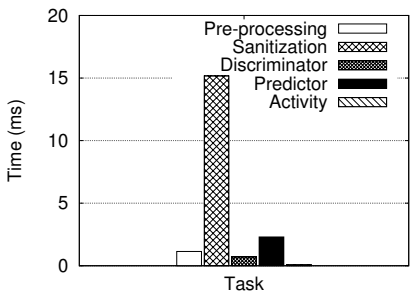


Figure 7: The limited cpu overhead of the sanitation of DySan is compatible to real-time processing on smartphone.

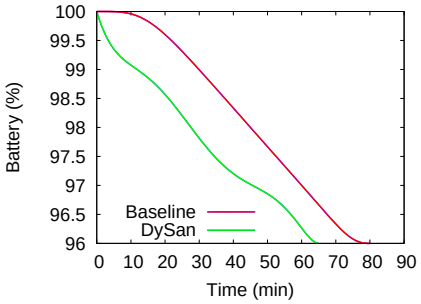


Figure 8: The impact of DySan on energy consumption is limited (1% less battery after 1 hour).

the sanitizing models have been learned with the same users, leading to a learning of the motion characteristics of all the considered users.

For the gender inference, the objective of the sanitizer is to provide an accuracy around 0.5 which corresponds to a random guess for all users. However, results depicted in Figure 6 clearly shows that a fixed model for all users fails to protect against gender inference. Indeed, the distribution reports a wide range of accuracy over the users where it is possible to infer the gender with 80% of confidence for 60% and 20% of the users for MobiAct and MotionSense dataset, respectively. Adopting a personalized sanitizer model for each user decreases the accuracy of the gender prediction compared to a fixed model for all users but the distribution of the accuracy is still large (from 0.3 to 0.75 for MotionSense and from 0.3 to 0.8 for MobiAct). By dynamically adapting the sanitizing model according to the incoming data, DySan greatly improves the protection against gender inference compared to using a fixed model with a sharper distribution centered around 0.5.

These results also show the capacity of DySan to transfer a learning performed on MotionSense to MobiAct (an activity recognition accuracy around 92% on average for a gender accuracy around 57%). For comparison, we evaluated the transfer learning of GEN using the original sanitizing model learned on MotionSense (and publicly available) to MobiAct dataset. In this case of transfer learning GEN provides an accuracy in term of activity recognition and gender detection around 43% and 56%, respectively. This result shows the limited capacity of GEN to transfer learning from MotionSense to another dataset assuming an over adaptation of the underlying neural network and parameters to the considered dataset.

To complete this analysis, we evaluate the variation of the sanitizing model selection of DySan compared to static approaches as well as the number of different models used by DySan for each user in Appendix D. We also quantify the possibility to use the set of selected models as a fingerprint to identify each user in Appendix F.

5.5 Performance as measured on devices

We now evaluate the cost of operating DySan on a smartphone. DySan protects the sensitive attribute while ensuring an accurate activity recognition and minimal data distortion. However, applying the sanitizing at run time on the mobile introduces an overhead. We do not consider the overhead of the learning as it is a one time operation. DySan evaluates multiple sanitizing models (i.e., according to each α and λ hyperparameter explored) before selecting the one that produces the best compromise between utility and privacy. Consequently, the overhead associated

with the sanitizing of raw data depends on the number of considered models.

Figure 7 describes the time (ms) spent by a Xiaomi Redmi Note 7 (equipped with a Qualcom Snapdragon 660 and 3 GB of memory running a java application using Pytorch 1.6) on each task associated to a single sanitizing model of a window of incoming data (i.e., 2.5 seconds of data). Specifically, these tasks include the pre-processing of signal, the sanitizing of raw data, the evaluation of the privacy and the utility on the sanitized data respectively by the discriminator and the predictor, and the classification of the activity performed by the user from the raw data. Excepting the pre-processing which is performed only once for a window of data, the other tasks have to be repeated for each explored sanitizing model. Results show that applying a sanitizing model once spends most of the time while all operations require 19 ms. Considering 20 or 36 sanitizing models increases this time to 366 ms and 658 ms, respectively. Although this processing is compatible with real-time processing (i.e., data processed after each data window), the number of models explored should be chosen to limit the overload.

We also evaluate the impact of running DySan on the energy consumption on the smartphone. Figure 8 reports the decrease in the battery charge over time for a baseline where no operation is performed on the smartphone, and for a real-time processing of DySan (i.e., after each window of raw data, and exploring 36 sanitizing models before to select the best one). In both cases, the screen remained on during the experiment. Results show that DySan consumed 1% more battery after 1 hour, which stays a reasonable energy consumption.

6 Related Works

With the availability of wearable and personal devices, there have been a growing research on the use of collected data for quantifying various aspects of personal life, such as the number of calories consumed, the blood pressure, etc. A growing literature 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 [21] and [27] 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. Our approach is closely related to Gansan [2], however, our framework goes beyond, by considering the data utility with respect to the Predictor network, in addition to the application on motion sensor data. Next, we focus on approaches used as baselines previously in the paper. [17] is the only one which focus on the gender as the sensitive information. [13] and [18] in their case, focus on the re-identification only while [28] 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 autoencoders, the sensitive information can be extracted from the representation produced by the encoder [16], the decoder [28] which also correspond to our approach, or both the encoder and the decoder [18] for data sanitization. Specific to the sensor data generation, SenseGen [3] is a deep learning architecture for protecting users privacy by generating synthetic sensor data. Unfortunately, they did not provide any guarantee on the protection.

To enlarge with other applications protecting sensitive informations using adversarial methods, [7] 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 [8] or identity information in the fingerprints [22].

From a broader privacy perspective, [34] 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 [34], authors in [30] have developed a method for learning an optimal privacy protection mechanism also inspired from GAN, which they have applied to location privacy. In [25], 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-cl and closeness.

Apart from techniques using adversarial approach to protect sensitive information on sensor data, [19] proposes two privacy preserving mechanisms based on clustering algorithms called Hierarchical Agglomerative Clustering to compress amount of disclosed data so that the amount of sensitive information can be reduced. [37] in their case, develop a framework for images data made on wearable cameras that can protect sensitive information such as face, objects or locations thanks to a neural network that detects the sensitive objects which will then be blurred or deleted. Rather than focusing on re-idenfication, [6] investigate what data to share, in such a way that certain kinds of inferences cannot be down. They propose *ipShield* that obfuscate data according to the quantification of an adversary's knowledge regarding a sensitive inference.

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 the activity or to monitor the number of steps. We also show that the dynamic sanitizing model selection of DySan successfully adapts the protection to each user over time according to the evolution of the incoming data. Moreover, we show that the overhead introduces on the smartphone to sanitize the data is compatible with real-time processing while keeping a reasonable energy consumption. Lastly, we compared our approach with existing approaches and demonstrated that DySan provides better control over privacy-utility trade-off.

We investigated the possibility to extend DySan to take into account multiple sensitive attributes. Our preliminary results by adding several discriminators accounted in the loss function of the sanitizer's training are encouraging, however, we are limited by the small size of the available datasets. Indeed, making the sanitizing models more complex requires more data to capture the specificity of each use case.

References

[1] A.Abadleh, E.Al-Hawari, E.Alkafaween and H.Al-Sawalqah, Step detection algorithm for accurate distance estimation using dynamic step length, MDM, 324-327, (2017)

[2] U.Aivodji, F.Bidet, S.Gambs, R.C.Ngueveu and A.Tapp, Agnostic data debiasing through a local sanitizer learnt from an adversarial network approach, ArXiv arXiv:1906.07858, (2019)

[3] M.Alzantot, S.Chakraborty, and M.Srivastava, Sensegen: A deep learning architecture for synthetic sensor data generation, PerCom Workshops, 188–193, (2017)

[4] C.BenAbdelkader, R.Cutler and L.Davis, Stride and cadence as a biometric in automatic person identification and verification, FGR, 372-377, (2002)

[5] D.J.Berndt and J.Clifford, Using Dynamic Time Warping to Find Patterns in Time Series, AAAIWS, 359-370, 12, (1994)

[6] S.Chakraborty, K.R.Raghavan, M.P.Johnson and M.B.Srivastava, A Framework for Context-Aware Privacy of Sensor Data on Mobile Systems, HotMobile, 6, (2013)

[7] C.Jiawei, K.Janusz and I.Prakash, VGAN-Based Image Representation Learning for Privacy-Preserving Facial Expression Recognition, ArXiv arXiv:1803.07100, (2018)

[8] E.Harrison and S.Amos, Censoring Representations with an Adversary, ArXiv arXiv:1511.05897, (2015)

[9] H.Ismail Fawaz, G.Forestier, J.Weber, L.Idoumghar and P.Muller, Deep learning for time series classification: a review, Data Mining and Knowledge Discovery, 33, 4, 917-963, (2019)

[10] M.Feldman, S.A.Friedler, J.Moeller, C.Scheideggera and S.Venkatasubramanian, Certifying and removing disparate impact, KDD, 259–268, (2015)

[11] G.D.Fulk and E.Sazonov, Using Sensors to Measure Activity in People with Stroke, Topics in Stroke Rehabilitation, 18, 6, 746-757, (2011)

[12] J.Han, E.Owusu, L.T.Nguyen, A.Perrig and J.Zhang, Accomplice: Location inference using accelerometers on smartphones, COMSNETS, 1–9, (2012)

[13] T.Jourdan, A.Boutet and A.Frindel, Toward privacy in IoT mobile devices for activity recognition, MobiQuitous, 155–165, (2018)

[14] J.L.Kröger, P.Raschke and T.R.Bhuiyan, Privacy implications of accelerometer data: a review of possible inferences, ICCSP, 81–87, (2019)

[15] S.Lee and K.Mase, Activity and location recognition using wearable sensors, Pervasive Computing, 1, 3, 24–32, (2002)

[16] C.Liu, S.Chakraborty and P.Mittal, DEEProtect: Enabling Inference-based Access Control on Mobile Sensing Applications, ArXiv arXiv:1702.06159, (2017)

[17] M.Malekzadeh, R.G.Clegg, A.Cavallaro and H.Haddadi, Protecting Sensory Data Against Sensitive Inferences, W-P2DS, 2:1–2:6, 6, (2018)

[18] M.Malekzadeh, R.G.Clegg, A.Cavallaro and H.Haddadi, Mobile sensor data anonymization, IoTDI, (2019)

[19] S.Menasria, J.Wang and M.Lu, The purpose driven privacy preservation for accelerometer-based activity recognition, WWW, 21, 1773-1785, (2018)

[20] Y.de Montjoye, C.A.Hidalgo, M.Verleysen and V.Blondel, Unique in the Crowd: The privacy bounds of human mobility, Nature, 3, (2013)

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

[22] W.Oleszkiewicz, P.Kairouz, K.Piczak, R.Rajagopal and T.Trzcinski, Siamese Generative Adversarial Privatizer for Biometric Data, ArXiv arXiv:1804.08757, (2018)

[23] Z.Pan, W.Yu, X.Yi, A.Khan, F.Yuan and Y.Zheng, Recent progress on generative adversarial networks (GANs): A survey, Access, 7, 36322–36333, (2019)

[24] H.Park, H.Chang and H.S.Nam, Use of Machine Learning Classifiers and Sensor Data to Detect Neurological Deficit in Stroke Patients, J Med Internet Res, 19, 4, e120, (2017)

[25] N.Park, M.Mohammadi, K.Gorde, S.Jajodia, H.Park and Y.Kim, Data synthesis based on generative adversarial networks, VLDB, 11, 10, 1071–1083, (2018)

[26] J.Qi, P.Yang, D.Fan and Z.Deng, A survey of physical activity monitoring and assessment using internet of things technology, CIT/IUCC/DASC/PICOM, 2353–2358, (2015)

[27] S.R.Ramamurthy and N.Roy, Recent trends in machine learning for human activity recognition-A survey, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8, 4, (2018)

[28] N.Raval, A.Machanavajjhala and J.Pan, Olympus: Sensor Privacy through Utility Aware Obfuscation, PETS, 5 - 25, 1, (2019)

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

[30] M.Romanelli, C.Palamidessi, K.Chatzikokolakis, Generating Optimal Privacy-Protection Mechanisms via Machine Learning, CoRR, abs/1904.01059, (2019)

[31] F.Schäfer and A.Anandkumar, Competitive gradient descent, NIPS, 7623–7633, (2019)

[32] S.Sprager and M. B.Juric, Inertial sensor-based gait recognition: a review, Sensors, 15, 9, 22089–22127, (2015)

[33] S.Tedesco, J.Barton and B.O'Flynn, A review of activity trackers for senior citizens: Research perspectives, commercial landscape and the role of the insurance industry, Sensors, 17, 6, 1277, (2017)

[34] A.Tripathy, Y.Wang and P.Ishwar, Privacy-preserving adversarial networks, Allerton, 495–505, (2019)

[35] G.Vavoulas, C.Chatzaki, T.Malliotakis, M.Pediaditis and M.Tsiknakis, The MobiAct Dataset: Recognition of Activities of Daily Living using Smartphones., ICT4AgeingWell, 143–151, (2016)

[36] H.Yang J.Yu, H.Zo and M.Choi, User acceptance of wearable devices: An extended perspective of perceived value, Telematics and Informatics, 33, 2, 256–269, (2016)

[37] E.Zarepour, M.Hosseini, S.S.Kanhere and A.Sowmya, A context-based privacy preserving framework for wearable visual lifeloggers, PerCom, 1-4, (2016)

Appendices

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)

	Mean	Std	Skewness	Kurtosis	Energy
Raw	0.81	0.47	1.65	4.81	139.06
DySan	0.68 (-15.9%)	0.77 (+62.9%)	0.40 (-75.7%)	1.28 (-73.5%)	230.87 (+66.0%)
GEN	0.28 (-65.4%)	0.12 (-74.7%)	0.51 (-69.2%)	0.08 (-98.3%)	12.11 (-91.3%)
Olympus	5.40 (+566.4%)	2.52 (+433.1%)	0.61 (-62.8%)	0.29 (-94.0%)	4631.47 (+3230.5%)
MSDA	0.54 (-33.5%)	0.24 (-49.9%)	0.41 (-75.2%)	-0.11 (-102.2%)	51.87 (-62.7%)

Figure 9: Similarities metric between the raw data and the different baselines. Mean, standard deviation (std), skewness, kurtosis, energy are given in percentage of relative error.

B Sanitized Data Distortion

Table 9 gives complementary results concerning the similarity analysis of the data sanitized between the different baselines, with simple quantitative measures. Here the raw measures plus the percentage relative error are given for each baselines. Even if those metrics gives few information about the shapes of the signals, we can still observe that Olympus, the only baselines that does not take into account the distortion of the data during training, is the one that have his measures very far from the raw data. For example the standard deviation is almost fives times higher than the original data showing a very noisy signal.

C Heterogeneous Activity Classification

The accuracy of the classification is not uniform for all activities. Table 2 details the True Positives and False Positives of this classification for DySan on MotionSense dataset. This table also reports the percentage of data in the dataset for each activity. We observe that the accuracy of the classification depends on the performed activity. This heterogeneity is a direct result of the unbalanced classes. Specifically, the walking activity has the highest precision which corresponds to the activity with the largest amount of data, while other activities contains less data and depicted lower good predictions. This difference in terms of good prediction between walking and other activities can also be explained by a calibration of the size window adapted for the walk (see Section 2).

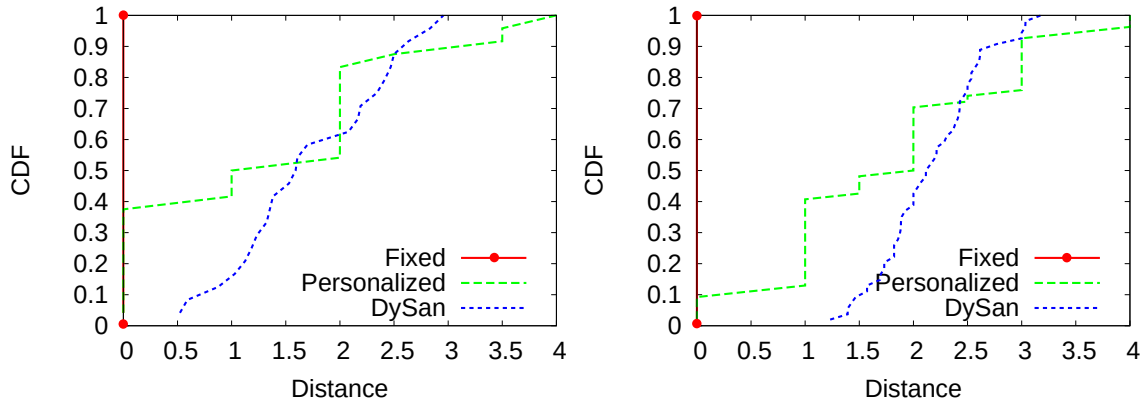
	TP	FP	Precision	Data percentage
Downstairs	221	112	66.4	17.2
Upstairs	223	198	53.0	20.5
Walking	918	74	92.5	44.9
Jogging	216	212	50.5	17.4

Table 2: True Positive, False Positive, Precision and percentage of data for each activity of Dysan (MotionSense dataset).

D Dynamic sanitizing mode selection

We evaluate the variation of the sanitizer model selection of DySan compared to static approaches using either one model fixed for all users or one personalized model for each user. To achieved that, we measure the distance between the hyperparameters α and λ corresponding to the best

privacy and utility trade-off on average for all users (*i.e.*, the model fixed for all users) and the model selected for each user (*i.e.*, a personalized model) or according to the incoming data (*i.e.*, the model dynamically selected by DySan). Figure 10 reports the distribution of this distance for both datasets. Results show that almost 40% of the users of MotionSense dataset have a personalized sanitized model which corresponds to the model providing the best trade-off on average for all users. In addition, for both datasets, results show a large variability in term of distance over all users highlighting the necessity to provide a variety of models to adapt the sanitization.



(a) MotionSense (b) MobiAct

Figure 10: DySan provides a large variability in terms of distance over all users highlighting the necessity to provide a variety of models to adapt the sanitization.

To complete this analysis, we also counted the number of different models used by DySan for each user. Figure 11 depicted for both datasets the distribution of the percentage of all possible sanitized models (36 in our experiment as presented Section 4.4) selected by DySan for each user. Results show a large range of number of different models selected ranging from 20% to 50%. This result show that DySan successfully adapts the sanitization according to the evolution of the incoming data.

E Utility and privacy trade-off selection for DySan

As described in Section 3.3, the best sanitizer model is selected according to the definition of the utility and privacy trade-off defined by weight coefficients x and y. Figure 12 depicts the evolution of the utility and privacy trade-off according to x and y for both datasets.

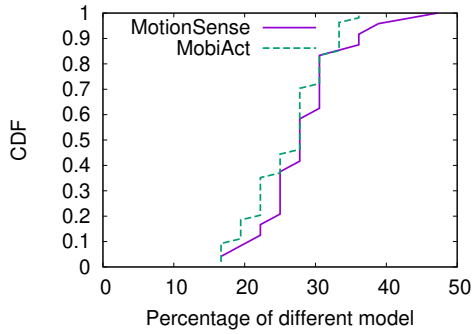
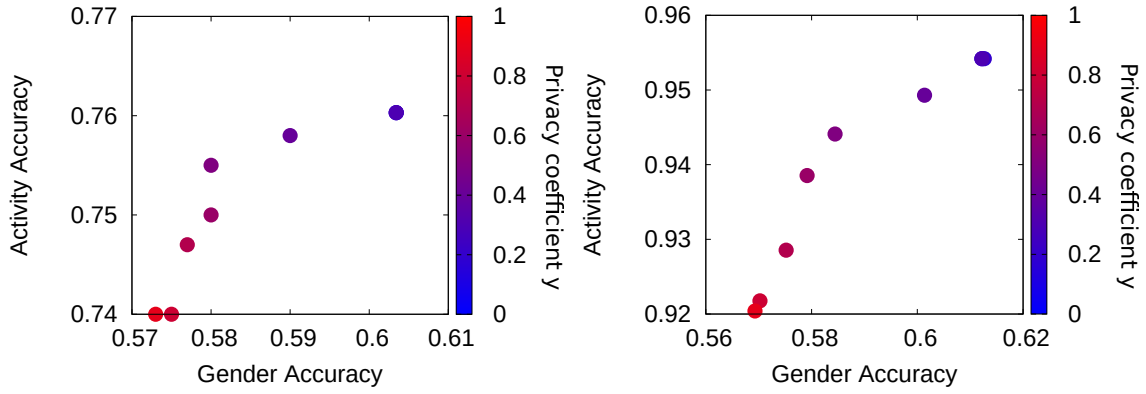


Figure 11: The data of each user is sanitized with a wide variety of models (from 20% to 50% of all the models) showing that DySan successfully adapts the sanitization according to the evolution of the incoming data.



(a) MotionSense (b) MobiAct

Figure 12: The variation of the Privacy coefficient y from 0.1 to 0.9 implies a variation of the trade-off between Utility and Privacy. For both dataset, when y increase, the Privacy increase (the gender accuracy decrease) and the Utility decrease (the activity accuracy decrease)

F Information leakage in model selection

As DySan dynamically selects the sanitizing model to use for each window of incoming data, the set of selected models could be leveraged to identify each user. Indeed, this set of sanitizing models chosen by a user could act as a unique fingerprint. To evaluate this potential information leakage, we quantify the uniqueness following the methodology presented in [20]. More precisely, the uniqueness for each user is estimated as the percentage of 100 random sets of p selected sanitizing models that are unique. Figure 13 reports for MobiAct dataset the distribution of the uniqueness with p (*i.e.*, the size of fingerprint) from 1 to 5 and with different number of sanitizing models available for the selection. As expected, results show that the larger the fingerprint, the more unique the behaviour of a user becomes. However, at least 5 models are needed to have a strong confidence (around 80% of uniqueness) when 36 sanitizing models are exploited. To reduce this uniqueness, a lower number of sanitizing models (i.e., through the hyperparameters values explored in the training phase) should be proposes. Indeed, less choice for model selection

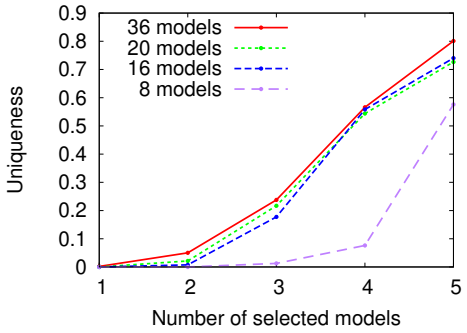


Figure 13: The uniqueness of the selected models remains low for fingerprints with less than 5 models, and depends on the number of available sanitizing models for the selection.

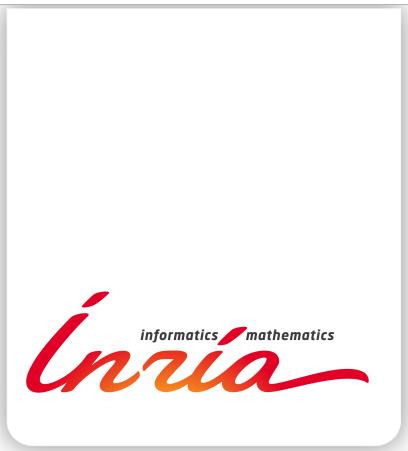
leads to have more users who share common models. Results show that exploiting less available sanitizing models reduces the uniquenes.

Reducing the number of sanitizing models by covering less hyperparameter values limits the achievable space for the utility and privacy trade-off. Consequently, a degradation of the accuracy for both the activity detection and the gender interference is observed. Table 3 presents the performances obtained with different number of sanitizing models available for the selection. Results show that from 36 to 20 sanitizing models, the accuracy in activity recognition decreases by only 3% and increase by 2% the gender inference.

Information leakage in model selection leading to user re-identification is only possible if the adversary is able to characterize each selected sanitizing model from the sanitized data. In this case, the adversary could maintain a fingerprint per user to conduct its re-identification attack. To evaluate this capability, we measure the level of distortion using the Dynamic Time Warping of the sanitized data for each sanitizing model. Over all sanitizing models, our results show a very low standard deviation of the DTW. This low value indicates a small difference in terms of distortion when different sanitizing models are exploited, thus making it difficult for an adversary to identify the selected model from the sanitized data. This re-identification attack consequently seems difficult to achieve.

	Activity accuracy (%)	Gender accuracy (%)
36 models	92	57
20 models	89	59
16 models	88	63
8 models	86	66

Table 3: Reducing the number of sanitizing models available for the selection decreases the accuracy in activity recognition while increasing the accuracy in gender inference.



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