



**HAL**  
open science

## **P-SGD: A Stochastic Gradient Descent Solution for Privacy-Preserving During Protection Transitions**

Karam Bou-Chaaya, Richard Chbeir, Mahmoud Barhamgi, Philippe Arnould,  
Djamal Benslimane

► **To cite this version:**

Karam Bou-Chaaya, Richard Chbeir, Mahmoud Barhamgi, Philippe Arnould, Djamal Benslimane. P-SGD: A Stochastic Gradient Descent Solution for Privacy-Preserving During Protection Transitions. International Conference on Advanced Information Systems Engineering (CAiSE'21), Jun 2021, MELBOURNE, Australia. pp.37-53, 10.1007/978-3-030-79382-1\_3 . hal-03276494

**HAL Id: hal-03276494**

**<https://hal.science/hal-03276494v1>**

Submitted on 2 Jul 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# P-SGD: A Stochastic Gradient Descent Solution for Privacy-Preserving During Protection Transitions

Karam Bou-Chaaya<sup>1</sup>(✉), Richard Chbeir<sup>1</sup>, Mahmoud Barhamgi<sup>2</sup>,  
Philippe Arnould<sup>3</sup>, and Djamel Benslimane<sup>2</sup>

<sup>1</sup> Universite de Pau et des Pays de l'Adour, E2S UPPA, LIUPPA, Anglet, France

{karam.bou-chaaya,richard.chbeir}@univ-pau.fr

<sup>2</sup> LIRIS Laboratory, Claude Bernard Lyon1 University, Lyon, France

{mahmoud.barhamgi,djamal.benslimane}@univ-lyon1.fr

<sup>3</sup> Universite de Pau et des Pays de l'Adour, E2S UPPA, LIUPPA,  
Mont-de-Marsan, France

philippe.arnould@univ-pau.fr

**Abstract.** Advances in privacy-enhancing technologies, such as context-aware and personalized privacy models, have paved the way for successful management of the data utility-privacy trade-off. However, significantly lowering the level of data protection when balancing utility-privacy to meet the individual's needs makes subsequent protected data more precise. This increases the adversary's ability to reveal the real values of the previous correlated data that needed more protection, making existing privacy models vulnerable to inference attacks. To overcome this problem, we propose in this paper a stochastic gradient descent solution for privacy-preserving during protection transitions, denoted P-SGD. The goal of this solution is to minimize the precision gap between sequential data when downshifting the protection by the privacy model. P-SGD intervenes at the protection descent phase and performs an iterative process that measures data dependencies, and gradually reduces protection accordingly until the desired protection level is reached. It considers also possible changes in protection functions and studies their impact on the protection descent rate. We validated our proposal and evaluated its performance. The results show that P-SGD is fast, scalable, and maintains low computational and storage complexity.

**Keywords:** Data privacy · Data protection transitions · Stochastic gradient descent methods · Context-awareness · Internet of Things

## 1 Introduction

The rapid expansion of cyber-physical systems and the technological advances in sensing technologies and data mining techniques have contributed to the tremendous development of smart people-driven applications. These applications tend

to reshape the lives of people in many domains by providing them with advanced services (e.g., increasing comfort, monitoring patients and elderlies). Delivering such services requires collecting and processing massive amounts of data (e.g., location data, health data) to discover underlying patterns and trends. However, privacy concerns hinder the wider use of these data especially as data processing may give rise to serious privacy risks for individuals, such as disclosing their health conditions, habits and daily activities [4]. Consequently, balancing the trade-off between data utility and privacy protection has been subject to intense study in recent years [2, 5, 6, 15, 16]. Current context-aware privacy solutions [2, 13, 16, 20] and personalized privacy solutions [6, 17, 24] aim to maximize the usefulness of data by optimizing the level of protection according to data sensitivity in the current context or/and user preferences. However, these solutions do not consider the effect of temporal correlations between sequential data values on privacy loss. They assign the appropriate level of protection to the data according to the user's context (e.g., privacy risks involved) or/and preferences. Nonetheless, continuously balancing the protection levels without considering previous protection patterns may entail temporal privacy leakage. In particular, this leakage occurs when the protection level significantly decreases, which widens the precision gap between prior/subsequent correlated data and makes subsequent data more precise. The large gap in precision improves the capabilities of an adversary, when using advanced mining techniques, to reveal the real values of prior data pieces that required more protection. This makes existing privacy-preserving solutions vulnerable to data inference attacks. A data inference attack is a data mining attack in which adversaries are capable of estimating/infering real values of protected data with high confidence. One of the possible solutions to address this vulnerability is to integrate a gradient descent mechanism at the protection descent phase. This helps to reduce the precision gap between sequential protected data when downshifting the protection level. Gradient descent is a general paradigm that underlies algorithms for solving optimization problems [8]. It has been widely applied to many fields such as location-based applications for predicting moving destination [23], differential privacy [18], and personalized privacy [14]. Nonetheless, to the best of our knowledge, there has not been any work on securing data protection transitions using gradient descent.

The implementation of a gradual descent process for the protection level is challenging, as the corresponding deviation rate depends on several dynamic factors. First, the temporal correlations between sequential data values, which may vary from sequence to sequence as the data can be generated in regular or irregular time series. Second, the dynamicity of the protection function chosen by the system to be executed on data values. In fact, the system can change the data protection function at the protection transition phases with a view to improving protection, reducing the cost of protection (i.e., computational costs), or due to errors in function operations. However, the protection functions can share similarities in their operations (e.g., generalization and random-noise functions add noise to the real value of data), making it important to consider their

dependence and its impact on the protection deviation rate. What makes it more challenging is the need for a fast and low complex solution, which makes it re-usable by various privacy models, including those offering real-time protection, and operational even for resource-constrained devices. Finally, the solution should follow a non-deterministic descent to avoid revealing the deviation rate by adversaries in case of repeated descent patterns.

To answer these challenges, this paper introduces P-SGD, a stochastic gradient descent solution for privacy-preserving during protection transitions. P-SGD empowers existing privacy models against data inference attacks, by minimizing the precision gaps of sequential protected data values during the protection descent phase. It follows an iterative process to identify the appropriate protection level to be assigned to each transitional data until the targeted level is reached. Computed protection levels consider the temporal dependencies between data values and the dependencies between protection functions (in case of change). Our solution is generic (i.e., it handles attributes with different data types and formats), and supports simultaneous reasoning over multiple attributes. We validated our proposal and evaluated its performance. Results show that P-SGD is fast, scalable, and maintains low computational and storage complexity.

The rest of the paper is organized as follows. Section 2 presents the motivating scenario. Section 3 details our proposal and provides formal definitions of the key terms used. Section 4 outlines the experiments and results. Section 5 discusses existing privacy models and data protection functions. Finally, Sect. 6 concludes the paper and discusses future research directions.

## 2 Motivating Scenario

To motivate our proposal, we investigate a real-life scenario of Alice, a cancer patient who shares her location data with a remote monitoring platform for cancer care. Alice shares also her location data with several other service providers through applications and social media platforms to benefit from their services (e.g., Facebook, Google Maps). The trust relationship between Alice and the providers may vary greatly due to many factors, such as the privacy risks associated with the sharing of data, the sensitivity of her context (e.g., private meeting), or the third parties with whom her data is communicated. Alice may therefore want to protect her privacy in some situations but without completely losing associated services. To do so, she uses a context-aware privacy-preserving system that optimizes the data protection according to her contexts and preferences. Consider that Alice has a medical appointment at the Belharra-Ramsay center for her cancer treatment. She takes the road from her home to the treatment center. However, locating Alice in the cancer center can entail the disclosure of her health conditions, which involves privacy concerns for her. Accordingly, assume that the privacy system increases data protection to 80% when Alice arrives at the center, and then shifts the level of protection to 20% when she leaves. The system protects sensed data using a generalization-based

protection function. In the following, three cases are considered to highlight the impact of the second protection transition phase (from 80% to 20%) on privacy loss.

In case-1, represented in Fig. 1, the system shifts the level of protection to 20% and continues to perform the same protection function on generated data (i.e., the generalization function). The location data are generated at a regular time interval. When processing and analyzing protected data values, an adversary can notice a significant gap in the level of precision between transitional/correlated data (see in Fig. 1). The precision gap limits the range for estimating previous user locations where protection was critical (e.g., Alice’s presence in the medical center), which entails privacy problems. This consequently underlines the need for a gradual descent in the protection level in order to overcome vulnerabilities that may arise during protection transitions.

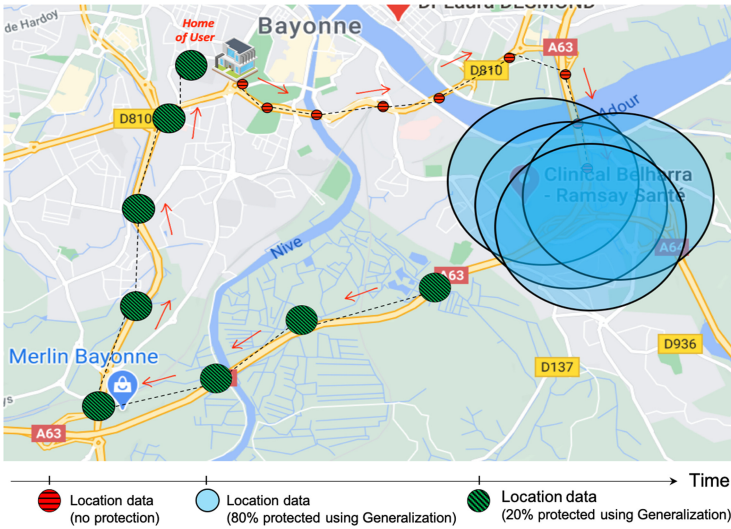


Fig. 1. Case-1

As previously mentioned in Sect. 1, the system can change the protection function to be executed on data at the protection transition phase. In case-2, illustrated in Fig. 2, the system changes the function when the protection level shifts to 20%, and adopts a randomization-based function that adds random noise to the real location positions. However, the generalization and randomization functions share similarities. They both add noise to the data, which makes them dependent, and the privacy issues related to lowering the protection level persist. This highlights the need to examine dependencies between protection functions and their impact on the protection deviation rate.

In the previous two cases we considered regular time series data. However, data can be also collected in irregular time series, i.e., the data

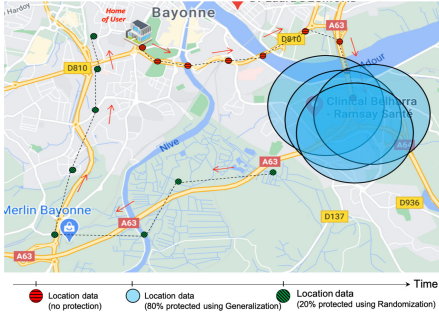


Fig. 2. Case-2

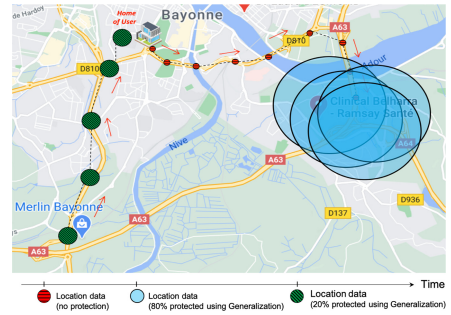


Fig. 3. Case-3

collected follow a temporal sequence, but the measurements may not occur at regular time intervals. For instance, case-3 assumes that after leaving the medical center, the system has stopped sharing (protected) location data only for a specific time interval due to loss of connectivity with the GPS sensor (cf. Fig. 3). When data sharing started again, the temporal distance between the last data shared and the current one has already exceeded the temporal granularity of the attribute (i.e., location). The two data pieces are thus independent and the adversary will not be able to link previous and subsequent location patterns. It is thereby important to measure the temporal correlations between sequential data and study its impact on data protection. Consequently, building up the gradient descent solution requires addressing the following challenges:

- **Challenge 1. *Data Dependency*:** How to track and measure the temporal dependencies of sequential data values and study their impact on the protection descent rate?
- **Challenge 2. *Protection Function Dependency*:** How to compute the similarity between transitional protection functions (in case of change) and adjust the downshifting mechanism accordingly?
- **Challenge 3. *Non-deterministic Solution*:** The protection level can fluctuate between two same values for several transitions. This may entail the disclosure of the deviation rate by adversaries if the executed process is deterministic (cf. Fig. 4). The solution should therefore be non-deterministic to overcome the vulnerabilities arising from repeated transition patterns.
- **Challenge 4. *Scalability & Efficiency*:** The solution must be scalable, i.e., handles simultaneous reasoning over an increasing number of attributes. Moreover, it should maintain computational and storage efficiency, which increases its re-usability to also include privacy models subject to real-time constraints, and makes it operational on a variety of devices, including those with limited resources.

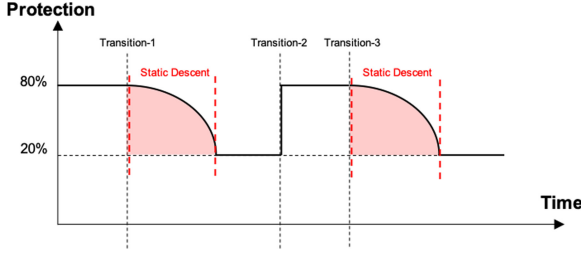


Fig. 4. Repeated protection transition patterns

### 3 P-SGD Proposal

Current privacy-preserving models, such as context-aware and personalized privacy models, enable data protection variation according to the individual’s needs or/and situations (e.g., privacy risks involved in the data sharing, preferences) in order to optimize the balancing of data utility-privacy. However, these models perform direct shifting of the data protection level, which may lead in certain cases to temporal privacy leakage due to data correlations. In particular, the data privacy leakage occurs when significantly decreasing the level of protection, creating a significant gap in the level of precision between previous and subsequent data. This increases the ability of an adversary to reveal the real values of previous correlated data that needed more protection, entailing privacy concerns for the individual. To overcome this vulnerability, we propose P-SGD, a privacy-based stochastic gradient descent solution that operates during protection descent phases to minimize precision gaps between sequential protected data values. P-SGD addresses the challenges mentioned in Sect. 2. It features an iterative protection descent process that identifies the appropriate level of protection to be assigned to each data prior to its release until the final level is reached (i.e., the lowest desired level). The proposed solution supports attribute diversity, i.e., it handles attributes with different data types and formats (e.g., scalar data such as location and temperature data, as well as multimedia data such as camera recordings). This makes it therefore generic and compatible with numerous existing privacy models in different application domains. The P-SGD process can be plugged into the privacy model, as shown in Fig. 5, to provide an additional layer of protection against inference attacks. Let  $u$  denotes the user (or data subject as defined by the General Data Protection Regulation [21]). In what follows, we formally define an *attribute* and a *data node*.

**Definition 1.** (Attribute). Let  $A$  be the *set of spatio-temporal attributes*  $\{a_1, a_2, \dots, a_n\}$  shared by  $u$  with data consumers.  $a \in A$  is defined as follows:

$a : \langle desc ; access ; source ; D_{consumer} ; \tau ; Log \rangle$ , where:

- *desc* is the textual description of  $a$  (e.g., location, heart-rate, temperature)
- *access*  $\in \{r; r/w\}$  denotes the access rights of the privacy model to the data values of  $a$ , which can be read or read/write
- *source*  $\in DN$  is a *data node* (cf. Definition 2) expressing the data source from which the data of  $a$  is collected (e.g., GPS sensor)

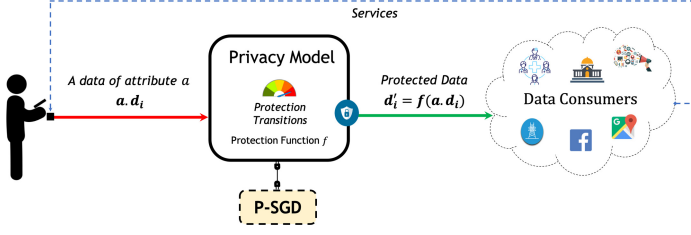


Fig. 5. Integration of P-SGD

- $D_{consumer}$  is the set of data consumers (i.e., service providers and third parties) with whom  $a$  is shared, such that:

$$D_{consumer} = \{ dc_1 ; dc_2 ; \dots ; dc_n \} \cup \{\perp\}, \text{ where:}$$

- $dc_i \in DN$  is a *data node* expressing a data consumer
- $D_{consumer} = \emptyset$  indicates that data consumers are unknown
- $D_{consumer} = \{\perp\}$  denotes that  $a$  is a public attribute
- $\tau$  denotes the standard time period during which two data values of  $a$  are said to be time-dependent
- $Log = \{\langle d ; M \rangle\}$  is the set of data values of  $a$  where:
  - $d$  denotes the data value
  - $M$  is the set of metadata characterizing  $d$  (e.g., time/location of capture, data-type, format) ■

**Definition 2.** (Data Node). Let  $DN$  be the set of source/destination *data nodes*  $\{n_1, \dots, n_n\}$ . Source nodes are data sources from which the data is collected. Destination nodes are data consumers with whom the data is shared.

$$\forall n \in DN, n : \langle desc ; id \rangle, \text{ where:}$$

- *desc* is the textual description of  $n$  (e.g., gps-sensor, health-provider)
- *id* is the identity of  $n$ , expressed as a uniform resource identifier (URI) ■

---

*Example 1.* The location attribute shared by Alice can be represented as follows:

- $a_1 : \langle desc : Location ; access : r/w ; source : sensor-1 ; D_{consumer} = \{prov-1\} ; t_{gran} : 86400 ; t_{gen} : 1 ; Log = \{\langle (-33.0534, 16.3103) ; M_1 \rangle\}$
- *sensor-1*:  $\langle desc : GPS-sensor ; id : 46.89.1.47 \rangle$
- *prov-1*:  $\langle desc : Healthcare-provider ; id : 58.17.37.23 \rangle$
- $M_1 : \{t_{capture} : 15:17:00 ; d_{type} : float ; d_{format} : (longitude, latitude)\}$

We consider here that  $t_{gran}$  is provided as an input parameter. The challenges of identifying the temporal granularity of attributes will be explored in future work.

---



P-SGD also supports protection function diversity. In fact, existing protection functions vary from data anonymization, data perturbation using noise addition, privacy-aware access control to encryption (cf. Sect. 5). Each of these functions achieves differently the desired protection level. We provide in what follows formal definitions of a *protection function* and *protection level*.

**Definition 3.** (Protection Function). A *protection function*,  $f \in PF$ , is a protection method performed by the privacy model on data values of an attribute  $a \in A$  prior to their release to consumers.  $f \in PF$  is formalized as follows:

$f : \langle name ; class ; Feature ; Param \rangle$ , where:

- *name* denotes the textual name of  $f$  (e.g., generalization, random-noise)
- *class* represents the class to which  $f$  belongs, such that:

$class \in \{\text{noiseAddition ; anonymization ; accessControl ; encryption}\}$

- *Feature* is the set of features characterizing  $f$ , including at least: *cost*, the computational cost of  $f$  in terms of processing time and memory overhead
- *Param* represents the set of input parameters of  $f$ , including at least:
  - $A' \subseteq A$  is the set of attributes on which  $f$  is performed
  - $P$  is the set of protection levels to reach for all  $a \in A'$  ■

**Definition 4.** (Protection Level). A *protection level*,  $p$ , expresses the amount of protection to be achieved for the data values of an attribute  $a \in A$ .  $p$  is probabilistic with a value between  $[0, 1]$ , where 0 means that data is shared without any protection, and 1 means that data is not shared. A value between 0 and 1 indicates the level of protection that should be reached when executing a *protection function*  $f \in PF$  on the data of  $a$ . Knowing that the way to achieve  $p$  depends on the selected *protection function*. ■

A stochastic gradient descent method is generally defined as an iterative method for optimizing an objective function with suitable smoothness properties [1]. It has been widely adopted mainly for high-dimensional optimization problems as it reduces the computational burden, achieving faster iterations in trade for a lower convergence rate. This agrees with our needs listed in Challenge 4. Consequently, we detail in the following our proposed P-SGD method. According to Fig. 6, let:

- $p_i^{target}$  refers to the *targeted protection level*, i.e., the next protection level specified by the privacy model for data of attribute  $a_i \in A$ . This level indicates the target level that must be reached in order to complete the P-SGD process
- $p_i^{old}$  denotes the level of protection of the previous data value of attribute  $a_i \in A$
- $p_i^{current}$  expresses the protection level to be assigned to the current data value of attribute  $a_i \in A$ , such that  $p_i^{current} \in [p_i^{target}; p_i^{old}]$

The iterative process followed by P-SGD is thus defined by the following formula:

$$p^{current} = p^{old} - \eta \nabla \quad , \text{ where:} \quad (1)$$

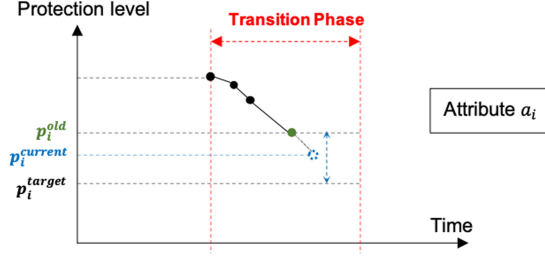


Fig. 6. P-SGD process

- $\eta$  represents the deviation rate of the protection level (the quantification of  $\eta$  is provided in the following subsection)
- $\nabla \in [0; 1]$  expresses the random noise added to  $\eta$

We consider in this study that attributes are independent. The P-SGD process is thus performed on the data values of each attribute separately. In order to track and measure the correlations in sequential data and the dependencies between their associated protection functions (cf. Challenges 1 and 2), we define a **transition matrix**,  $\mathbf{Trans}$ , that contains only the properties of the last data value ( $\mathbf{d}_i^{old}$ ) of each shared attribute  $\mathbf{a}_i \in \mathbf{A}$ . We store only the properties of the last data values since the process operates iteratively. This helps reduce storage overhead and allows for scalability in attribute number (cf. Challenge 4).  $\mathbf{Trans}$  denotes therefore the cache, and can be represented as follows:

$$\mathbf{Trans} = \begin{bmatrix} t_1^{old} & p_1^{old} & f_1^{old} \\ t_2^{old} & p_2^{old} & f_2^{old} \\ \vdots & \vdots & \vdots \\ t_n^{old} & p_n^{old} & f_n^{old} \end{bmatrix}, \text{ where:} \quad (2)$$

- $t_i^{old}$  denotes the time of capture of  $\mathbf{d}_i^{old}$  of attribute  $\mathbf{a}_i$
- $f_i^{old}$  is the protection function associated to  $\mathbf{d}_i^{old}$  of attribute  $\mathbf{a}_i$

### 3.1 Deviation Rate Quantification

$\eta$  depends on (1) the temporal dependency of previous and current data values of  $\mathbf{a}_i$ , i.e.,  $\mathbf{d}_i^{old}$  and  $\mathbf{d}_i^{current}$ ; and (2) the level of dependency of their related protection functions, i.e.,  $f_i^{old}$  and  $f_i^{current}$ .

**Definition 5.** (Time Dependency of Data). Let  $depend_t$  denotes the temporal dependency score of two data values,  $\mathbf{d}_i^{old}$  and  $\mathbf{d}_i^{current}$ , of an attribute  $\mathbf{a}_i \in \mathbf{A}$ .  $depend_t$  has a value between 0 and 1, where 0 means that the data are time-independent, and 1 means that the data are fully dependent (time-wise), which typically occurs only when  $t_i^{old}$  and  $t_i^{current}$  are similar. The higher the temporal distance between the two data values is, the lower their time dependency is. The

two data values are said to be time-dependent only if their temporal distance is less than the *standard time period* of their attribute  $a_i$  (i.e.,  $a_i.\tau$ ).  $depend_t$  is therefore computed as follows:

$$depend_t(d_i^{old}, d_i^{current}) = \begin{cases} 1 - \frac{t_i^{current} - t_i^{old}}{a_i.\tau} & \text{if } (t_i^{current} - t_i^{old}) \leq a_i.\tau \\ 0 & \text{otherwise} \end{cases} \quad \blacksquare$$

**Definition 6.** (Protection Function Dependency). Let  $f_i^{old}$  and  $f_i^{current}$  denotes two protection functions.  $f_i^{old}$  and  $f_i^{current}$  are said to be dependent only if their similarity score is above or equal 0.

$$sim(f_i^{old}, f_i^{current}) \rightarrow [0; 1], \text{ where:}$$

- **sim** is a unit similarity function that checks the exact matching between the classes and the lists of features of the two protection functions, and returns a value between 0 and 1, such that:

$$sim(f_i^{old}, f_i^{current}) = 1 \text{ only if:}$$

$$f_i^{old}.class = f_i^{current}.class \text{ and } f_i^{old}.Feature = f_i^{current}.Feature \quad \blacksquare$$

The P-SGD process will be therefore executed only if the sequential data values are dependent and their associated protection functions are also dependent (i.e.,  $depend \neq 0$  and  $sim \neq 0$ ). The higher the temporal distance between previous/current data is, the lower is their time dependency, and the higher is  $\eta$  (i.e., the larger can be the protection gap between the two data). As well, the higher is the similarity between protection functions, the lower is  $\eta$ . Accordingly,  $\eta$  is quantified as follows:

$$\eta = c_i \times sim(f_i^{old}, f_i^{current}) \times depend_t(d_i^{old}, d_i^{current}) \quad (3)$$

- $c_i \in \mathcal{C}$  is a system parameter that expresses the maximum deviation value of data protection for attribute  $a_i \in \mathcal{A}$ . It therefore controls the convergence speed of the protection level towards  $p_i^{target}$
- $sim(f_i^{old}, f_i^{current}) \rightarrow ]0; 1]$  is the similarity score
- $depend_t(d_i^{old}, d_i^{current}) \in ]0; 1]$  is the temporal dependency score

### 3.2 P-SGD Algorithm

We present here the reasoning algorithm of our solution.

**Algorithm 1.** Presents the algorithm of our P-SGD solution that takes as input the concerned attribute  $a$ , the properties of the current data value (i.e.,  $t^{current}$  and  $f^{current}$ ), and the targeted protection level  $p^{target}$ . It outputs the calculated protection level to be assigned to the current data value, i.e.  $p^{current}$ . The process starts by computing the dependency score of previous/current data values and the similarity score of associated protection functions (lines 3–4). If data or/and associated functions are independent (line 12), the gradual descent process is not executed, and the protection level is downshifted directly to  $p^{target}$  (line 13). Else, this means that data and associated functions are dependent (line 5). The process calculates the random noise  $\nabla$  to be appended to  $\eta$ , the value of  $\eta$ , and then the value of  $p^{current}$  (lines 7–9). It checks after the validity of the  $p^{current}$  value (lines 10–11). Finally, the properties of the relevant attribute are updated in the transition matrix  $Trans[][]$  (line 14) and the process is ended.

---

**Algorithm 1: P-SGD Process**


---

```

Input:  $a, c, t^{current}, f^{current}, p^{target}$ ; // attribute, default deviation value, time of
capture and protection function of  $d^{current}$ , and the targeted protection level;
Output:  $p^{current}$ ; // the protection level to be assigned to  $d^{current}$ ;
1 Variables:  $Trans[][]$ ,  $depend_t$ ,  $simScore$ ,  $\nabla$ ,  $\eta$ ; // transition matrix, dependency
score of data, similarity score of prot-functions, random noise and deviation rate;
2 begin
3    $depend_t := 1 - \frac{t^{current} - Trans[a][0]}{a \cdot \tau}$ ; //  $Trans[][0]$  is the  $t^{old}$  column of  $d^{old}$  values;
4    $simScore \leftarrow sim(f^{current}, Trans[a][2])$ ; //  $Trans[][2]$  is the  $f^{old}$  column
associated to  $d^{old}$  values;
5   if ( $depend_t \neq 0$  AND  $simScore \neq 0$ ) then
6     // dependent data values and dependent protection functions;
7      $\nabla \leftarrow randomNumber(0, 1)$ ; // returns a random value between 0 and 1;
8      $\eta := c \times simScore \times depend_t$ ;
9      $p^{current} := p^{old} - \eta \nabla$ ;
10    if ( $p^{current} \leq p^{target}$ ) then
11       $p^{current} := p^{target}$ ; // check the validity of the calculated  $p^{current}$  value ;
12    else
13       $p^{current} := p^{target}$ ; // data values or/and protection functions are independent;
14     $Trans \leftarrow updateTransMatrix(a, t^{current}, p^{current}, f^{current})$ ;
15 return  $p^{current}$ 

```

---

This paper presents only the pseudo-code of the main P-SGD process due to space limitations. The pseudo-codes of the aforementioned functions are detailed in the prototype source code provided in Sect. 4.

## 4 Experimental Validation and Evaluation

In order to implement and validate our approach, we developed a Java-based prototype (the source code is available online through this link<sup>1</sup>). We illustrate in the following the prototype operation by considering the scenario of Alice

<sup>1</sup> <https://spider.sigappfr.org/research-projects/psgd/> (P-SGD Prototype).

described in Sect. 2. We focus on the second protection transition (i.e., from 80% to 20%), and assume that the protection function remains unchanged. We repeated the descent process three times to emphasize the non-deterministic nature of the solution in the case of repeated transition patterns (cf. Challenge 3). We consider here regular time series data with a data generation time of 1s, and we fix  $c$  at 0.5 (i.e., the maximum protection deviation is 50%). As shown in Fig. 7, the proposed P-SGD process is able to iteratively and gradually decrease the protection level until reaching the targeted one (i.e., 20%), with an average of 35 ms per iteration. The deviation pattern varied between the three similar transition cases, as well as the number of data values required to achieve protection convergence (7 for transitions 1–2 and 8 for transition 3). This is due to the noise value associated with the deviation rate (i.e.,  $\nabla$ ), which varies randomly with each iteration.

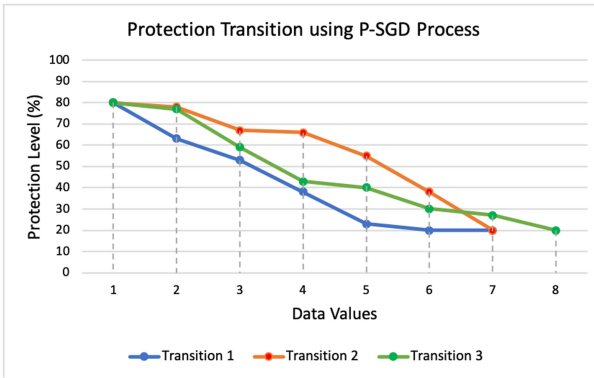


Fig. 7. Securing protection transitions using the P-SGD process

#### 4.1 Performance Evaluation

The objective here is to evaluate the approach’s effectiveness, in terms of performance, to operate in different scenarios. The approach is said to be effective if it meets the needs outlined in Challenge 4: (1) fast; (2) scalable (i.e., supports multi-attribute handling); and (3) low-complex in time and space (i.e., in terms of memory overhead and storage). To do so, we start by considering two cases to study the impact of the following two metrics on performance: (i) the complexity of the protection functions dependency; and (ii) the number of attributes handled simultaneously. Then, we formally study the storage complexity of the proposal. The performance is evaluated based on two criteria: the total execution time of one iteration and the memory overhead. The tests were conducted on a machine equipped with an Intel i7 2.80 GHz processor and 16 GB of RAM. The chosen execution value for each scenario is an average of 10 sequenced values.

**Case 1:** We consider two dimensions to study the complexity of the functions dependency: the first increases the number of features and the second increases the diversity in features between the two functions. We execute the P-SGD process 13 times, taking into account the following number of features for each iteration: 1, 5, 10, 15, 20, 30, 40, 50, 60, 70, 80, 90 and 100. For each of these scenarios, we consider three sub-scenarios where we vary respectively the percentage of diverse features from 0%, 50% to 100%. As shown in Fig. 8 and 9, the number and diversity of the features have no impact on the function dependency procedure, and thus on performance. This is due to the fact that the procedure verifies only the exact matching of the features' names and values. The process is executed in all scenarios with an average time of 35 ms and 10 MB of RAM usage.

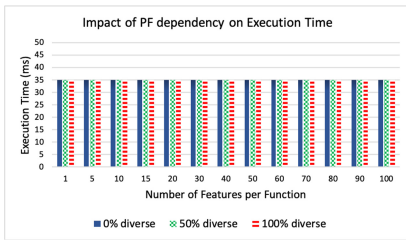


Fig. 8. Case-1: execution time

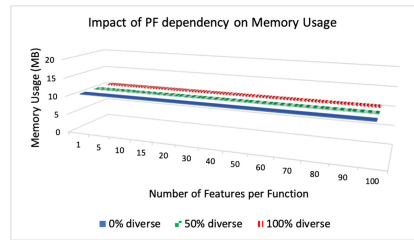


Fig. 9. Case-1: memory usage

**Case 2:** To study the impact of multi-attribute handling, we incorporate multithreading features in order to perform parallel execution of the process on an increasing number of attributes. We consider the following number for each iteration: 1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100. Figure 10 shows that increasing the number of attributes has a quasi-linear impact on the total execution time, with an average time of 35 ms for 5 attributes and up to 100 ms for 100 attributes. The RAM usage remains constant with an average of 10 MB (cf. Fig. 11). This highlights the importance of integrating a low-cost transition matrix.

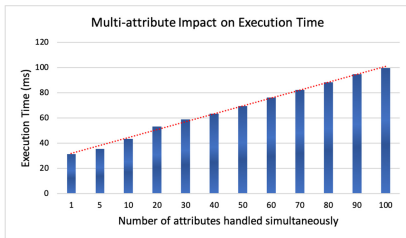


Fig. 10. Case-2: execution time

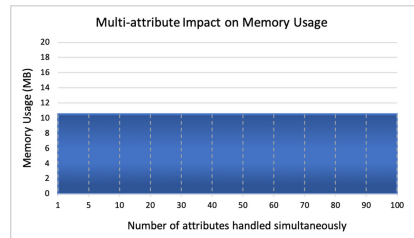


Fig. 11. Case-2: memory usage

**Theorem 1.** The P-SGD process maintains low storage complexity.

**PROOF.** Let  $n$  denotes the maximum number of *attributes* that could be shared by the user with data consumers. As previously mentioned in Sect. 3, the solution stores only the three properties of the last data value for each attribute in *Trans*, and the values of  $c_i \in \mathcal{C}$ , resulting in a linear storage complexity of  $O(4n)$ . However, the number of *attributes* shared by the user will not practically exceed 100, which makes the storage complexity low.  $\square$

**Discussion.** The experiments conducted show that P-SGD is scalable and efficient in time and space (cf. Challenge 4). The solution is able to maintain effective performance in different scenarios, including worst-case ones. This increases its re-usability to also include privacy models that require real-time reasoning, and allows it to operate on a variety of devices, including resource-constrained ones.

## 5 Related Work

Several approaches have been proposed in the literature to address the challenges of security and privacy in the fields of pervasive Internet of Things (IoT) environments, also known as connected environments. However, to the best of our knowledge, this is the first work to tackle the problem of preserving user privacy against data inference attacks during protection transitions. Therefore, we discuss in this section existing privacy-preserving models to which our solution could be connected. Then, we introduce a classification of existing protection functions that could be used by these models.

### 5.1 Context-Aware and Personalized Privacy Models

Balancing data utility-privacy has received extensive attention in the last decade. Existing approaches vary from context-aware to personalized privacy-preserving. Bou-Chaaya et al. [2] introduced CaPMan, a user-centric context-aware model for privacy management in connected environments that meets current privacy standards (i.e., Privacy by Design and ISO/IEC 27701 standards). Matos et al. [13] proposed a context-aware security approach, that provides authentication, authorization, access control, and privacy-preserving to fog and edge computing environments. Gheisari et al. [7] introduced a context-aware privacy-preserving approach for IoT-based smart city using Software Defined Networking. Sylla et al. [20] presented a context-aware security and privacy as a service (CASPaaS) architecture to inform the user about the contextual risks involved. Gao et al. [6] proposed a personalized anonymization model for balancing trajectory privacy and data utility. Qiu et al. [17] provided a semantic-aware personalized privacy model that studies user requirements and location’s privacy sensitivity to adapt the trajectory construction accordingly. Xiong et al. [24] proposed a personalized privacy protection model based on game theory and data encryption.

## 5.2 Privacy Protection Functions

Existing functions for data protection vary from data perturbation (anonymization and noise-addition), to data restriction (access control and encryption). On this basis, we introduce a new classification of these functions based on their perspective for data protection. The first category consists of data perturbation functions, which comprises two sub-categories: anonymization and noise-addition. Anonymization functions focus on masking user’s identity from generated data by removing explicit identifiers, and decreasing the granularity of quasi-identifiers using operations such as generalization and suppression (e.g., k-Anonymity [19], l-Diversity [12], CASTLE [3]). Noise-addition functions focus on perturbing original data values instead of protecting the owner identity, and that by injecting additive noise (e.g., Generalization [10], Random-noise [9]). The second category regroups data restriction functions that aim at limiting data use by blocking access or encrypting inputs. This category is composed of two sub-categories: access control and encryption. Access control functions (e.g., [11]) achieves privacy protection through authorization models and access control policy operations. Encryption functions applies encryption mechanisms on data values (e.g., Secure Multi-party Computation [22]).

## 6 Conclusion and Future Work

This paper introduces a privacy-based stochastic gradient descent solution (P-SGD) that can be integrated into numerous existing privacy models in order to provide an additional layer of protection against data inference attacks during protection transitions. P-SGD features an iterative non-deterministic process that gradually decreases the data protection level during the protection descent phases. This allows preserving an appropriate precision gap between sequential protected data values to avoid potential data leakages. However, several improvements still need to be considered for this solution and addressed in future work. First, sensor data are spatio-temporal in nature, which means they also hold spatial dependencies that must be considered when measuring data dependency. In addition, the spatial and temporal distances between sequential data vary according to the user’s context. For example, distances between location data vary whether the user is driving a vehicle, running, or walking. Consequently, we aim to improve the data dependency measurement by introducing a three-dimensional dependency graph that considers temporal, spatial, and contextual dimensions. Second, we want to improve the protection function dependency procedure to further consider the semantic similarity of the features. Finally, we aim to connect P-SGD to an existing privacy model in order to test its applicability in real-life scenarios.

## References

1. Bottou, L., Bousquet, O.: The tradeoffs of large scale learning. *Adv. Neural Inf. Process. Syst.* **20**, 161–168 (2007)



2. Bou-Chaaya, K., et al.:  $\delta$ -Risk: Toward Context-aware Multi-objective Privacy Management in Connected Environments. *ACM Trans. Internet Technol.* **21**(2), 1–31 (2021)
3. Cao, J., et al.: Castle: continuously anonymizing data streams. *IEEE Trans. Dependable Secure Comput.* **8**, 337–352 (2010)
4. Chaaya, K.B., Barhamgi, M., Chbeir, R., Arnould, P., Benslimane, D.: Context-aware system for dynamic privacy risk inference: application to smart IoT environments. *Future Gener. Comput. Syst.* **101**, 1096–1111 (2019)
5. Chamikara, M., et al.: An efficient and scalable privacy preserving algorithm for big data and data streams. *Comput. Secur.* **87**, 101570 (2019)
6. Gao, S., Ma, J., Sun, C., Li, X.: Balancing trajectory privacy and data utility using a personalized anonymization model. *J. Netw. Comput. Appl.* **38**, 125–134 (2014)
7. Gheisari, M., et al.: A context-aware privacy-preserving method for IoT-based smart city using software defined networking. *Comput. Secur.* **87**, 101470 (2019)
8. Han, S., et al.: Privacy-preserving gradient-descent methods. *IEEE Trans. Knowl. Data Eng.* **22**, 884–899 (2010)
9. Islam, M.Z., Brankovic, L.: Privacy preserving data mining: a noise addition framework using a novel clustering technique. *Knowl.-Based Syst.* **24**, 1214–1223 (2011)
10. Komishani, E.G., Abadi, M., Deldar, F.: PPTD: preserving personalized privacy in trajectory data publishing by sensitive attribute generalization and trajectory local suppression. *Knowl.-Based Syst.* **94**, 43–59 (2016)
11. Li, M., Sun, X., Wang, H., Zhang, Y., Zhang, J.: Privacy-aware access control with trust management in web service. *World Wide Web* **14**, 407–430 (2011). <https://doi.org/10.1007/s11280-011-0114-8>
12. Machanavajjhala, A., Kifer, D., Gehrke, J., Venkatasubramanian, M.: l-diversity: privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data (TKDD)* **1**(1), 3-es (2007)
13. de Matos, E., et al.: Providing context-aware security for IoT environments through context sharing feature. In: *TrustCom/BigDataSE*, pp. 1711–1715. IEEE (2018)
14. Meng, X., et al.: Towards privacy preserving social recommendation under personalized privacy settings. *World Wide Web* **22**(6), 2853–2881 (2018). <https://doi.org/10.1007/s11280-018-0620-z>
15. Michael, J., Koschmider, A., Mannhardt, F., Baracaldo, N., Rumpe, B.: User-centered and privacy-driven process mining system design for IoT. In: Cappiello, C., Ruiz, M. (eds.) *CAiSE 2019. LNBIP*, vol. 350, pp. 194–206. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-21297-1\\_17](https://doi.org/10.1007/978-3-030-21297-1_17)
16. Pingley, A., Yu, W., Zhang, N., Fu, X., Zhao, W.: Cap: a context-aware privacy protection system for location-based services. In: *2009 29th IEEE International Conference on Distributed Computing Systems*, pp. 49–57. IEEE (2009)
17. Qiu, G., et al.: Mobile semantic-aware trajectory for personalized location privacy preservation. *IEEE IoT J.* (2020). <https://doi.org/10.1109/JIOT.2020.3016466>
18. Shin, H., Kim, S., Shin, J., Xiao, X.: Privacy enhanced matrix factorization for recommendation with local differential privacy. *IEEE Trans. Knowl. Data Eng.* **30**(9), 1770–1782 (2018)
19. Sweeney, L.: k-anonymity: a model for protecting privacy. *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.* **10**, 557–570 (2002)
20. Sylla, T., Chalouf, M.A., Krief, F., Samaké, K.: Towards a context-aware security and privacy as a service in the internet of things. In: Laurent, M., Giannetsos, T. (eds.) *WISTP 2019. LNCS*, vol. 12024, pp. 240–252. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-41702-4\\_15](https://doi.org/10.1007/978-3-030-41702-4_15)

21. Vollmer, N.: Table of contents EU General Data Protection Regulation (2018)
22. Vu, D.H., et al.: An efficient approach for secure multi-party computation without authenticated channel. *Inf. Sci.* **527**, 356–368 (2020)
23. Wang, L., Yu, Z., Guo, B., Ku, T., Yi, F.: Moving destination prediction using sparse dataset: a mobility gradient descent approach. *ACM Trans. Knowl. Discov. Data (TKDD)* **11**(3), 1–33 (2017)
24. Xiong, J., et al.: A personalized privacy protection framework for mobile crowd sensing in IoT. *IEEE Trans. Industr. Inf.* **16**, 4231–4241 (2019)