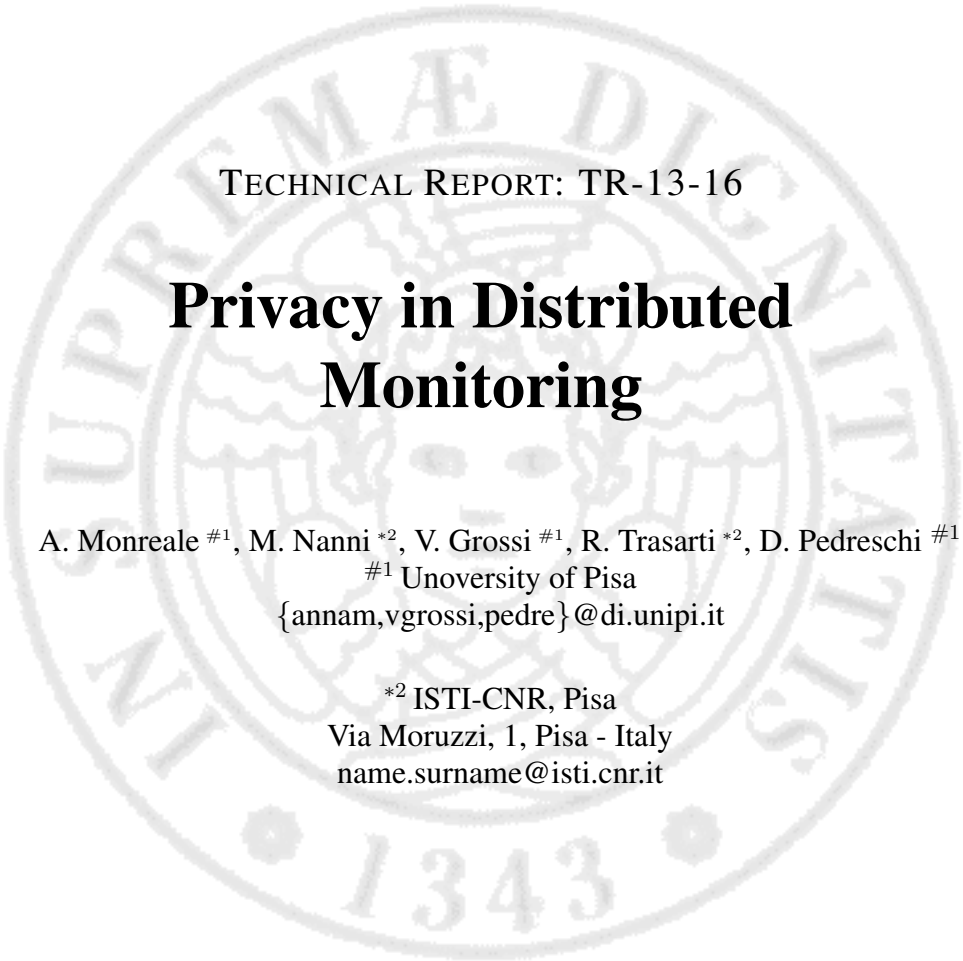
UNIVERSITÀ DI PISA
DIPARTIMENTO DI INFORMATICA



TECHNICAL REPORT: TR-13-16

Privacy in Distributed Monitoring

A. Monreale [#1], M. Nanni [*2], V. Grossi [#1], R. Trasarti [*2], D. Pedreschi [#1]
[#1] Unoversity of Pisa
{annam,vgrossi,pedre}@di.unipi.it

[*2] ISTI-CNR, Pisa
Via Moruzzi, 1, Pisa - Italy
name.surname@isti.cnr.it

31 July 2013

ADDRESS: Largo B. Pontecorvo 3, 56127 Pisa, Italy. TEL: +39 050 2212700 FAX: +39 050 2212726

Privacy in Distributed Monitoring

A. Monreale [#1], M. Nanni [*2], V. Grossi [#1], R. Trasarti [*2], D. Pedreschi [#1]
[#1] Unoversity of Pisa
{annam,vgrossi,pedre}@di.unipi.it

[*2] ISTI-CNR, Pisa
Via Moruzzi, 1, Pisa - Italy
name.surname@isti.cnr.it

31 July 2013

Abstract

In many emerging applications, such as real-time traffic monitoring, financial analysis, sensor network monitoring an important task is the continuos monitoring of stream data. In these contexts where large amount of data arrive continually the data processing requires to access often valuable personal information. As a consequence, the entire monitoring process could put at risk the privacy of people represented in the stream data. In this paper, we study the privacy issues in distributed systems during the monitoring of thresholds functions, where several nodes contribute with their data to the monitoring of a specific event. We provide a privacy-preserving framework suitable to find an acceptable trade-off among privacy protection, data quality and system performance. Using real-life data from GPS devices of private cars, we demonstrate the effectiveness of our approach in a case study consisting of the monitoring of customers mobility behaviors; in other words, we show how techniques for efficient communication can be used while preserving the individual privacy of the actors who are participating to the collective analysis.

1 Introduction

In the last years, many new emerging applications require sophisticated, real-time processing and monitoring of high-volume data streams. These stream-based applications include real-time financial analysis, network and infrastructure monitoring, fraud detection, mobility traffic monitoring and command and control in military environments. All these applications require an important task: the *continuos monitoring of stream data*. In the last years, many study in the literature have been addressed the problem of monitoring queries in distributed systems [25, 26, 7, 20, 30], where massive amount of data arrive continually and the data processing requires to access often valuable personal information. As a consequence the entire monitoring process could put at risk

the privacy of people represented in the stream data. Especially, in cases in which the data streams describe some individual activity, revealing some behavior and habit.

In this paper, we study the privacy issues in distributed systems during the monitoring of thresholds functions. We refer to the monitoring model presented in [30] and we provide a privacy-preserving approach suitable for this kind of systems. In particular, we assume a framework where some there are geographically dispersed sites called *nodes* and a central station called *coordinator*. The communication is only allowed between every site and the coordinator.

Assuring privacy protection in this kind of systems is challenging. Many privacy models proposed in literature are inapplicable due to the absence of communications among the nodes and because the communication is always *point-to-point* with the coordinator that is supposed to be untrusted. Moreover, addressing privacy issues also means finding an acceptable trade-off between privacy protection and data quality; in this specific context the goal is harder because we need to consider an additional requirement the *system performance*; in other words, in distributed monitoring systems it is important to preserve efficiency as well as privacy.

In this paper we propose a solution based on the well-know *additive randomization* [2, 6] that is suitable to guarantee privacy *at collection time* without requiring any trusted entity for the data collection. We exploit some results in the literature [18, 19] to bound the possible reconstruction of the perturbed data by an adversary. We test the proposed privacy-preserving framework in a real-world application for the monitoring of customers mobility behaviors in the context of car insurances. In our experiments on real world data coming from GPS devices of private cars, we show that our privacy-preserving framework provides acceptable results in terms of amounts of communications, privacy protection and quality of the global function to be monitored.

The proposed solution is perfectly compatible with the change of perspective towards a user-centric model for personal data management highlighted by the reform of the data protection rules proposed on January 25, 2012 by the EC and the last report of the World Economic Forum [14]. One possible way to achieve a user-centric paradigm is to enable individuals to have the full control for the user on the lifecycle of his personal data (e.g., collection, storage, processing, sharing). Thus, the user has to have an active role into a righteous and fruitful ecosystem based on personal data. Moreover, the user has to have the right to dispose or distribute his data for receiving the desired service with the desired privacy level that reflects the his privacy expectation. This is exactly the basic idea behind our framework.

The remaining of the paper is organized as follows. Section 2 introduces some background information. Section 3 provides the details of the distributed monitoring of threshold functions we refer to. Section 4 describes the privacy model and states the problem that we want to address. In Section 5 we present the details of or privacy-preserving solution. Section 6 discusses the correctness of the privacy-preserving monitoring process. Section 8 describes the details of the monitoring of the clustering quality where we want to apply our privacy-preserving method. In Section 9 we introduce the application scenario where we test our method and we show our empirical results. In Section 10 we discuss some work proposed in the literature. Finally, Section 11 concludes the report.

2 Preliminaries

In this section we introduce some notions that are important for better understanding the proposed privacy-preserving scheme.

2.1 Additive Randomization

Randomization methods are used to modify data with the aim of preserving the privacy of sensitive information. They were traditionally used for statistical disclosure control [1] and later have been extended to the privacy-preserving data mining problem [5]. Randomization based approaches use a noise quantity in order to perturb data. The algorithms belonging to this group of techniques first of all modify the data by using randomization techniques. Then, from the perturbed data it is still possible to extract patterns and models. The most famous random perturbation technique is called *additive random perturbation* This method can be described as follows. Denote by $U = \{u_1 \dots u_m\}$ with m records and n attributes, the original dataset. The new distorted dataset, denoted by $\tilde{U} = \{\tilde{u_1} \dots \tilde{u_m}\}$, is obtained drawing independently from a certain probability distribution a noise quantity z_i and adding it to each record $u_i \in U$. The set of noise components is denoted by $Z = \{z_1, \dots, z_m\}$. Most commonly used distributions are the uniform distribution over an interval $[-\alpha, \alpha]$ and Gaussian distribution with mean $\mu = 0$ and standard deviation σ. The original record values cannot be easily guessed from the distorted data as the variance of the noise is assumed enough large. Instead, the distribution of the dataset can be easily recovered. Indeed, if U is the random variable representing the data distribution for the original dataset, Z is the random variable denoting the noise distribution, and $\tilde{U}$ is the random variable describing the perturbed dataset, we have:

$$\tilde{U} = U + Z$$
$$U = \tilde{U} - Z.$$

Notice that, both m instantiations of the probability distribution $\tilde{U}$ and the distribution Z are known. In particular, the distribution Z is known publicly. Therefore, by using one of the methods discussed in [5, 3], we can compute a good approximation of the distribution $\tilde{U}$, by using a large enough number of values of m. Then, by subtracting Z from the approximated distribution of $\tilde{U}$, we can compute an approximation of U. At the end of this process individual records are not available, while obtain a distribution only along individual dimensions describing the behavior of the original dataset U.

2.2 Spectral Filtering Attack

Kargupta et al. in [24] addressed the problem to extract the real data U from the perturbed data $\tilde{U}$ by knowing only the noise distribution applied to perturb the data. The authors in this paper present an attack capable to separate the data from the noise by using a spectral filtering technique. This attack is based on the observation that the distribution of eigenvalues of random matrices presents some well-known characteristics that can be exploited for the data reconstruction.

In the following we briefly describe the spectral filtering approach. Consider a noise matrix Z with same dimensions as U. The random value perturbation techniques generate a perturbed data matrix $\tilde{U} = U + Z$. The objective of the spectral filtering based approach is to derive the estimation $\hat{U}$ of U from the perturbed data $\tilde{U}$ based on random matrix theory. An explicit filtering procedure is shown below:

Step 1: Calculate the covariance matrix of $\tilde{U}$ by $\tilde{A} = \tilde{U}^T\tilde{U}$.

Step 2: Since the covariance matrix is symmetric and positive semi-definite, we apply spectral decomposition on $\tilde{A}$ to get $\tilde{A} = \tilde{Q}\Lambda\tilde{Q}^T$, where $\tilde{Q}$ is orthogonal matrix whose column vectors are eigenvectors of $\tilde{A}$, and Λ is the diagonal matrix with the corresponding eigenvalues on its diagonal.

Step 3: Derive information of the eigenvalues from the covariance matrix of the noise Z.

Step 4: Extract the first k components of $\tilde{A}$ as the principal components by comparing $\tilde{\lambda_i}$ with eigenvalues of the noise. $\tilde{\lambda_1} \geq \tilde{\lambda_2} \geq \ldots \geq \tilde{\lambda_k}$ are the first k largest eigenvalue of $\tilde{A}$ and $\tilde{e_1}, \tilde{e_2}, \ldots, \tilde{e_k}$ are the corresponding eigenvectors. These eigenvectors form an orthonormal basis of a subspace $\tilde{\mathcal{X}}$. Let $\tilde{X} = [\tilde{e_1}, \tilde{e_2}, \ldots, \tilde{e_k}]$. The orthogonal projection onto $\tilde{\mathcal{X}}$ is $P_{\tilde{\mathcal{X}}} = X\tilde{X}^T$.

Step 5: Obtain the estimated data set using $\hat{U} = \tilde{U}P_{\tilde{\mathcal{X}}}$.

Based on the random matrix theory, we can derive the theoretical bounds of the eigenvalues corresponding to the noise matrix Z as $\lambda_{Z_{min}} = \sigma^2(1 - 1/\sqrt{Q})^2$ and $\lambda_{Z_{max}} = \sigma^2(1+1/\sqrt{Q})^2$, where Q is linear to the ratio between the number of records and the number of attributes. As in most data mining applications, the number of records far exceeds that of attributes (hence Q is large), we can see $\lambda_{Z_{min}} \approx \lambda_{Z_{max}} \approx \sigma^2 = \lambda_Z/(m-1)$.

2.3 Error Lower Bound for spectral filtering based reconstruction methods

Guo et al. in [18, 19] theoretically explore the problem which originates from the usage of additive noise for privacy preservation and give a bound for the reconstruction error and the perturbations in terms of matrix norm. This bound can help data owners to decide how much noise should be added to satisfy a given threshold of tolerated privacy breach. In other words, they provide an approach for generating the noise matrix Z with the suitable σ^2 value in such a way that a significant differences between $\hat{U}$ and U is introduced and so, a desirable privacy level is guaranteed.

The approach for discovering the lower bound is based on the notion of relative error that is defines as

$$re(\hat{U}, U) = \|\hat{U} - U\|_F / \|U\|_F \tag{1}$$

where $\|.\|_F$ denotes the Frobenius norm.

In [18, 19] authors derive a lower bound for the Singular Value Decomposition (SVD) based reconstruction and show that this bound can be considered valid also for the spectral filtering method. The SVD based reconstruction method estimates U as

$$\hat{U} = \tilde{U}_k = \tilde{L}_k \tilde{D}_k \tilde{R}_k = \sum_{i=1}^{k} \tilde{d}_i \tilde{l}_i \tilde{r_i}^T$$

where $\tilde{D}_k$ is the diagonal matrix the diagonal matrix with k principal singular values of $\tilde{U}$ and $\tilde{L}_k$ and $\tilde{R}_k$ contain the corresponding left and right singular vectors. Based on this reconstruction of the original data U the reconstruction error between $\hat{U}$ and U is

$$\|\hat{U} - U\|_F \geq \|U_k - U\|_F.$$

In order to preserve privacy, data owners need to make sure that the relative error is greater than the privacy threshold τ specified before the perturbation. Therefore, we need to have

$$\tau \|U\|_F \leq \|U_k - U\|_F = d_{k+1}^2 + \ldots + d_k^2 \tag{2}$$

Here, k which might be chosen by attackers can be determined by

$$k = max\{i | \tau \leq (d_{i+1}^2 + \ + d_n^2)/\|U\|_F\} \tag{3}$$

For i.i.d. noise based on equation (3) we have that $d_i \geq d_Z$, and the data owner should generate Z such that the eigenvalue of $(Z^T Z)$ satisfies $d_k < d_Z \leq d_{k+1}$. Since d_Z is the eigenvalue of $Z^T Z$, the variance of the noise can be derived $\sigma^2 = d_Z/(m-1)$ where m is the number of rows in Z.

3 Distributed monitoring of threshold functions

In this paper we consider the monitoring problem described in [30] where the specific scope is to solve the problem of monitoring the value of a function computed over data that are distributed in a network. We assume a framework with m geographically dispersed sites (*nodes*) and a central coordinator (*coordinator*) that can communicate with every site, while pairwise site communication is only allowed via the coordinating source. Each site receives a stream of data updates and maintains a s-dimensional local statistics vector $v_i(t)$. The coordinator has to assure that at each time instant t the following condition holds:

$$f(v(t)) \leq T \tag{4}$$

where f is a given function, $f : \mathcal{R}^s \rightarrow \mathcal{R}$, $T \in \mathcal{R}$ is a threshold and $v(t)$ is the weighted average of the $v_i(t)$ of all sites, i.e. $v(t) = (\sum_i w_i v_i(t))/\sum w_i$ for some weights $w_i \geq 0$. While the latter condition is apparently a strong limitation to the applicability of the framework, it has been shown that several interesting problems can be reformulated in this way. A way to do this, which will also be used later in this paper, is a *vector augmentation trick*, consisting in adding to vectors $v_i(t)$ (sent by the sites to the coordinator) one or more extra components. As an example consider the

monitoring of the variance of all $v_i(t)$, assuming $s = 1$, i.e. ensure that $var_i(v_i(t)) \leq T$. Clearly, it does not directly fit the form in (4), but we can exploit the well known property $var(X) = avg(X^2) - [avg(X)]^2$ to rewrite our problem as $f(v(t)) = v(t)_1 - [v(t)_2]^2$. This only requires that each site has to communicates a 2-dimensional vector $\big(v_i(t), v_i(t)^2\big)$.

In [30] authors propose the so called *geometric approach* to perform the monitoring of (4). In the following, we provide some details on that method. All the points in $\mathcal{R}^s$ where (4) is satisfied form the *admissible region* G, and our objective is simply to ensure that $v(t) \in G$. The method is based on the following property: the convex hull of a set $\{x_i\}_i \subset \mathcal{R}^s$ of points is entirely contained in $\bigcup_i B(x_i, e)$, where e is any point in $\mathcal{R}^s$ and $B(x_i, e)$ is the ball having the segment $\overline{x_i e}$ as diameter. It is straightforward to see that our $v(t)$ is contained in the convex hull of the set $\{v_i(t)\}_i$. Therefore, if every ball $B(v_i(t), e)$ is contained in the admissible region (namely, it is *monochromatic*), then also $v(t)$ will be, and therefore (4) will be satisfied. Once the coordinator has communicated to all sites the point e, each site will be able to test whether its ball $B(v_i(t), e)$ is monochromatic. As long as no site detects a failure, we are guaranteed to satisfy (4), without any need of communicating information to the coordinator. When a site fails, it notifies the coordinator, who will ask to every site to send their new vector values, and test condition (4). Notice that the test performed on each site might cause false alarms (its ball intersects the inadmissible region, while the overall $v(t)$ is completely inside the admissible one) but not false negatives; in other words, when condition (4) is violated the system will always capture that. In principle the point e can be chosen freely, however it is convenient to compute it as $e = v(t')$, where t' is the time of the most recent *synchronization*, i.e. the phase where every site communicates its new values to the coordinator. Note that Synchronizations usually occur during the set-up phase but also any time there is a site rising an alarm.

3.1 Safe Zones for convex inadmissible regions

The geometric method discussed above provides a means to decide locally to the node which vector values guarantee that the overall function satisfies the global constraint to monitor. The set of such values is also called *safe zone* and, since it is only based on the global function and on the reference point e, all nodes have the same safe zone.

In [26] it is shown that the safe zones built by the geometric method can also be computed as the intersection of an infinite set of hyperplanes. Yet, it is also shown that part of them are unnecessary, which makes the safe zones smaller (and thus less effective) than what strictly needed. A particular case is that where the inadmissible region (the set of values that violate the global constraint) is convex. In this situation we can easily find an optimal safe zone in two steps: first, find the point p of the inadmissible region which is closest to the reference point e; second, draw the hyperplane that passes through p and is orthogonal to the segment $\overline{ep}$, and then, of the two half-spaces determined by the hyperplane take as safe zone the one that contains e.

4 Privacy Model

In Section 3 we described the computational model that we refer to and we explained that each node observes local update streams and verifies that the local constraint on its stream has not been violated. In case of violation the node has to communicate its value to the coordinator. In this case we can have serious privacy issues especially when each node observes information about a single individual, thus the transmitted vector may contains sensitive information. As an example, in a scenario where the coordinator is responsible for monitoring functions on mobility data the local vector could describe the mobility behavior of a person. An attacker accessing the user vector could learn information such as the typical speed or typical trips.

Moreover, the non-communication from a specific node can reveal sensitive information about the state of the node. Finally, when the node has to communicate to the coordinator means that it is violating a local constraint, and this information itself could be sensitive. *How can we protect this sensitive information?* We think that a suitable method that we can apply in this setting is the *additive randomization* for perturbing the data to be sent. Clearly, the data randomization affects also the safe zone inserting also there an uncertainty.

In our setting, we assume that each node in our system is secure; in other words, we do not consider attacks at the node level. We also assume that the coordinator is untrusted. Therefore, we focus on designing privacy-preserving techniques to defend against an untrusted coordinator. In particular, our goal in this paper is to inscribe privacy protection in the monitoring system (Section 3) enabling the distributed monitoring of global functions while preserving the privacy of each node.

In the following we formally define the problem that we want to address.

Definition 1 *Let $\{n_1, n_2, \ldots, n_m\}$ be the m nodes of the system. We want to find a privacy-preserving technique such that the following requirements are satisfied:*

- *individual privacy is guaranteed;*
- *the system performance, in terms of number of communications, is reasonable;*
- *the correctness and the quality of the global function f to be monitored is not compromised.*

In order to address this problem we propose a method based on the additive randomization of each local vector before sending it to the coordinator.

5 Preserving Privacy in Distributed Monitoring

In this section, we present the algorithm for privacy-preserving distributed monitoring. The basic idea of our approach is to add to the original vector a noise vector where the components are drawn from a Gaussian distribution with mean 0 and standard deviation σ. Then, during the whole process for the geometric-based monitoring, described in Section 3, in the system has to be considered the noisy version of each vector. In

particular, each node uses the noisy version of the local statistics vector for checking the local constraint and if there is a violation the node transmits it to the coordinator. The coordinator averages all these noisy vectors, and checks whether the function of the global average has crossed the threshold T.

5.1 Setup Phase.

Our proposal considers an initial phase where each node adds to its initial local statistics vector $v_i(0)$ a noise vector $z_i(0)$ obtaining $\tilde{v_i}(0)$ and sends it to the coordinator, that checks that the global vector computed by using the noisy vectors $\tilde{v_i}(t)$ is within the admissible region; otherwise a global violation is raised. The coordinator defines the initial vector e and communicates it to all sites. At this point each site is able to construct its ball $B(\tilde{v_i}(t), e)$ with radius $\tilde{r_i} = \frac{\|\tilde{v_i}(t)-e\|}{2}$ and center is $\tilde{c_i} = \frac{\tilde{v_i}(t)+e}{2}$. It is immediate to understand that the addition of the noise vector affects the radius and the center of the ball and as a consequence the construction of the safe zone; in other words also the safe zone is randomized.

5.2 Local Monitoring Phase.

After constructing its ball a node monitors the local statistics vector against that safe zone; in other words, for each time t the node n_i adds a noise vector z_i to the current statistics vector $v_i(t)$ and tests its local constraints, i.e., checks if the perturbed vector $\tilde{v_i}(t)$ is contained in the admissible region (i.e., if the ball $B(\tilde{v_i}(t), e)$ is *monochromatic*). If no violation occurs, the monitoring cycles simply goes on without any communication and any action from the controller's side. If, instead, there is some local violation, the controller has to check whether there is a global violation. In particular, to verify whether the global threshold T was crossed the coordinator requires a *synchronization*, i.e., all the nodes have to transmit their perturbed statistics vectors and then evaluates whether the average of this vector is within the admissible region. In the case that a global breach is detected the coordinator computes a new estimate vector e according to the updated statistics vectors sent by the nodes.

6 Correctness of the Monitoring

As already stated above, the randomization of each local statistics vector $\tilde{v_i}(t)$ implies the randomization of each ball $B(\tilde{v_i}(t), e)$. In particular, when we add a noise vector z_i (with components drawn by a Gaussian distribution with mean 0 and standard deviation σ) to $v_i(t)$ the diameter of the original ball could increase or decrease and the ball could be also change it position. All these changes can lead to a fake or missing alarms. The first case is due to the fact the a non-monochromatic ball after the randomization could become monochromatic and generate fake violations. In other words, we could increase the false positive alarms and so, the number of communications with respect to the communications required by the monitoring without any privacy protection. The second case represents the opposite situation: a monochromatic ball becomes

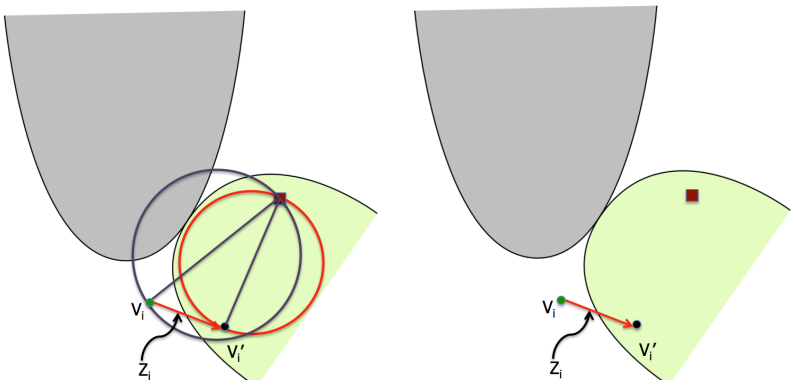


Figure 1: Missing Alarms caused by the randomization

non-monochromatic with the randomization. This means that the node might not communicate when really happens a violation of the original constraint. In other words, the correctness of the system could be compromised because of missing alarms. This case is represented in Figure 1 where grey area represents the inadmissible zone, the red ball represents the randomized ball while the other ball is the original one. We can observe that the construction of the red ball given the perturbed vector leads to a missing alarm. The same figure in the right side shows what happens in the system in terms of safe zones. We can note that the original vector lies out of the safe zone while the adding of noise moves the vector within the safe zone generating the missing alarm.

In the following we give the correctness guarantees of the privacy-preserving monitoring. In particular, we provide a probabilistic guarantee about *missing alarms*.

Given a vector $\tilde{v}_i(t)$, we know that it is the result of adding noise to each original component drawn by a Gaussian distribution with mean 0 and standard deviation σ. Fixed a probability $1-\delta$, we want to find the minimum radius such that the original vector $v_i(t)$ is one of the points in the area covered by the sphere (in s dimensions) with center $\tilde{v}_i(t)$ and a specific radius r_l; in other words, $||z_i|| = ||v_i(t) - \tilde{v}_i(t)|| \leq r_l$ with probability at least $1-\delta$. In order to do that we can observe that $||z_i||^2$ follows a χ^2_s distribution, and in particular the distribution is $\sigma^2\chi^2_s$.

Given the ball $B(\tilde{v}_i(t), e)$ of the node n_i with center $\tilde{c}_i$, we denote by $dist(\tilde{c}_i, b)$ the distance between $\tilde{c}_i$ and the boundary of the non-admissible region. We are ready to formulate the theorem that states the correctness of the monitoring.

Theorem 1 Given a perturbed local statistics vector if its ball $B(\tilde{v}_i(t), e)$ is monochromatic and $dist(\tilde{c}_i, \tilde{v}_i(t)) + r_l < dist(\tilde{c}_i, b)$ then the probability to have a missing alarm is at most δ.

Proof: As explained above with probability at least $1-\delta$ we have $||v_i(t) - \tilde{v}_i(t)|| \leq r_l$. So, we observe that $dist(\tilde{c}_i, \tilde{v}_i(t)) + r$ represents the radius of the original ball $B(v(t), e)$ with probability at least $1-\delta$. In fact, we have that $dist(\tilde{c}_i, \tilde{v}_i(t)) = \frac{||\tilde{v}_i(t)-e||}{2}$, i.e., it is the radius of the ball $B(\tilde{v}_i(t), e)$ while $\frac{||\tilde{v}_i(t)-e||}{2} + r_l \geq \frac{||\tilde{v}_i(t)-e||}{2} +$

$||v_i(t) - \tilde{v_i}(t)|| = \frac{||v_i(t)-e||}{2}$, i.e., the original ball will have at most this radius. Since, $dist(\tilde{c_i}, \tilde{v_i}(t))+r_l < dist(\tilde{c_i}, b)$ we can infer that with probability at least $1-\delta$ the original ball $B(v(t), e)$ is monochromatic and as a consequence the probability of missing alarms (non-monochromatic) ball is at most δ.

The above theorem is related to the missing alarms at node level, i.e., missing alarms that each single node can generate with the construction of its ball after the perturbation. Another form of missing alarms are those that we call *global missing alarms*. We have a missing alarm of this kind when the coordinator receives one or more alarms from the nodes, computes the average vector $\tilde{v}(t)$ and it is within the admissible region while the original $v(t)$ would not be within that region. Before providing the theorem that states the probability of global missing in the monitoring process, we note that if each node vector is perturbed by a noise vector with components drawn by a Gaussian distribution $\mathcal{N}(0, \sigma)$, then the average vector is affected by a noise from a Gaussian distribution with standard deviation $\frac{\sigma}{\sqrt{m}}$, where m is the number of nodes in the system. By following the same reasoning as in the case of local missing alarms we have that, given the perturbed average vector $\tilde{v}(t)$, with probability at least $1 - \delta$ its original version is within the area covered by the sphere (in s dimensions) with center $\tilde{v}(t)$ and radius r_g. Therefore, we have that $||v(t) - \tilde{v_i}|| \leq r_g$ with probability at least $1 - \delta$ and the noise $||v(t) - \tilde{v_i}||^2$ follows the distribution $\frac{\sigma}{\sqrt{m}}^2 \chi^2_s$.

In the following we denote by $dist(\tilde{v}(t), b)$ the distance between the global vector $\tilde{v}(t)$ and the boundary of the non-admissible region.

Theorem 2 Given the perturbed global vector $\tilde{v}(t)$, if $r_g < dist(\tilde{v}(t), b)$ then the probability to have a missing alarm is at most δ.

Proof: The proof is straightforward and derives from the observation that with probability at least $1 - \delta$ we have $||v(t) - \tilde{v_i}(t)|| \leq r_g$.

7 Protection against Spectral Filtering Attack

In Section 2.2 we discuss the weakness of the additive randomization. In our setting we assume that an attacker can access the coordinator site, obtain the matrix $\tilde{U}$ where each row is a perturbed node vector $\tilde{v}(t)$. From $\tilde{U}$ the attacker applying the spectral filtering reconstruction obtains $\hat{U}$. The distance between U and $\hat{U}$ represents the privacy protection that we measure by the relative error $re(U, \hat{U})$: greater relative error means more privacy protection. The relative error increases when we increase the magnitude of the noise to be added to the original data; in other word, a Gaussian distribution with a greater standard deviation σ guarantees more privacy protection. The goal is to find the suitable σ for the noise distribution so that a minimum level of privacy is guaranteed.

In Section 2.3 we presented the result obtained in [18, 19] related to a lower bound for the relative error after a spectral filtering attack. We also discussed as this bound can be exploited for identifying the standard deviation of the noise distribution to guarantee a minimum level of privacy τ. This methodology is perfect in a centralized system

where the data owner has the original matrix U and can find the best k such that with U_k the condition (2) is satisfied and, as a consequence, can identify the best σ of the noise distribution to be used for the perturbation. In a distributed setting, we do not have a global vision of the original vectors and so of the matrix U, thus finding the best σ for the perturbation is not possible. To solve this problem we propose to learn the standard deviation observing the historical data of the nodes N. The idea is to analyze for a long time the data about the nodes in the system and by observing the typical behavior of the data we can learn the standard deviation σ suitable to have the minimum privacy level τ. The learnt values of σ will be used during the monitoring phase. The basic assumption here is that user's behaviors present some typical regularities and we want to exploit them for finding the suitable standard deviation of the noise distribution. In the following we describe the details of the procedure for the learning phase.

For each monitor iteration tp, we consider the matrix $U(tp)$ composed of all the node vectors in the historical data $v_i(tp)$. For each possible value of $k = 1, \ldots, p$ we computes the eigenvalues, namely d_k, d_{k+1} and then d_Z defined as the average of the two eigenvalues, respecting the properties that $d_k < d_Z \leq d_{k+1}$. Finally the value of $\sigma(tp) = \sqrt{d_Z/(m-1)}$ is computed; in the following we denote by $\sigma_k(tp)$ the standard deviation at the iteration tp computed with the value k. Then, we compute the corresponding relative error corresponding to the privacy level guaranteed by the computed σ value; in other words, we compute $re(U(tp), U_k(tp)) = \tau_k(tp)$.

The learnt information, composed of a set of pairs $\langle \sigma_k(tp), \tau_k(tp) \rangle$, can be used by each node during the monitoring phase after setting the global privacy level that we desired to be guaranteed in the system. In particular, given a monitoring iteration tp and the global privacy level to be guaranteed τ the node will draw the noise from the Gaussian distribution with standard deviation $\sigma_k(tp)$ corresponding to minimum the $\tau_k(tp)$ such that $\tau_k(tp) \geq \tau$. Clearly, the learnt information could be used in a different way. As an example, after the learning we could decide to always use the maximum standard deviation found in the historical data. This could bring to use in some steps too much noise that corresponds to a better privacy but also a worst impact on the correctness of the monitoring function.

8 Distributed Monitoring of Clustering Quality

In order to evaluate our privacy-preserving method we need to verify in real applications the empirical privacy guarantees, the impact of the privacy approach on the number of communications and on the correctness of the monitoring function. To this aim we decide to apply our privacy-preserving mechanism in the application presented in [28], where the goal is a distributed monitoring of clustering quality.

The measure used to evaluate the quality of a clustering is the simple and very popular *Sum of Squared Error* (SSE in short), defined as follows:

$$SSE = \sum_{i=1}^{h} \sum_{p \in C_i} ||p - c_i||_2^2 \tag{5}$$

where C_i represents the i-th cluster, and c_i is its center (average vector).

The monitoring problem in this setting deals with dynamic data consists in continuously checking whether the last clustering computed is still good enough, recomputing the clusters only in the negative case. This requirement can be easily translated in terms of SSE by asking that the dispersion of the objects within the clusters did not grow, or at least not significantly. That means computing the SSE at each time stamp t, denoted by SSE_t, and test that it stays below some threshold. We refer to this continuous testing with the term monitoring. Finally, such a threshold should take into account the dispersion obtained at the very moment the clusters were created, which we denote with SSE_0 (i.e. time counting starts from the moment the most recent clustering was performed), suggesting to adopt a relative threshold. That is summarized in the following problem definition:

Definition 2 (Cluster Monitoring Problem)
Given a clustering $C = \{C_1, \ldots, C_h\}$ having initial SSE equal to SSE_0, and given a tolerance $\alpha \in \mathcal{R}^+$, we require to ensure that at each time instant t the following holds for the SSE of the (dynamic) dataset D_t:

$$SSE_t \leq (1 + \alpha)SSE_0 \tag{6}$$

When that does not happen, a re-computation/update of cluster assignments should be performed.

In [28] a strict version of this problem is also considered with the motivation that SSE describes all the clusters together, aggregating the dispersions of each single clusters, but this does not guarantee that a good SSE implies that each single cluster is compact, since some slightly over-dispersed cluster might be balanced in the sum by some virtuous one that adds very little to the SSE.

In this paper we use the strict variant of the clustering monitoring problem, where the constraints are imposed over each single cluster:

Definition 3 (Strict Cluster Monitoring) *Given a clustering $C = \{C_1, \ldots, C_h\}$ having initial SSE equal to SSE_0, and given a tolerance $\alpha \in \mathcal{R}^+$, we require to ensure that at each time instant t the following holds:*

$$\forall_{i=1}^{h}.SSE_t^{(i)} \leq SSE_0^{(i)} + \theta^{(i)} \tag{7}$$

where $SSE_t^{(i)}$ is the contribution of cluster i to the SSE at time t, i.e. $SSE_t = \sum_{i=1}^{h} SSE_t^{(i)}$, and the $\theta^{(i)} \in \mathcal{R}^+$ are fixed thresholds such that $\sum_{i=1}^{h} SSE_0^{(i)} + \theta^{(i)} = (1 + \alpha)SSE_0$. When condition (7) is violated, a recomputation/update of cluster assignments should be performed.

The clustering monitor problem (Definitions 2 and 3) can be fitted to the geometric approach described above, by properly rewriting it as a variance monitoring. We observe that the formulation of SSE is very similar to a variance, though on s dimensions. Each cluster C_i, having centroid c_i, contributes to the SSE by the following value:

$$
\begin{aligned}
SSE^{(i)} &= \sum_{p \in C_i} ||p - c_i||_2^2 \\
&= \sum_{j=1}^{s} \sum_{p \in C_i} (p^j - avg_{q \in C_i}(q^j)) \\
&= |C_i| \sum_{j=1}^{s} var_{p \in C_i}(p^j) \\
&= |C_i| \sum_{j=1}^{s} \left[avg_{p \in C_i}((p^j)^2) - \left(avg_{p \in C_i}(p^j)\right)^2 \right]
\end{aligned}
\tag{8}
$$

where p^j represents the j-th component of the s-dimensional vector p. This means that by augmenting the vector $v_i(t)$ of each node with the additional s features $v_{i,1}(t)^2, \ldots, v_{i,s}(t)^2$, we can compute the variance for each component. Actually, we can do slightly better, by aggregating the terms in the last line:

$$
SSE^{(i)} = |C_i| \cdot \left[avg_{p \in C_i} \left(||p||_2^2 \right) - ||avg_{p \in C_i}(p)||_2^2 \right] \tag{9}
$$

which means that only one additional component is needed, corresponding to $||p||_2^2$ of the node (p represents our $v_i(t)$).

The relation (9) states that the geometric approach can be applied for the monitoring of a single cluster, provided that we have defined a threshold value for it. This represents a solution to the strict version of our monitoring problem (Definition 3). We have an implicit partition of the problem into K separate subproblems, that means having a threshold for each single $SSE^{(i)}$. In [28] propose the fallowing partition of the global SSE; in other words they provide the values for the constants $\theta^{(i)}$ in Definition 3:

$$
\forall i. \theta^{(i)} = \beta \left(\alpha SSE_0^{(i)} \right) + (1 - \beta) \left(\alpha \frac{SSE_0}{K} \right) \tag{10}
$$

where the parameter $\beta \in [0, 1]$.

For evaluating our privacy-preserving approach we make simpler the task and assume that an initial set of profiles are provided to the coordinator, and in the set-up phase it assigns each node to the profile more similar to it. This assignment generates the initial clustering. The monitoring consists in continuously checking whether this clustering is still good enough. In the negative case a new assignment is computed. In particular, we observe that in our process when a node violates its constraint, communicates the new statistics vector to the coordinator indicating its cluster C_j. The coordinator requires to each node belonging to C_j a synchronization and checks the new value of the $SSE_t^{(j)}$, In case of violation at cluster level the coordinator raises a global violation that requires the re-assignment of the statistics vectors to the initial profiles and so all nodes have to communicate their new vectors. Note that, in this version of the problem we only change the way to compute the clustering because here the initial centroids (profiles) are already provided.

8.1 Privacy in Distributed Monitoring of Clustering Quality

As explained in Section 5, each node before sending its local statistics vector adds a noise vector. In this application each node has to perturb the vector and the additional component thus it has to send the pair $\langle \tilde{v_j(t)}, \widetilde{||v_j(t)||_2^2} \rangle$. The coordinator will use this information to compute each $SSE^{(i)}$; in particular, the formula will be:

$$
SSE_t^{(i)} = |C_i| \cdot \left[avg_{v_j(t) \in C_i} \left(\widetilde{||v_j(t)||_2^2} \right) - ||avg_{\tilde{v_j}(t) \in C_i}(\tilde{v_j}(t))||_2^2 \right].
$$

We noted that the additional component, i.e.,$|||v_j(t)||_2^2$ is useful for the clustering monitoring but could be used by an attacker for reducing the uncertainty in the vector reconstruction. So, in this particular application we propose to add a semi-trusted entity that has the goal of: 1) receiving all the perturbed additional components from the nodes; 2) computing for each cluster C_i the right component of the $SSE^{(i)}$, i.e, $||avg_{\tilde{v_j}(t)\in C_i}(\tilde{v_j}(t))||_2^2$; and 3) sending this value to the coordinator that in this way can compute the $SSE^{(i)}$ without any information about the single value of each additional component. Note that the additional component alone cannot reveal any private information about the single node.

9 Application Scenario

The monitoring of clustering quality can be used in different contexts. Here, we consider the particular case of continuous monitoring of the quality of the profiles of similar drivers by using the safe zones approach. This scenario is the same used in [28] where authors identified four categories of measures for the description of the *user driving behavior* in a time window: (i) *basic*, (ii) *space-time distribution*, (iii) *context-aware*, and (iv) *behavioral*.

The first one contains measures, directly computable from the raw GPS traces, describing the *basic* features of the trajectories in the time window. These measures allow understanding the behavior of the car usage:

- **Length**: distance travelled by the user.
- **Duration**: time spent traveling by the user.
- **Count**: number of different user's trips.
- **MaxAcceleration**: maximum user's acceleration.
- **MaxDeceleration**: maximum user's deceleration.

The second category involves more complex measures that capture how the drivers use territory in space and time and in some way describe the spatial and temporal *user distribution movements*:

- **Avg_Dist_L1**: average distance of the user from his most frequent location L1.
- **Radius_g**: radius of gyration of the user (i.e. the standard deviation from the center of mass of his movements).

- **Radius_g_L1**: radius of gyration w.r.t. to the user's L1.

- **TimeL1L2**: time spent by the user in L1 or L2.

- **EntropyLocation**: entropy of the location frequencies where the user stops.

- **EntropyTime**: entropy of user's travel time frequencies.

The third category is composed of the *context-aware* features, where information about the user's movement is related to the spatial and temporal context in which he moves:

- **EntropyArc**: entropy of road segment frequencies traversed by the user.

- **Phighway**: distance travelled on highways by the user.

- **Pcity**: distance travelled inside urban areas by the user.

- **Length_arc_crowded**: distance travelled on top 20% most crowded road segments.

- **Pnight**: distance travelled during night time (i.e. between 10 p.m. and 5 a.m.) by the user.

The last category focuses on capturing some specific mobility behaviors:

- **PAccelerationDeceleration**: percentage of rapid accelerations/decelerations of the user during his movements.

- **Pover**: how much the user drives over the speed limits.

- **Profile**: how much the user follows his profiles, i.e. trips that he performs frequently.

9.1 Dataset and data preprocessing

We performed our experiments on measures extracted from both real-world data and synthetic data.

The real-world data are provided by an Italian company called *OctoTelematics* collecting data for insurance purposes. This dataset is composed by GPS observations of 11,470 [1] private cars active in Tuscany in a period of 35 days between June and

[1] The dataset is available at *kdd.isti.cnr.it/node/493*

July 2011. Due some pre-processing (i.e. aggregation and filtering) performed by the device on board the sampling rate is reduced to a observation every 3 minutes and it is not regulated by any policy of synchronization. Moreover, we divided the dataset temporally in order to create a training and a test set: the first week for the training set and the remaining 4 weeks for the test set.

We used the training set for two tasks: (1) extracting the measures presented in the previous section using a time window of 3 days with a time granularity of 15 minutes; and (2) learning the regularity of the drivers to extract the suitable standard deviation to use in the Gaussian distribution for drawing the random noise and assuring a minimum level of privacy.

Concerning the first task we computed the measures and applied a pre-processing on them for making them suitable to the specific use. First of all, due the low sampling rate in the data, some of the measures, described in the above section, cannot be extracted. These measures involve all the acceleration based measurements and thus we have excluded them from our experiments. Once all the measures are computed on the first week, we transformed them into a log scale because some variables follow a skewed distribution. We also normalized the variables through z-score to have zero average and variance equal to 1. Finally, since the idea is to build the *customer profiles* using a clustering method, we have also studied the correlation between the measures, discovering that several strong correlations held. Therefore, we selected a subset of measures to avoid strong biases in successive analyses and, as side effect, this reduced the dimensionality of the dataset. After the above pre-processing the remaining attributes are the following: *Duration*, *Radius_g_L1*, *TimeL1L2*, *EntropyArc*, *Pcity*, *Phighway*, *Pnight*, *Pover* and *Profile*.

Concerning the second task we performed the algorithm for learning suitable standard deviation values for the noise distribution (Section 7). In other words, given the data in the first week we learnt a set of pairs $\langle \sigma_k(tp), \tau_k(tp) \rangle$, can be used by each node during the monitoring phase after setting the global privacy level that we desired to be guaranteed in the system. In our experiments we set the global privacy level at $\tau = 0.25, 0.5, 0.8$.

9.2 Experimental Evaluation

Now we empirically analyze the effects of the privacy transformation on the number of communications and on the quality of clustering and global function. Moreover, we also measure the privacy guarantees. In our experiments we set the probability of missing alarm to $\delta = 0.01$, this means that we capture possible local and global missing alarms with a probability at least equal to 99.99% and we consider a number of profiles equal to 10.

9.2.1 Communication Evaluation

To evaluate the performances of the proposed privacy-preserving approach we consider the amount of communications exchanged between the nodes and the controller and between the nodes and the semi-trusted entity for the communication of the ad-

ditional component. The communications of the first type are always a vector with s dimensions while the messages of the second type are vectors of 1 dimension. In both cases the channel is a *point-to-point* link between the node and the controller/third party. Here, we do not consider the communications from the controller to the nodes; these communications can be of different sizes and they can use *broadcasting* capabilities of the networks to reach all the nodes at once. The number of communications of this kind are negligible as showed in [28] thus, we decided to not include them in the analysis.

We compare the amount of communications required by the monitoring process without any privacy guarantee and the one required in the system when we use our privacy-preserving method with different levels of privacy. In the privacy-preserving monitoring the number of communications also includes the communications between the nodes and the semi-trusted entity.

The Figure 2 shows the behavior of the communications increasing the α parameter. As expected the number of communications increases with the privacy protection: more privacy requires more communications. This is due to two reasons: 1) in the privacy-preserving approach any time the node has to transmit the vector it has also to transmit the additional component with another transmission, so we have double communications; 2) the randomization can increase the number of false negative alarms. However, we can see that with a reasonable $\alpha = 1.5$ the privacy-preserving approach adds about 30% of communications to the original ones. This is also effect of the double communications due to the third party; indeed without this additional messages we would have a number of communications very similar. We can also note that after the α value of about 2 we have that increasing the level of privacy leads to a decreasing of the communications. This is probably due to the bad effect of a too big value of α in computing Equation (6).

9.2.2 Quality of the Clustering Monitoring

We also analyzed the impact of the randomization on the monitored global SSE and on the quality of clusters. Figure 3 shows the behavior of the SSE measure by varying α and with different levels of privacy. We observe that the SSE value increases when the level of privacy in the data is greater; however, the effect of the privacy is reasonable because we have an increasing of about 7% of the original value at worst case.

For evaluating the quality of the obtained clusters we also measured the F-measure, that is the harmonic mean of precision and recall. The recall measures how the cohesion of a cluster is preserved; it is 1 if the whole original cluster is mapped into a single randomized cluster, it tends to zero if the original elements are scattered among several randomized clusters. The precision measures how the singularity of a cluster is mapped into the private version: if the private cluster contains only elements corresponding to the original cluster its value is 1, otherwise the value tends to zero if there are other elements corresponding to other clusters. As expected we have that the increasing of privacy protection reduces the quality of the clusters. This result is depicted in Figure 4 where we vary α and plot for each value the average of F-measure obtained in each monitoring iteration. Finally, we also analyzed how changes the number of

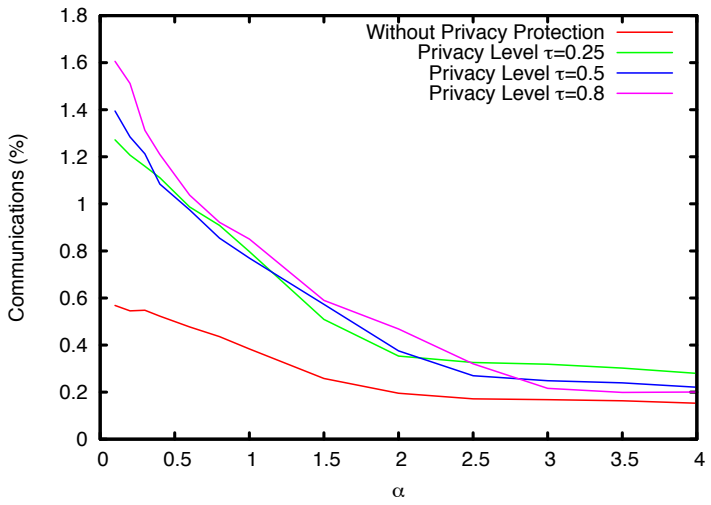


Figure 2: Communications evaluation by varying α and for different levels of privacy protection.

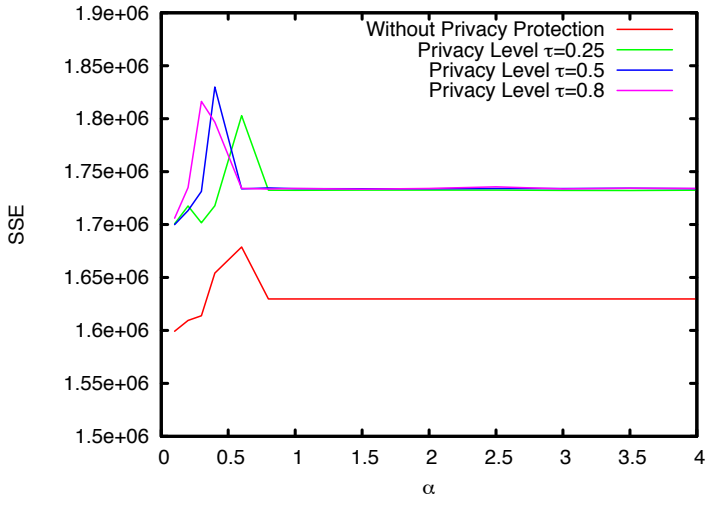


Figure 3: SSE by varying α and for different levels of privacy protection.

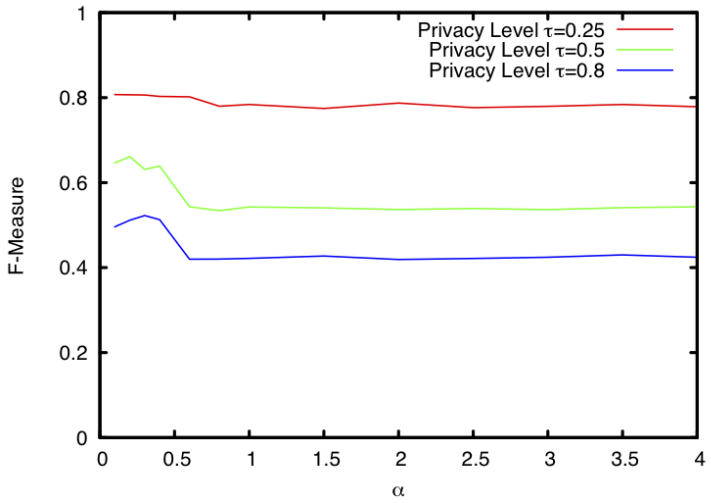


Figure 4: F-Measure behavior by varying α and for different levels of privacy protection.

re-clustering with the application of the privacy transformation. Figure 5 shows that again the effect of the privacy level is the increasing of the average of re-clustering for each value of α; however, we can note that the increment is negligible.

9.2.3 Privacy Guarantee

Finally, we also evaluated the level of privacy guaranteed at coordinator site. We simulated an attack by spectral filtering technique. Our assumption in this experiment is that the attacker has access to the list of learnt values of standard deviation that the nodes used for the noise distribution. We computed the relative error (Equation 1) any time that the nodes have sent their vectors to the coordinator. Figure 6 depicts the obtained results where we plot for each α the average of the global privacy level guaranteed during the whole process of monitoring. We can observe that the level of privacy provided is much higher w.r.t. the theoretical level of privacy set as a parameter. This is a good result considering that the performance of the system and the correctness of the global function can be considered acceptable even with this high data distortion. In particular, we can observe that setting the global privacy level at 0.25 we have a data protection corresponding to almost 0.5.

Although the methodology described in Section 7 aims at learning the suitable standard deviation for the noise distribution to have a specific global privacy level, we also analyzed the the individual privacy level. In other words, we measured the relative error at record granularity with the following formula: $re_l(\hat{X}, X) = \|\hat{X} - X\|_2 / \|X\|_2$. The three plots in Figure 7 depict the cumulative distribution of the individual privacy level for α values equal to 0.6, 1.5 and 3. We can observe that we obtain a reasonable privacy protection also at individual level; in particular all the plots show a similar

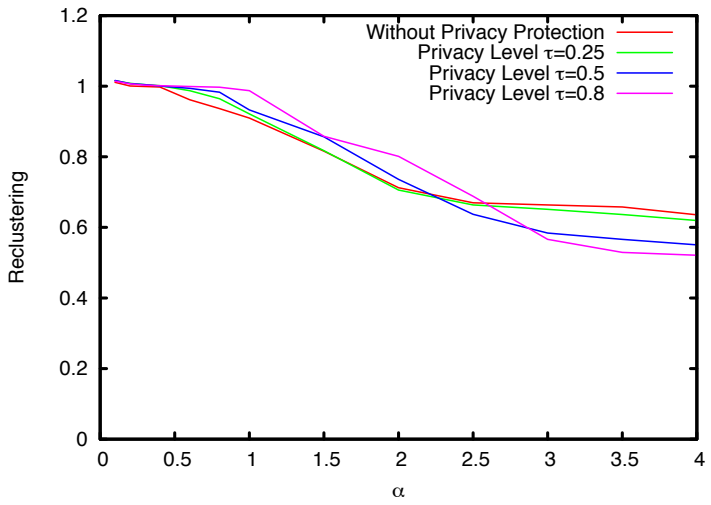


Figure 5: Number of re-clustering operation required by varying α and for different levels of privacy protection.

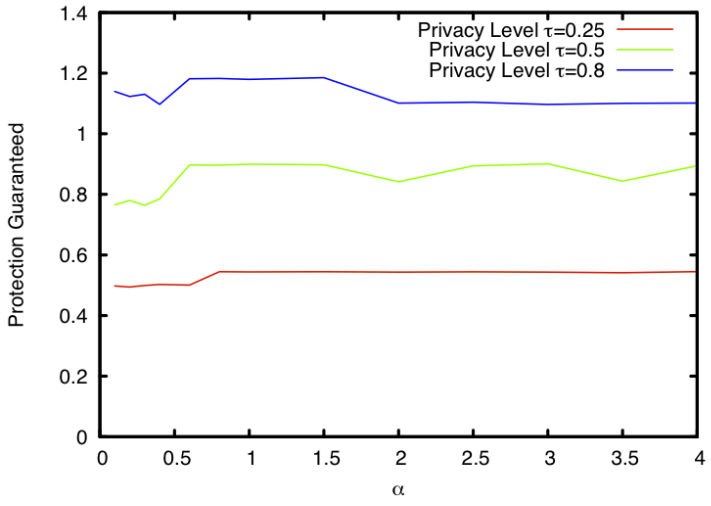


Figure 6: Global privacy level guaranteed by varying α

result. For example we have that for global privacy $\tau = 0.5$ the 80% of vectors have an individual protection of at least 0.7.

10 Related Work

Various solutions have been proposed to preserve privacy in distributed systems. Some of these solutions propose to share information using the trusted third party services [22]. But in real system sometime it is hard to have a trusted by all entities. In case where the architecture does not consider trusted third party the privacy problem is usually formulated as a variation of the secure multiparty computation (SMC) problem, which has been extensively studied in the literature [16]. However, even if in [17, 34] it has been proved that a general solution to SMC problems exists it has a high computational overhead and thus cannot be efficiently used in practice. By making a tradeoff between generality and efficiency, different solutions have been proposed to solve various information sharing issues such us intersection and equijoin [21, 4, 15], association rule mining [23, 32], classification [10, 35], top-k queries [33], and statistical analysis [9].

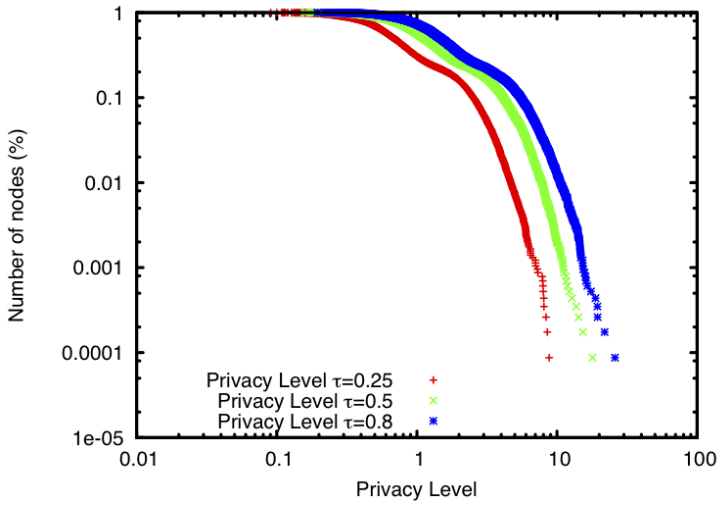
A recent model proposed in the literature is the differential privacy model [12] that provides privacy guarantees against adversaries with arbitrary background information. There are two popular mechanisms to achieve differential privacy, *Laplace* mechanism that supports queries whose outputs are numerical [12] and *exponential mechanism* that works for any queries whose output spaces are discrete [27]. Dwork et al. in [13] recently propose the notion of pan privacy, i.e., how to achieve differential privacy when the adversary is allowed access to the mechanism's internal states. The authors use the pan privacy in the continual counter mechanism [13, 11], and show how to make their counter mechanism resilient against a single unannounced intrusion. A similar problem is addressed in [8]. Rastogi et al. [29] and Chan et al. [31] consider the problem of privately aggregating sums over multiple time periods. Both of them consider untrusted coordinator, in particular, malicious coordinator, and use both encryption and differential privacy for the design of privacy-preserving data aggregation methods.

However, all these works does not consider monitoring systems of thresholding function, where the main goal is to monitor the value of a function and in the same time maintain under control the communications between the nodes and the monitor allowing the communication only when it is necessary.

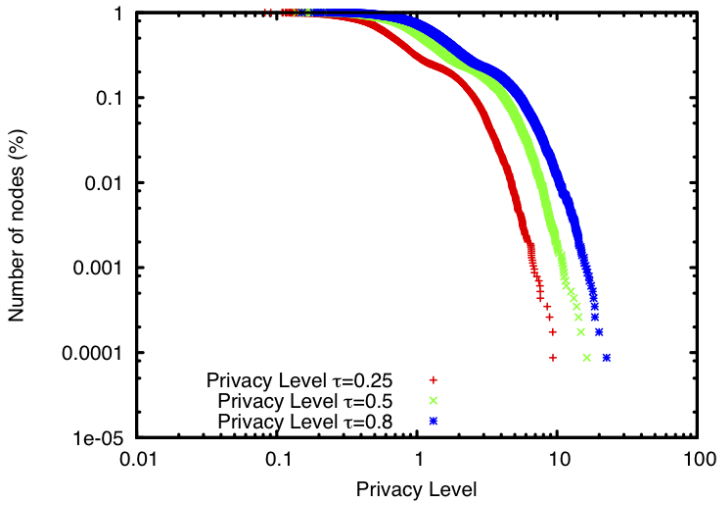
11 Conclusion

In this paper, we have proposed a method for inscribing privacy in a distributed monitoring process. Our approach is based on the well-know additive randomization and exploits some results in the literature to bound the possible reconstruction of the perturbed vectors.

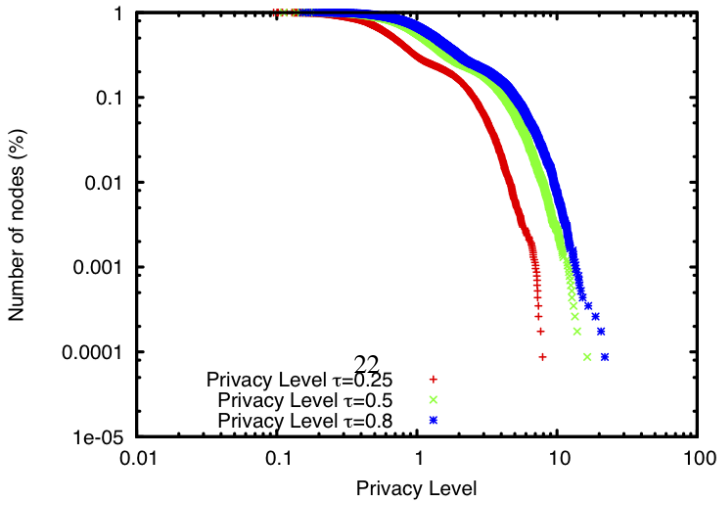
We have applied our privacy preserving technique for the monitoring of the clustering quality in a real-world application and we have evaluated the system performance in terms of number of communications, privacy guarantees and quality of the global



(a) $\alpha = 0.6$



(b) $\alpha = 1.5$



(c) $\alpha = 3$

Figure 7: Privacy protection a individual level by varying α and τ.

function to be monitored. The results show that the quality of the monitoring is reasonable while preserving good levels of privacy.

Further investigations will be directed to test our privacy preserving approach in other real-world applications such as the quality monitoring of customer segmentation with respect to their shopping habits in distributed market basket data.

References

[1] Nabil R. Adam and John C. Wortmann. Security-control methods for statistical databases: A comparative study. *ACM Computing Surveys*, 21(4):515–556, 1989.

[2] Charu C. Aggarwal and Philip S. Yu. A survey of randomization methods for privacy-preserving data mining. In Charu C. Aggarwal and Philip S. Yu, editors, *Privacy-Preserving Data Mining*, volume 34 of *Advances in Database Systems*, pages 137–156. Springer, 2008.

[3] Dakshi Agrawal and Charu C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the Twentieth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. ACM, 2001.

[4] Rakesh Agrawal, Alexandre V. Evfimievski, and Ramakrishnan Srikant. Information sharing across private databases. In Alon Y. Halevy, Zachary G. Ives, and AnHai Doan, editors, *SIGMOD Conference*, pages 86–97. ACM, 2003.

[5] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pages 439–450, 2000.

[6] Shipra Agrawal, Jayant R. Haritsa, and B. Aditya Prakash. Frapp: a framework for high-accuracy privacy-preserving mining. *Data Min. Knowl. Discov.*, 18(1):101–139, 2009.

[7] Chrisil Arackaparambil, Joshua Brody, and Amit Chakrabarti. Functional monitoring without monotonicity. In *ICALP (1)*, pages 95–106, 2009.

[8] T.-H. Hubert Chan, Elaine Shi, and Dawn Song. Private and continual release of statistics. *ACM Trans. Inf. Syst. Secur.*, 14(3):26, 2011.

[9] Wenliang Du, Yunghsiang S. Han, and Shigang Chen. Privacy-preserving multivariate statistical analysis: Linear regression and classification. In Michael W. Berry, Umeshwar Dayal, Chandrika Kamath, and David B. Skillicorn, editors, *SDM*. SIAM, 2004.

[10] Wenliang Du and Zhijun Zhan. Building decision tree classifier on private data. In *n Proceedings of the IEEE International Conference on Privacy, Security and Data Mining*, 2002.

[11] Cynthia Dwork. Differential privacy in new settings. In Moses Charikar, editor, *SODA*, pages 174–183. SIAM, 2010.

[12] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *TCC*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer, 2006.

[13] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. Differential privacy under continual observation. In Leonard J. Schulman, editor, *STOC*, pages 715–724. ACM, 2010.

[14] World Economic Forum. Report: Unlocking the value of personal data: From collection to usage. February 2013.

[15] Michael J. Freedman, Kobbi Nissim, and Benny Pinkas. Efficient private matching and set intersection. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 1–19. Springer, 2004.

[16] Oded Goldreich. *The Foundations of Cryptography - Volume 2, Basic Applications*. Cambridge University Press, 2004.

[17] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In Alfred V. Aho, editor, *STOC*, pages 218–229. ACM, 1987.

[18] Songtao Guo, Xintao Wu, and Yingjiu Li. On the lower bound of reconstruction error for spectral filtering based privacy preserving data mining. In *Proceedings of the 10th European conference on Principle and Practice of Knowledge Discovery in Databases*, PKDD'06, pages 520–527, 2006.

[19] Songtao Guo, Xintao Wu, and Yingjiu Li. Determining error bounds for spectral filtering based reconstruction methods in privacy preserving data mining. *Knowl. Inf. Syst.*, 17(2):217–240, November 2008.

[20] Ling Huang, XuanLong Nguyen, Minos N. Garofalakis, Joseph M. Hellerstein, Michael I. Jordan, Anthony D. Joseph, and Nina Taft. Communication-efficient online detection of network-wide anomalies. In *INFOCOM*, pages 134–142, 2007.

[21] Bernardo A Huberman, Matt Franklin, and Tad Hogg. Enhancing privacy and trust in electronic communities. In *Proceedings of the 1st ACM conference on Electronic commerce*, pages 78–86. ACM, 1999.

[22] Nigel Jefferies, Chris J. Mitchell, and Michael Walker. A proposed architecture for trusted third party services. In Ed Dawson and Jovan Dj. Golic, editors, *Cryptography: Policy and Algorithms*, volume 1029 of *Lecture Notes in Computer Science*, pages 98–104. Springer, 1995.

[23] Murat Kantarcioglu and Chris Clifton. Privacy-preserving distributed mining of association rules on horizontally partitioned data. *IEEE Trans. Knowl. Data Eng.*, 16(9):1026–1037, 2004.

[24] Hillol Kargupta, Souptik Datta, Qi Wang, and Krishnamoorthy Sivakumar. Random-data perturbation techniques and privacy-preserving data mining. *Knowl. Inf. Syst.*, 7(4):387–414, 2005.

[25] Ram Keralapura, Graham Cormode, and Jeyashankher Ramamirtham. Communication-efficient distributed monitoring of thresholded counts. In *SIGMOD Conference*, pages 289–300, 2006.

[26] Daniel Keren, Izchak Sharfman, Assaf Schuster, and Avishay Livne. Shape sensitive geometric monitoring. *IEEE Trans. Knowl. Data Eng.*, 24(8):1520–1535, 2012.

[27] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *FOCS*, pages 94–103. IEEE Computer Society, 2007.

[28] Mirco Nanni, Monreale Anna Trasarti, Roberto, Valerio Grossi, and Dino Pedreschi. Distributed monitoring of cluster quality for car insurance customer segmentation. *Technical Report: TR-13-11, Department of Computer Science, University of Pisa*, 2013.

[29] Vibhor Rastogi and Suman Nath. Differentially private aggregation of distributed time-series with transformation and encryption. In Ahmed K. Elmagarmid and Divyakant Agrawal, editors, *SIGMOD Conference*, pages 735–746. ACM, 2010.

[30] Izchak Sharfman, Assaf Schuster, and Daniel Keren. A geometric approach to monitoring threshold functions over distributed data streams. *ACM Trans. Database Syst.*, 32(4), 2007.

[31] Elaine Shi, T.-H. Hubert Chan, Eleanor G. Rieffel, Richard Chow, and Dawn Song. Privacy-preserving aggregation of time-series data. In *NDSS*. The Internet Society, 2011.

[32] Jaideep Vaidya and Chris Clifton. Privacy preserving association rule mining in vertically partitioned data. In *KDD*, pages 639–644. ACM, 2002.

[33] Jaideep Vaidya and Chris Clifton. Privacy-preserving top-k queries. In Karl Aberer, Michael J. Franklin, and Shojiro Nishio, editors, *ICDE*, pages 545–546. IEEE Computer Society, 2005.

[34] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *FOCS*, pages 162–167. IEEE Computer Society, 1986.

[35] Justin Zhijun Zhan, Stan Matwin, and LiWu Chang. Privacy-preserving naive bayesian classification over horizontally partitioned data. In Tsau Young Lin, Ying Xie, Anita Wasilewska, and Churn-Jung Liau, editors, *Data Mining: Foundations and Practice*, volume 118 of *Studies in Computational Intelligence*, pages 529–538. Springer, 2008.