Online Appendix to:
Driving Profiles Computation and Monitoring for Car Insurance CRM

MIRCO NANNI and ROBERTO TRASARTI, ISTI-CNR, Pisa, Italy
ANNA MONREALE, VALERIO GROSSI, and DINO PEDRESCHI, University of Pisa, Italy

This document contains additional details which are secondary to the presented article entitled *Driving Profiles Computation and Monitoring for Car Insurance CRM*. In particular, some additional experiments on the real showcase and aspects of privacy are reported here.

A. INFLUENCE OF α AND β ON SSE

In this work, we studied how the value of α influences the SSE, and how the SSE evolves over time. Figure 1 shows the results obtained in the case of the *strict clustering monitor*. In the figure, we compare different settings of our approach with a baseline where a reclustering is performed at every step of the algorithm. While the experiments in this article shows how α influences communications, if we consider Figure 1 (left), we can state that higher values of α do not influence the average SSE obtained. We can observe only small differences between the variants of our system and the baseline. Finally, considering Figure 1 (right), we can see how the introduction of the option for balancing or the use of predictive models provides a more stable behavior with respect to the baseline, forcing reclustering only when necessary. This behavior is quite similar for different values of α. A set of tests has also been performed to study the impact of β considering both communications and SSE quality. In this case, we observed that changing the value of β has no impact on communications and SSE because there is no communication reduction, and the SSE is stable as for α, thus providing a result quite similar to the one proposed in Figure 1 (right). This is due to the fact that our monitoring is not influenced by the distribution of SSE (see Equation (7)).

B. PRIVACY IN DISTRIBUTED CLUSTERING MONITORING

In the clustering monitoring model described in our article, each node observes local updated streams and verifies that the local constraint on its stream has not been violated. If there is a violation, the node has to communicate its value to the coordinator. In this case, serious privacy issues can arise. Effectively, the coordinator is responsible for monitoring functions on mobility data, and the local vector, transmitted by each node, describes the mobility behavior of a specific person. An attacker accessing the user vector could learn information such as typical speed or typical trips. Moreover, noncommunication from a specific node can reveal sensitive information about the state of that node. Finally, when the node communicates to the coordinator, it is violating a local constraint, and this information itself could be sensitive. *How can we protect this sensitive information*? A suitable method consists of additive randomization for perturbing the data to be sent. The data randomization affects also the safe zone. Our setting assumes that each node is secure, and therefore we do not consider attacks at the node level. This is motivated by the fact that GPS traces are automatically collected by safe black-boxes installed by insurance companies and made accessible exclusively to authorized personnel. This prevents potential malicious users (including the car owner) from tampering with the system for fraudulent purposes, or at least makes it extremely difficult and risky to do. On the contrary, we assume here that the

DOI: http://dx.doi.org/10.1145/2912148

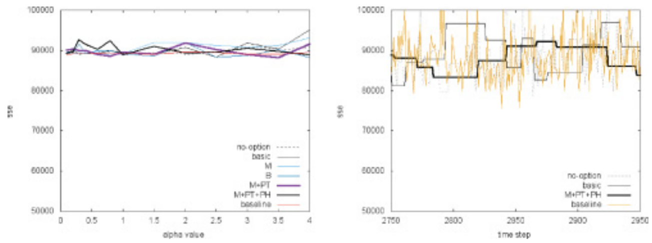

Fig. 1. SSE comparison varying α (left) and SSE behavior through time with $\alpha = 1.0$ (right).

coordinator is untrusted. Therefore, we focus on designing a privacy-preserving technique to defend against an untrusted coordinator, enabling the distributed monitoring of global functions while preserving the privacy of each node. This assumption is necessary for two reasons. First, it allows us to protect data with respect to attacks during communications and attacks at the coordinator site by external adversaries; second, the coordinator could be a third party that offers the service of monitoring to the car insurance company, and this requires protecting data from unauthorized access. We formally define the problem as:

Definition 1. Let $\{n_1, n_2, \dots, n_m\}$ be the m nodes of the system. We define 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 monitored function f is not compromised.

In this context, we propose a method based on the *additive randomization* [Agrawal and Srikant 2000] of each local vector before sending it to the coordinator.

B.1. Privacy-Preserving Technique

The 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. During the whole process, for the geometric-based monitoring, the system considers the noisy version of each vector. 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.

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, which checks if 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 builds its ball $B(\tilde{v_i}(t), e)$ with radius $\tilde{r_i} = \frac{\|\tilde{v_i}(t)-e\|}{2}$ and center $\tilde{c_i} = \frac{\tilde{v_i}(t)+e}{2}$. 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; then, even the safe zone is randomized.

Local Monitoring Phase. After constructing its ball, a node monitors the local statistics vector against that safe zone; 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; that is, it 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 goes on without any communication and no further action. If there is some local violations, the controller has to check whether there is a global violation. To verify whether the global threshold T

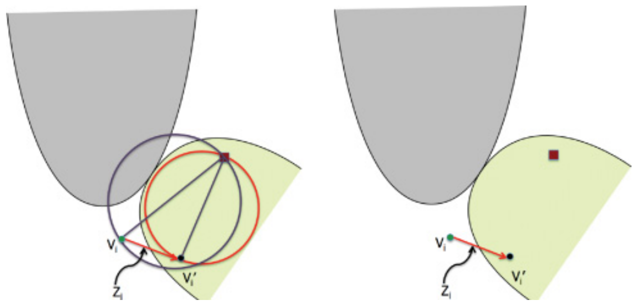

Fig. 2. Missing alarms caused by the randomization.

was crossed, the coordinator requires a *synchronization* (i.e., all nodes have to transmit their perturbed statistics vectors) and then evaluates whether the average of this vector is within the admissible region. If a global breach is detected, the coordinator computes a new estimate vector e according to the updated statistics vectors sent by the nodes.

B.2. Correctness of the Monitoring

The randomization of each local statistics vector $\tilde{v}_i(t)$ implies the randomization of each ball $B(\tilde{v}_i(t), e)$. When we add a noise vector z_i to $v_i(t)$, the diameter of the original ball could increase or decrease, and the ball could also change its position thus generating fake or missing alarms. The first case is due to the fact that a non-monochromatic ball after the randomization could become monochromatic and generate fake violations. Therefore, privacy protection might increase the number of communications because of false-positive alarms. The second case represents the opposite situation: A monochromatic ball becomes non-monochromatic with the randomization. This means that the node might not communicate when a violation of the original constraint actually happens. The correctness of the system could be compromised because of missing alarms. This case is represented in Figure 2, where the gray area shows the inadmissible zone, the red ball represents the randomized ball, while the other ball is the original one. The construction of the red ball, given the perturbed vector, leads to a missing alarm. The same figure on the right outlines what happens in the system in terms of safe zones. The original vector lies outside of the safe zone while the adding of noise moves the vector within the safe zone, thus generating the missing alarm. In the following, we give the correctness guarantees of privacy-preserving monitoring, providing 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 with 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; $||z_i|| = ||v_i(t) - \tilde{v}_i(t)|| \leq r_l$ with probability at least $1 - \delta$. We can observe that $||z_i||^2$ follows a χ_s^2 distribution, and, in particular, the distribution is $\sigma^2 \chi_s^2$.

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 nonadmissible region. Now, we 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 of having a missing alarm is at most δ.*

PROOF. As stated earlier, with probability at least $1-\delta$ we have $||v_i(t)-\tilde{v_i}(t)|| \leq r_l$. So, $dist(\tilde{c}_i, \tilde{v_i}(t)) + r$ represents the radius of the original ball $B(v(t), e)$ with probability at least $1-\delta$. 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) is at most δ. $\square$

Another form of missing alarms are those that we call *global missing alarms*: 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 alarms 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 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, 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 $\frac{\sigma}{\sqrt{m}}^2 \chi_s^2$. We denote by $dist(\tilde{v}(t), b)$ the distance between the global vector $\tilde{v}(t)$ and the boundary of the nonadmissible region.

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

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

B.3. Protection Against Spectral Filtering Attack

An attacker can access the coordinator data, obtaining the matrix $\tilde{U}$ where each row is a perturbed node vector $\tilde{v}(t)$. From $\tilde{U}$, the attacker applying the spectral filtering attack [Kargupta et al. 2005] can reconstruct an approximation of the original matrix called $\hat{U}$. The distance between U and $\hat{U}$ is the privacy protection measured by the relative error $re(U, \hat{U})$: Higher re means more privacy protection. The relative error increases with the magnitude of the noise to be added to the original data; a Gaussian distribution with a greater σ guarantees more privacy protection. So, to counter this attack, we exploit the methodology presented in Guo et al. [2008], allowing us to find a suitable σ that guarantees a minimum level of privacy. It gives a bound for the reconstruction error obtained by a spectral filtering attack, helping data owners to decide how much noise should be added to satisfy a given threshold of tolerated privacy breach. In a centralized system, the data owner identifies the best σ of the noise distribution by accessing the original matrix U. This is not possible in a distributed system because each node does not have a global vision of all the original vectors; thus, we propose to learn offline the standard deviation by observing the historical data of the nodes N. The idea is to analyze over an extended period the data pertaining to the nodes in the system; by observing the typical behavior of the data, we can learn the standard deviation σ suitable to setting the minimum privacy level τ for each monitor iteration tp. The learned values of σ will be used during the monitoring phase. The basic assumption here is that a user's behaviors present some typical regularities, and we want to exploit them to find the suitable standard deviation of the noise distribution. In the following,

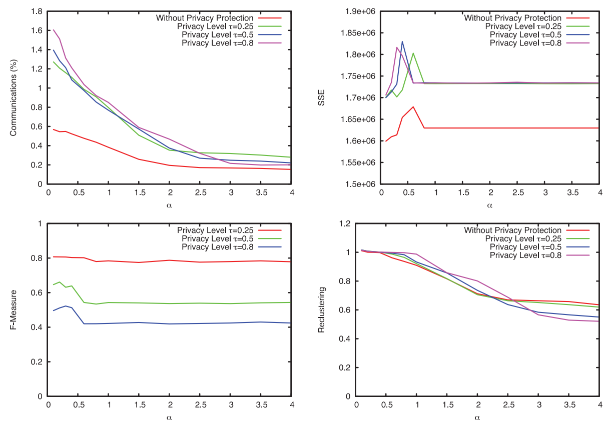

Fig. 3. Communications, SSE, F-measure, and reclustering by varying α for different levels of privacy.

we describe the details of the procedure for the learning phase, showing how to adapt this methodology to our distributed scenario.

The learned information (i.e., a set of pairs $\langle \sigma_{tp}, \tau_{tp} \rangle$) can be used by each node during the monitoring phase after setting the global privacy level that we want guaranteed in the system. 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 σ_{tp} corresponding to minimum τ_{tp} such that $\tau_{tp} \geq \tau$. Clearly, the learned information could be used in a different way. As an example, after learning, we could decide to always use the maximum standard deviation found in the historical data. This could cause us to use too much noise in some steps; this corresponds to better privacy but also to a worse impact on the correctness of the monitoring function.

C. EVALUATING THE PRIVACY PROTECTION IMPACT

Now we analyze the effects of the privacy transformation on the number of communications and on the quality of clustering and global function f. 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. To evaluate the performance 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 additional component. The communications of the first type are always a vector with d dimensions, while 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 communications from the controller to the nodes; these communications can be of different sizes, and they can use the network's *broadcasting* capabilities to reach all nodes at once. The number of communications of this kind is negligible; 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 privacy-preserving monitoring, the number of communications also includes communications between the nodes and the semi-trusted entity.

Figure 3 shows the effect of the privacy method on performance considering communications, the SSE, the F-measure, and the reclustering operations when varying the α parameter. As expected, the number of communications increases with privacy protection: More privacy requires more communications. This is due to two reasons:

(i) 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 to double communications; and (ii) 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 the effect of double communications due to the third party; indeed, without these additional messages, we would have a very similar number of communications. We note that above an α value of about 2, increasing the level of privacy leads to decreasing communications. This is probably due to the bad effect of a too-large value of α in computing Equation (2). Moreover, we analyze the impact of the randomization on the monitored global SSE and on the quality of clusters. The results show the behavior of the SSE measure by varying α and with different levels of privacy. The SSE value increases when the level of privacy is higher; however, the effect of privacy is reasonable because we have an increase of about 7% of the original value in the worst case. To evaluate the quality of the obtained clusters, we measured the F-measure, which is the harmonic mean of precision and recall.[1] As expected, by increasing privacy protection, we reduce cluster quality. This result is confirmed by the F-measures computed for the different privacy levels. Finally, the results show that the perturbations introduced by the privacy process do not have a significant impact on the number of reclusterings made by the system.

REFERENCES

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

Songtao Guo, Xintao Wu, and Yingjiu Li. 2008. Determining error bounds for spectral filtering based reconstruction methods in privacy preserving data mining. In *Knowledge and Information Systems* 17, 2 (Nov. 2008), 217–240.

Hillol Kargupta, Souptik Datta, Qi Wang, and Krishnamoorthy Sivakumar. 2005. Random-data perturbation techniques and privacy-preserving data mining. In *Knowledge and Information Systems* 7, 4 (2005), 387–414.

[1]Recall measures the cohesion of a cluster; 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. Precision shows the singularity of a cluster: If the private cluster contains only elements of the original cluster, its value is 1; otherwise, the value tends to zero if it contains elements corresponding to other clusters.