

Privacy-Preserving Calibration for Participatory Sensing

Kevin Wiesner^(✉), Florian Dorfmeister, and Claudia Linnhoff-Popien

Mobile and Distributed Systems, Ludwig-Maximilians-Universität
München (LMU Munich), Oettingenstr. 67, 80538 Munich, Germany
{kevin.wiesner,florian.dorfmeister,linnhoff}@ifi.lmu.de

Abstract. By leveraging sensors embedded in mobile devices, participatory sensing tries to create cost-effective, large-scale sensing systems. As these sensors are heterogeneous and low-cost, regular calibration is needed in order to obtain meaningful data. Due to the large scale, on-the-fly calibration utilizing stationary reference stations is preferred. As calibration can only be performed in proximity of such stations, uncalibrated measurements might be uploaded at any point in time. From the data quality perspective, it is desirable to apply backward calibration for already uploaded values as soon as the device gets calibrated. To protect the user's privacy, the server should not be able to link all user measurements. In this paper, we therefore present a privacy-preserving calibration mechanism that enables both forward and backward calibration. The latter is achieved by transferring calibration parameters to already uploaded measurements without revealing the connection between the individual measurements. We demonstrate the feasibility of our approach by means of simulation.

Keywords: Participatory sensing · Mobile sensing · On-the-fly calibration

1 Introduction

Today, mobile phones already include an increasing set of embedded sensors. Currently available phones come with built-in accelerometers and gyros, as well as location, audio, and image sensors. With this development, mobile phones evolve from standard phones, intended for personal communication only to ubiquitous sensing devices that are globally distributed. These devices can be utilized to form a new kind of sensor network, so-called participatory sensing networks (PSN) (also referred to as *mobile phone sensing* [1] or *people-centric sensing networks* [2]), where people serve as carriers for mobile phone-based sensing devices. PSNs allow for large-scale, global data collection and real-time information display. In future, they could be used, e.g., to monitor environmental pollution, temperature or the noise intensity of urban areas. The main advantage of PSNs is that data can be collected on a large-scale with automatically deployed and virtually always-on, consumer-paid and continuously recharged sensor nodes.

Leveraging the sensors built into mobile phones as information source typically entails two main problems: On the one hand, those sensors are heterogeneous, due to the great number of different manufacturers and device models. On the other hand, sensors embedded in mobile phones. Consequently, calibration is necessary in order to obtain meaningful data and poses a crucial aspect for the success of PSNs. In general, there are two types of calibration: manual and on-the-fly. The former is typically performed by field experts and is used for high precision instruments, especially if manageable amounts of sensors have to be calibrated. On-the-fly calibration describes an online process, in which sensors are automatically calibrated while being deployed and running. It is done by utilizing stationary reference stations, whose measurements are used as ground-truth. For large-scale PSNs, manual calibration is too elaborate and time-consuming, and thus on-the-fly calibration is preferred.

A calibration process can only be performed if a mobile phone user comes sufficiently close to one of those reference stations. As the mobility of users cannot be controlled, this can lead to the upload of uncalibrated measurements, especially in case of long intervals without a user's encounter with a reference station. Hence, in order to improve the system's overall quality of information, it is desirable that the server can apply backward calibration for already uploaded values, as soon as the calibration process is carried out for a client, i.e., the server adjusts previously uploaded measurement values with the newly determined calibration parameters. In order to protect the user's privacy, though, the server should not be able to link all conducted measurements of a client, as this could reveal the user's entire mobility trace. In other scenarios, this could be achieved by using changing pseudonyms in combination with MIX networks [3] to avoid the traceability of users and their measurements. But the quasi uniqueness of the calibration parameters would allow to link calibrated measurements of a user.

In this paper, we present a privacy-preserving calibration mechanism that enables both forward and backward calibration, while protecting a participating user's privacy. The latter is achieved by transferring calibration parameters to already uploaded measurements in a way that completely blurs the connection between the individual measurements.

The remainder of this paper is organized as follows. Section 2 discusses related work. In Sect. 3, we introduce the calibration model, followed by the description of our privacy-preserving calibration system in Sect. 4. Then, we evaluate our approach in Sect. 5 and finally conclude in Sect. 6.

2 Related Work

There is a lot of research work related to participatory sensing. Most work focuses on approaches and techniques that enable data collection with mobile phones [1, 2, 4], but neglect calibration issues. In addition, there is also a wide range of work dealing with sensor calibration in general. Those approaches often cannot be applied to participatory sensing, as dense networks of static and resource-constrained nodes are assumed [5].

Miluzzo et al. proposed CaliBree [6], a distributed self-calibration system for mobile wireless sensor networks. Mobile sensors compare their data with those of ground-truth nodes when they experience the same environment, i.e., upon reception of locally broadcasted ground-truth information. As their nodes do not possess any positioning capabilities, they are dependent on the broadcasted information. In our approach, we assume that mobile phones are able to determine their position (e.g., using GPS), which allows for a more precise determination of whether nodes should experience the same environment. Furthermore, no direct wireless communication link between ground-truth stations and sensors is necessary, thereby facilitating the integration of already existing measurement stations and avoiding investments in new hardware. In contrast to the distributed CaliBree calibration, Honicky [7] presented a centralized approach, where the automatic calibration of sensors embedded into mobile phones is achieved by using Gaussian process regression. Through the cloud-based approach, global information about all of the sensors in the system can be integrated into the calibration process. Hasenfratz et al. [8] introduced new calibration algorithms, i.e., backward and instant calibration for on-the-fly calibration of low-cost gas sensors. The focus of the paper lies on applying the algorithms on actual data and no mechanisms for the exchange of data between the entities is described.

These approaches either neglect the privacy aspect as a central instance knows about all measurements of the nodes [7] or do not take into account that nodes pass by reference stations infrequently. The latter leads to the upload of possibly uncalibrated measurements. To the best of our knowledge, our approach is the first that preserves the users' privacy and allows for backward and forward calibration.

3 Calibration Model

We assume mobile phones to be equipped with low-cost gas sensors, which we aim to calibrate with our system. In this section, we therefore introduce the underlying calibration model.

PSNs can be seen as a special type of sensor network. Sensor networks usually aim to monitor one or multiple phenomena of interest. In order to be able to detect a phenomenon P , there needs to be a measurable signal $p : T \rightarrow D$ that arises from P , with $T \subseteq \mathbb{R}^+$ being the time and $D \subseteq \mathbb{R}$ being the value domain. Let $m_s(t_i)$ be the measurement of a sensor s at time $t_i \in T$, and $p(t_i)$ the actual value of the phenomenon at that time. If sensor s is a *perfect sensor*, $m_s(t_i) = p(t_i)$ is true for any point in time and no calibration is necessary.

However, sensors are typically not behaving perfectly, and especially for low-cost gas sensors there is a significant precision loss due to sensor aging [9] and influencing contextual settings (e.g., humidity) [10]. Calibration of sensors can hence be described as the process of minimizing the deviation of the measured values $m_s(t_i)$ from the actual values $p(t_i)$, which is achieved by applying a *calibration curve* ϕ to the measured values. We use a polynomial of order k as a representation of $\phi : \mathbb{R}^{k+1} \times D \rightarrow D$ with a vector of *calibration parameters* $c = (c_0, c_1, \dots, c_k) \in \mathbb{R}^{k+1}$ and x as the measurement input:

$$\phi(c, x) = \sum_{n=0}^k c_n * x^n. \quad (1)$$

As a sensor can be calibrated several times, we denote $\omega : T \rightarrow \mathbb{R}^{k+1}$ as the function returning the effective calibration parameters at a certain point of time. As a result, the calibrated value $\tilde{m}_s(t_i)$ of a sensor s at time t_i is

$$\tilde{m}_s(t_i) = \phi(\omega(t_i), m_s(t_i)) = \sum_{n=0}^k \omega(t_i)_n * m_s(t_i)^n. \quad (2)$$

For a *perfect sensor* s that needs no calibration, it is $\forall t_i \in T : \omega(t_i) = (0, 1, 0, 0, \dots, 0) \in \mathbb{R}^{k+1}$ and $m_s(t_i) = p(t_i)$. By means of calibration we aim for *perfectly calibrated* sensors that behave like *perfect sensors* from a point t_c in time onwards, so that $\forall t_i \geq t_c, t \in T : \omega(t_{i+1}) = \omega(t_i)$ and $\tilde{m}_s(t_i) = p(t_i)$. This ideal state is typically not reached, as sensors continuously degrade and thus do not remain perfectly calibrated. However, by continuously repeating the calibration process an approximation of the ideal state can be reached.

In order to determine the above introduced *calibration curve* ϕ , a set C (with $|C| \geq (k + 1)$) of calibration tuples $(m_s(t_i), p(t_i))$ is needed, i.e., for a certain number of measurements we need to know the actual value of the phenomenon of interest. For this purpose, we utilize stationary reference stations, as we assume those sensors to be perfectly calibrated at any point. For each measurement $m_s(t_i)$ and actual value $p(t_i)$, we store the time t_i and the location l_i of the mobile phone, respectively of the reference station, so that we have a set of measurements M , consisting of tuples of the form $(t, m_s(t), l(t))$, and a set of actual values S , consisting of tuples of the form $(t, p(t), l(t))$. To access the different parts of these tuples, we use the dot notation, e.g., $m.l$ for the location of a tuple $m \in M$. Hence, the set of calibration tuples C can be written as:

$$s \in S, m \in M : C = \{(s.p, m.m_s) \mid |s.t - m.t| \leq \delta_t \wedge |s.l - m.l| \leq \delta_l\}. \quad (3)$$

δ_t and δ_l are parameters describing the temporal and spatial distance between ground-truth and mobile measurements, which have to be adapted according to the phenomenon of interest.

4 Privacy-Preserving Calibration System

In this section, we will describe our Privacy-Preserving Calibration System (PPCS). Overall, we assume that users conduct measurements using their mobile phones and upload their data to a server, which is responsible for storing all measurements. The upload is done via MIX networks with users utilizing self-generated pseudonyms for communicating their measurements and change those on a regular basis. Users can even use a new pseudonym for each measurement. These pseudonyms are necessary in order to be able to reference specific measurements within the backward calibration process. PPCS is an on-the-fly

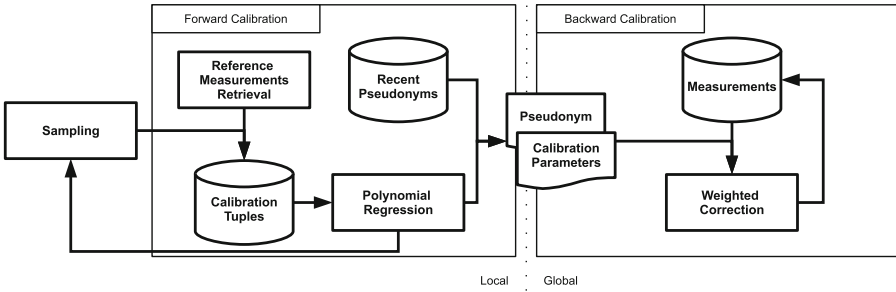


Fig. 1. Calibration pipeline showing the two calibration phases

calibration system, i.e., it calibrates sensors while they are in use by utilizing stationary reference stations providing ground-truth data. Many cities already deployed stationary sensor stations measuring the air quality in use. For instance, Zurich has four stations¹, and in Munich there are even 10 stations deployed². We assume such reference stations to be available and that their measurements are accessible through well-defined web service interfaces.

Figure 1 illustrates the calibration pipeline of our system. By comparing reference measurements to the user’s measurement data, instant *forward calibration* can be performed. Forward calibration refers to the process of determining a calibration curve on a user’s mobile device that is applied to future measurements before uploading those. In contrast, *backward calibration* refers to the process of adjusting previous measurements by applying a newly determined calibration curve to already uploaded data. In order to instruct the server to perform backward calibration for concerned measurements, users have to transfer the freshly acquired calibration parameters in combination with the previously used pseudonym to the server. In the following, the two calibration phases are described in more detail.

4.1 Forward Calibration

In the forward calibration process, a calibration curve is determined based on the comparison of recent measurements of both the mobile phone and a reference station. First, the user’s device (hereafter referred to as the *client*) needs to be aware of any reference stations within its area. Therefore, the server provides a list of reference stations together with their locations and the accessible data interface for the reference measurement retrieval. This list is requested as soon as the client enters an unknown area, and is refreshed by periodical updates. Knowing the locations of nearby reference stations, the client checks for each measurement, whether it is in proximity of one of those. If so, the reference measurements are retrieved. As mentioned in the previous section, the temporal

¹ <http://www.ostluft.ch/>

² <http://maps.muenchen.de/rgu/luftmessstationen>

and spatial ranges stating what is to be considered as “proximity” depend on the phenomena of interest and have to be specified by adapting the parameters δ_l and δ_t in Eq. 3.

The locally recorded measurements and the reference measurements are then combined and the calibration tuples are formed through a temporal and spatial filtering process (cf. Sect. 3). Basically, this step combines measurements that were taken at approximately the same time and location. These calibration tuples are then used to determine a calibration curve that is specific to the current state of a mobile user’s sensing equipment. In order to avoid distorted or premature calibrations, PPCS takes the following countermeasures: First, forward calibration is only performed if a predefined minimal number of calibration tuples ($C_{MinCount}$) exist in order to reduce the impact of possible outliers within the calibration tuples. Second, calibration is only started if a certain value range within the calibration tuples is covered ($C_{MinRange}$), to avoid a calibration optimized for a limited value range. Third, in order to avoid unnecessary calibrations, the calibration process is only started if a certain timeout has been exceeded since the last calibration ($C_{Timeout}$). The actual determination of the calibration curve parameters is done by polynomial regression. The model is fitted using the method of least squares, which minimizes the sum of the squares of the deviations between reference and mobile sensor measurements. The determined calibration tuples are then used to correct future measurements before uploading them (see Fig. 2b). In a discretized form, they are also used during backward calibration to correct already uploaded measurements.

4.2 Backward Calibration

In the backward calibration process, already uploaded measurements should be adjusted with a newly determined calibration curve. As already mentioned, users change their pseudonyms on a regular basis in order to protect their privacy. As a result, only the users themselves know which pseudonyms the calibration curve should be applied to. Thus, a client that has locally determined a new calibration curve has to inform the server about the pseudonyms and the calibration parameters. A naive approach would be to send tuples consisting of the pseudonym to be adjusted and the calibration vector c . However, this would naturally lead to a breach of the user’s privacy: as the exact calibration parameter vector typically differs from phone to phone, sending c could reveal the link between the different pseudonyms of a user (see Fig. 2a).

In PPCS, this is counteracted by incorporating the concept of *k-anonymity* [11]. To obfuscate the exact calibration parameter, the client discretizes the calibration parameters before uploading them to the server. By this, the probability of having the same calibration vector c as other clients and achieving *k-anonymity* is increased. For this process, a discretization function $\psi : \mathbb{R}^{k+1} \times \mathbb{R}^{k+1} \rightarrow \mathbb{R}^{k+1}$ is used, which returns a discretized (and thereby generalized) calibration vector \tilde{c} :

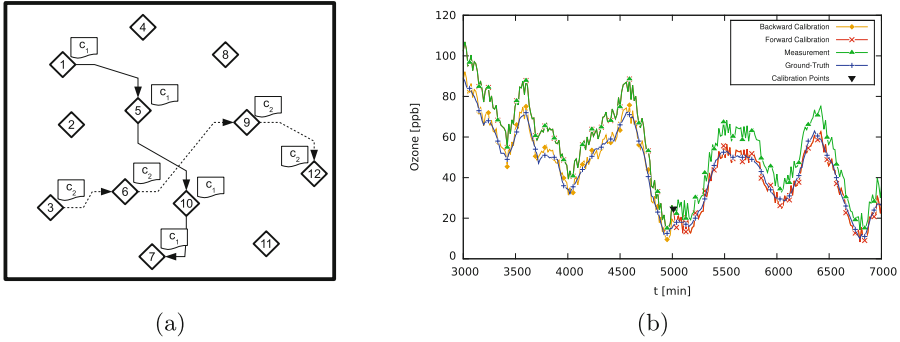


Fig. 2. (a) Applying exact backward calibration parameters (here: c_1 and c_2), can reveal the link between uploaded measurements (indicated with diamonds). (b) Example excerpt of simulated and calibrated measurements of a node over time.

$$\tilde{c} = \psi(c, d) = \begin{pmatrix} \lceil \frac{c_0}{d_0 * \theta(c)} \rceil * (d_0 * \theta(c)) \\ \vdots \\ \lceil \frac{c_k}{d_k * \theta(c)} \rceil * (d_k * \theta(c)) \end{pmatrix}, \tag{4}$$

where $d \in \mathbb{R}^{k+1}$ is the discretization vector that is known system-wide (i.e., all clients use the same d) and $\lceil x \rceil$ denotes the rounding function to the nearest integer. θ describes a factor for adjusting the discretization granularity to the extent of the deviation δ of c from the perfect sensor s : $\theta(c) = 2^{\max(\lceil \lg \delta(c) - \varphi \rceil, 0)}$, with δ being the degree of deviation $\delta(c) = \|c - s\|_2 = (\sum_{n=0}^k (\frac{c_n - s_n}{d_n})^2)^{\frac{1}{2}}$, and φ being a constant for determining the steps of adjustment. To clarify this step, we illustrate the discretization with an example: We assume a calibration vector $c = (9.3292, 0.8567)$ and a discretization vector $d = (2.0, 0.1)$ with $\varphi = 2$. This leads to $\delta(c) = 4.8798$ and $\theta(c) = 2$, and finally to the discretized calibration vector $\tilde{c} = (8.0, 0.8)$.

Naturally, as c is distorted, the discretization process leads to a loss of precision, with the amount of distortion depending on d . However, the error introduced should be relatively small compared to the gain of precision achieved by calibrating and adjusting $m_s(t_i)$ to $\tilde{m}_s(t_i)$, even with deliberately distorting the calibration parameters. Furthermore, as \tilde{c} is only used within the backward calibration process, the error does not propagate to future measurements.

To avoid privacy attacks based on the upload time, backward calibration parameters are only uploaded at certain specified times, resulting in so-called “calibration bursts”. By this, all users that want to apply backward calibration to their measurements, upload their parameters for the total interval since the last calibration burst. As done before, the upload of \tilde{c} to the server is carried out via a MIX network, so that the updates cannot be linked to the physical device.

The last step is the weighted correction of former measurements by the server. This is done by applying the received calibration parameters and calculating a new measurement value. Ideally, this new value and the former value should be

combined to a corrected measurement value by using weights that depend on the point of time within the last calibration period of the corresponding node. Measurements closer to the calibration point at which the backward calibration parameters have been determined should be stronger affected by the correction than measurements closer to the previous calibration point. The idea behind this is that it is typically not reasonable to alter measurements that have just been (forward) calibrated by applying a much later determined backward calibration. However, as the server does not know the actual calibration times of a node, only an approximation can be calculated. Instead of using the actual calibration times, the server uses weights that depend on the point of time within the calibration burst. The corrected value $\tilde{m}_s(t_i)$ is calculated with the following formula

$$\tilde{m}_s(t_i) = \frac{(t_i - cb_{n-1}) * \phi(\omega(cb_n), m_s(t_i))}{cb_n - cb_{n-1}} + \frac{(cb_n - t_i) * \phi(\omega(cb_{n-1}), m_s(t_i))}{cb_n - cb_{n-1}}, \quad (5)$$

where cb_n and cb_{n-1} denote the times of the current calibration burst and the previous calibration burst respectively. As this might heavily deviate from the ideal weighted correction, the client calculates the ideal weighted correction $\hat{m}_s(t_i)$ itself before uploading the backward calibration parameters

$$\hat{m}_s(t_i) = \frac{(t_i - ct_{n-1}) * \phi(\omega(ct_n), m_s(t_i))}{ct_n - ct_{n-1}} + \frac{(ct_n - t_i) * \phi(\omega(ct_{n-1}), m_s(t_i))}{ct_n - ct_{n-1}}, \quad (6)$$

with ct_n and ct_{n-1} denoting the actual calibration times of that node. Only if a backward calibrated value is closer to the ideally corrected value, i.e., if $|\hat{m}_s(t_i) - m_s(t_i)| > |\hat{m}_s(t_i) - \tilde{m}_s(t_i)|$, the client uploads the calibration parameters and initiates the backward calibration process.

5 Evaluation

We evaluated our concept by means of simulation. As ground truth data for our simulated measurements, we used real ozone measurements of 14 days collected at stationary stations in Munich (cf. Footnote 2). We interpolated this data in the time domain to increase the resolution from 1/hour to 1/minute, as well as in the spatial domain, in order to have a ground truth value for each position within the simulation area. For the latter, we employed Shepard's method for Inverse Distance Weighting [12] with the power parameter $p = 2$.

To simulate the deviation of mobile sensors, we used the model for ozone measurements presented in [8]: the authors deployed sensors with *MiCS-OZ-47* ozone sensing heads, and found that the measurement errors are normally distributed, if they are only initially calibrated. They observed a normal distribution $\mathcal{N}(\mu, \sigma^2)$ with $\mu \sim \mathcal{U}(-9, 9)$ ppb and $\sigma \sim \mathcal{N}(3, 1)$ ppb over the period of a day. For our simulations, we applied this model to generate artificial data, i.e., based on this model we determined an error curve for each sensor node. The error curve was set to an order of 1, i.e., a polynomial of the form $a * x + b$, where a was set to a random value ranging from $[-8.0, 8.0]$ and b to a value ranging

Table 1. Simulation setup

No. of nodes	1000, 1500, 2000	Simulation time	14 days
Mobility model	Random Walk	Max. speed	$8.33 \frac{m}{s}$
Measurement frequency	4x per hour	No. of reference stations	5
δ_l	250 m	$C_{MinCount}$	5
$C_{MinRange}$	30 ppm	$C_{Timeout}$	5 days

from $[-0.2, 0.2]$, as those values closely modeled the mentioned behavior. We also integrated an aging factor of 0.2 ppm/day (as in [8]) to account for the loss of precision over time. As a result, a measurement was simulated by applying the error curve on the ground truth value, adding the deviation arising from sensor aging, and finally adding some noise from the aforementioned distribution.

We then conducted simulations with the setup stated in Table 1. The backward calibration was performed once per week. The calibration curve ϕ was set to an order of 1, thus c, \bar{c} , and $d \in \mathbb{R}^2$. In our evaluations we used the following discretization parameters: $d_0 = \{1.0, 1.5, 2.0\}$, $d_1 = \{0.05, 0.1, 0.15, 0.2\}$, and $\varphi = \{2, 3, 4\}$, resulting in 36 different discretization combinations. In the following, discretization parameter combinations are written in the form $d_0, d_1; \varphi$.

5.1 K-Anonymity

In a first step, we analyzed our approach regarding the level of k-Anonymity. We therefore run simulations with each of the above mentioned discretization combination and analyzed how often k-Anonymity was reached for $k = \{2, 3, \dots, 10\}$.

Figure 3a–c show the achieved k-Anonymity for 1000 nodes. It is obvious that more fine-grained discretization vectors, i.e., vectors with small discretization steps (such as 1.0, 0.05; 4.0) perform worse than more coarse-grained vectors (such as 2.0, 0.2; 2.0). It can be seen that especially the discretization parameter d_1 is decisive, and that discretizations with $d_1 = 0.15$ or $d_1 = 0.2$ reached the desired k-Anonymity level significantly more often. The results also show that smaller values for φ have a more positive impact on the anonymity level than larger values, as the discretization parameters are adapted more rapidly and thus become more coarse-grained. For $k = 5$, the k-Anonymity level was reached in more than 80 % of the time with 28 out of the 36 discretization combinations. For $k = 10$, 23 discretization combinations reached the specified level in more than 60 % of the time. We then selected the worst and the best performing discretization from the former results and simulated it with varying node numbers, i.e., $\#nodes = \{1000, 1500, 2000\}$. The results are shown in Fig. 3d. It can be seen that especially in the worst case, the increase of participating nodes significantly increases the percentage of achieved k-Anonymity.

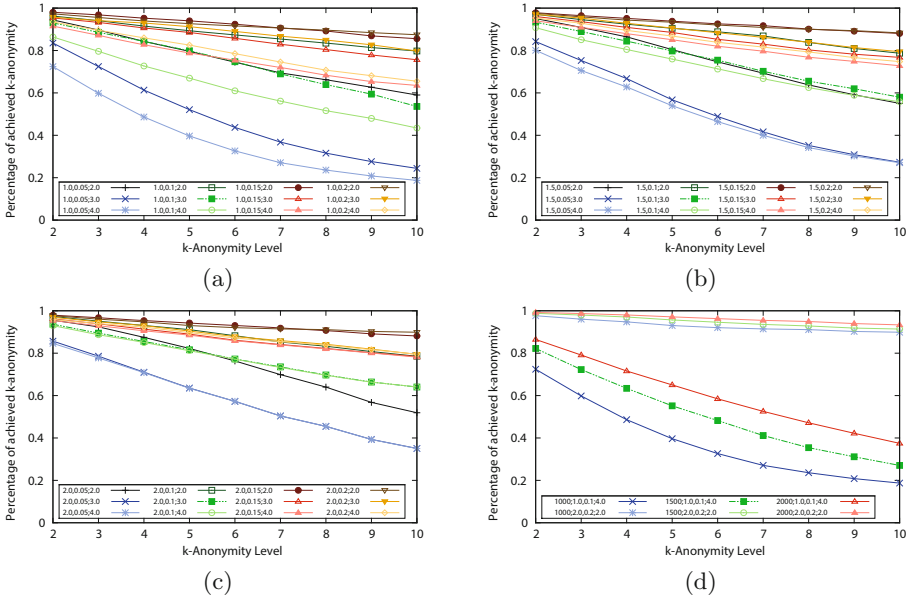


Fig. 3. Achieved average k-Anonymity level for varying discretization parameters over the simulated 14-day period.

5.2 Discretization Error

In a next step, we analyzed the error introduced by discretizing the calibration parameters in the backward calibration process. In this step, we only considered discretization parameters that achieved a k-Anonymity level of 10 at least 60% of the time. Figure 4a, b show the average discretization error in relation to the average calibration gain (the average was calculated only over the amount of nodes that performed a calibration). For the former, we compared the results using the discretized calibration vector \tilde{c} with those using the exact calibration parameters c (in relation to the ground truth value). The calibration gain is the average gain in precision when applying the discretized calibration curve \tilde{c} , compared to results without calibration. Here, the results are obviously orthogonal to the aforementioned results: the most fine-grained discretization results in the lowest error and the highest gain. It can be seen again that especially the choice of d_1 and φ are decisive for the result. Even though a few exceptions resulted in a negative backward calibration gain, i.e., the discretization of the calibration lead to a worse result than without the calibration, with most parameters a positive result was achieved.

We further examined the calibration gain for each calibration period, which is the time interval between two calibration points, e.g., the first calibration period (C_1) is the time interval from the simulation start until the first calibration. More precisely, we define the set of calibration periods as follows: $\{i \in 1, \dots, n + 1 : C_i = [t_{c_{i-1}}, t_{c_i}]\}$, with $\{t_{c_1}, t_{c_2}, \dots, t_{c_n}\}$ being the set of calibration times. As we

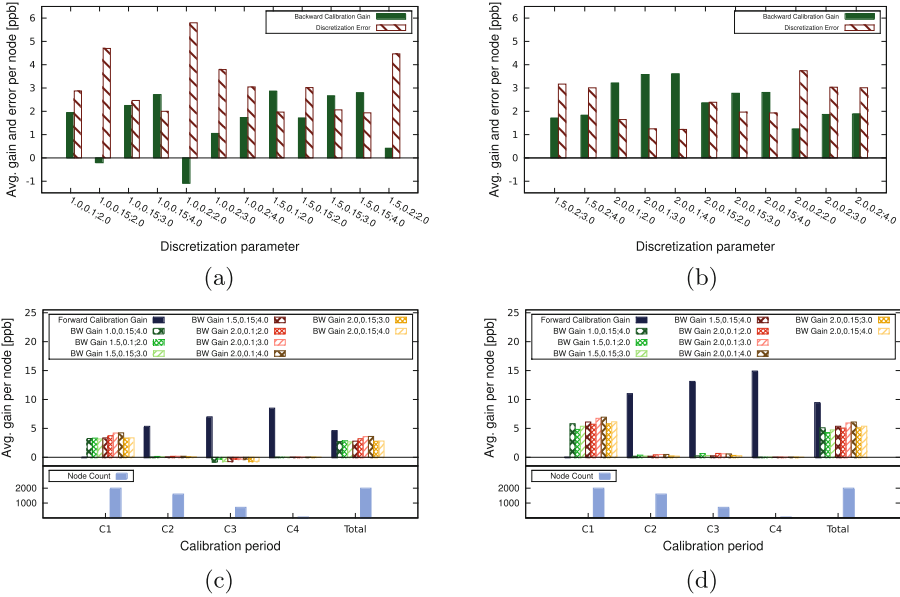


Fig. 4. (a, b) Average backward calibration gain and discretization error for varying discretization parameters over the simulated 14-day period. (c, d) Comparison of forward and backward calibration gain per calibration period with varying aging factors.

set $C_{Timeout} = 5$ for our simulations, a maximum of three calibration points was possible and consequently a maximum of four calibration periods (C_1 to C_4).

The upper parts of Fig. 4c, d show the average calibration gain for the individual calibration periods and the overall gain, whereas the lower parts show the number of nodes that were calibrated in the individual round. In each figure, the forward calibration gain was only plotted once, since forward calibration does not depend on discretization parameters. We illustrated the results for 2000 nodes and chose those discretization parameters, whose backward calibration gain was higher than the discretization error (see Fig. 4a, b). In Fig. 4c, the results with the aforementioned aging factor of 0.2 ppb/day are illustrated. In period C_1 no forward calibration gain is achieved, as forward calibration adapts only future measurements, i.e., from t_{c_1} onwards. But for the following rounds, an increasing forward calibration gain can be observed, however, with a strongly decreasing number of nodes. The backward calibration has the highest impact in C_1 , as uploaded values in this period are completely uncalibrated. In the following rounds, the backward calibration is comparatively small and in the third round even negative. This stems from the relatively short time interval between the calibration points. In C_3 , the sensors have already been calibrated twice and the aging factor does not distort the measurements strongly enough within this calibration interval, so that the discretized backward calibration is not reason-

able in this case. In Fig. 4d, we increased the aging factor to 1.2 ppb/day. This simulates a stronger aging of the sensors, but can also be interpreted as longer periods between the calibration points with a constant aging factor (i.e., 6 times longer calibration intervals with an aging factor of 0.2 ppb/day). It can be seen that both the forward and the backward calibration gain increased; the latter now results in a positive gain in each round. As could be expected, this shows that backward calibration is reasonable if the calibration interval is long enough for the sensors to significantly deviate from their former calibration.

6 Conclusion

We presented a privacy-preserving calibration system that enables forward as well as backward calibration, while simultaneously protecting the users' privacy. We proposed a pseudonym-based system that allows for transferring calibration parameters to other pseudonyms without revealing the connection between those. Our analysis shows that we can achieve a high degree of anonymity, but only at the price of sacrificing precision. More precisely, the anonymity level and the backward calibration gain are negatively correlated, i.e., an increase of the one leads to a decrease of the other. Our results show that there are several discretization parameters that lead to promising results for both, however, the "optimal" setting depends on the application scenario and the subsequent weighting of anonymity in relation to precision. As the loss of precision is small in relation to the overall gain, we believe that PPCS represents a valid concept for privacy-preserving calibration in PSNs. In future work, we want to evaluate our concept with more extensive simulations using a realistic urban simulation environment and implement a prototype to evaluate the concept in real-life settings.

References

1. Lane, N.D., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., Campbell, A.T.: A survey of mobile phone sensing. *IEEE Commun.* **48**(9), 140–150 (2010)
2. Campbell, A.T., Eisenman, S.B., Lane, N.D., Miluzzo, E., Peterson, R.A., Lu, H., Zheng, X., Musolesi, M., Fodor, K., Ahn, G.-S.: The rise of people-centric sensing. *Internet Comput.* **12**(4), 12–21 (2008). (IEEE)
3. Chaum, D.L.: Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM* **24**(2), 84–90 (1981)
4. Burke, J., Estrin, D., Hansen, M., Parker, A., Ramanathan, N., Reddy, S., Srivastava, M.B.: Participatory sensing. In: *Proceedings of WSW'06 at SenSys '06* (2006)
5. Bychkovskiy, V., Megerian, S., Estrin, D., Potkonjak, M.: A collaborative approach to in-place sensor calibration. In: Zhao, F., Guibas, L.J. (eds.) *IPSN 2003*. LNCS, vol. 2634, pp. 301–316. Springer, Heidelberg (2003)
6. Miluzzo, E., Lane, N.D., Campbell, A.T., Olfati-Saber, R.: CaliBree: a self-calibration system for mobile sensor networks. In: Nikolettseas, S.E., Chlebus, B.S., Johnson, D.B., Krishnamachari, B. (eds.) *DCOSS 2008*. LNCS, vol. 5067, pp. 314–331. Springer, Heidelberg (2008)

7. Honicky, R.E.: Automatic calibration of sensor-phones using gaussian processes. Technical report UCB/EECS-2007-34, EECS, UC Berkeley, March 2007
8. Hasenfratz, D., Saukh, O., Thiele, L.: On-the-fly calibration of low-cost gas sensors. In: Picco, G.P., Heinzelman, W. (eds.) EWSN 2012. LNCS, vol. 7158, pp. 228–244. Springer, Heidelberg (2012)
9. Huan, C., Zhiyu, L., Gang, F.: Analysis of the aging characteristics of SnO₂ gas sensors. *Sens. Actuators* **156**(2), 912–917 (2011)
10. Kamionka, M., Breuil, P., Pijolat, C.: Calibration of a multivariate gas sensing device for atmospheric pollution measurement. *Sens. Actuators B* **118**(12), 323–327 (2006)
11. Sweeney, L.: k-anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **10**(5), 557–570 (2002)
12. Shepard, D.: A two-dimensional interpolation function for irregularly-spaced data. In: *Proceedings of the ACM '68*, New York, NY, USA, pp. 517–524. ACM (1968)