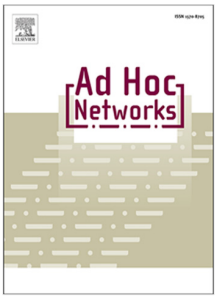
Journal Pre-proof



Confidence interval estimation for fingerprint-based indoor localization

Mohammad Nabati, Seyed Ali Ghorashi, Reza Shahbazian

PII: S1570-8705(22)00072-5
DOI: https://doi.org/10.1016/j.adhoc.2022.102877
Reference: ADHOC 102877

To appear in: *Ad Hoc Networks*

Received date : 20 April 2021
Revised date : 23 January 2022
Accepted date : 25 April 2022

Please cite this article as: M. Nabati, S.A. Ghorashi and R. Shahbazian, Confidence interval estimation for fingerprint-based indoor localization, *Ad Hoc Networks* (2022), doi: https://doi.org/10.1016/j.adhoc.2022.102877.

Confidence Interval Estimation for Fingerprint-based Indoor Localization

Mohammad Nabati[a], Seyed Ali Ghorashi[b,a,*], Reza Shahbazian[c]

[a] *Cognitive Telecommunication Research Group, Department of Telecommunications, Faculty of Electrical Engineering, Shahid Beheshti University, Tehran, Iran*
[b] *School of Architecture, Computing and Engineering, University of East London, E16 2RD London, U.K.*
[c] *Department of Electrical Engineering, Faculty of Technology and Engineering, Standard Research Institute, Alborz, Iran.*

Abstract

Fingerprint-based localization methods provide high accuracy location estimation, which use machine learning algorithms to recognize the statistical patterns of collected data. In these methods, the users' locations can be estimated based on the received signal strength vectors from some transmitters. However, the data collection is a labor-intensive phase, and the collected data should be updated periodically. Many researchers have contributed to reducing this cost. The easiest way to remove the data collection cost is to use fingerprints generated by the model-based approaches, in which the trained machine learning algorithm can be updated based on the environment changes. Probabilistic-based localization algorithms, in addition to the user location, can estimate a region of interest called 2σ confidence interval in which the probability of user presence is 95%. Gaussian process regression (GPR) is a probabilistic method that can be used to achieve this goal. However, conventional GPR (CGPR) cannot accurately estimate the confidence interval when noise-free fingerprints generated by the model-based approaches are used in the training phase. In this paper, we propose a novel GPR-based localization algorithm, named enhanced GPR (EGPR), which improves the accuracy level of confidence interval estimation compared to the existing methods while fixing the level of computational

*Corresponding author

complexity in the online phase. We also theoretically prove that GPR-based algorithms are minimum variance unbiased and efficient estimators. Experiments under line-of-sight and non-line-of-sight conditions demonstrate the superiority of our proposed method over counterparts in terms of accuracy as well as applicability in real-time localization systems.

Keywords: Fingerprint-based Localization, Gaussian Process Regression, Minimum Variance Unbiased, Cramer-Rao Lower Bound.

1. Introduction

Recently, indoor location-based services have attracted a fair amount of attention in a wide variety of applications [1, 2]. Available services that use the global positioning system can be employed for most of the application requirements. However, such methods do not provide high accuracy performance in indoor environments due to the limited coverage of satellites and non-line-of-sight (NLOS) errors [3, 4]. Satellite-based methods regularly use ranging information that can be obtained from techniques such as time of arrival (TOA), angle of arrival (AOA), and received signal strength (RSS) [2].

The time of transferred signals are measured in TOA based methods [5, 6], and the distance between transmitter and receiver can be calculated by multiplying the measurement time and the speed of light [7]. AOA can be calculated by antenna arrays [8], however, even in modern smartphones, the number of antennas is still less than three, and it is hard to measure the angles information. Ranging information can also be extracted via RSS by using the path-loss model. Among these, the RSS is still a well-known method to acquire the distance between the transmitter and receiver, since it does not implicate hardware complexities [1]. After calculating the distance between the transmitters and a receiver, the trilateration or triangulation methods can estimate the users' location. The main concern about the ranging information is that it suffers from NLOS error [3].

Fingerprint-based localization is one of the most popular and effective meth-

ods to provide high accuracy, since it considers the NLOS conditions during the data collection phase [3]. The localization process in this method is divided into two phases: training (offline) and test (online). In the training phase, fingerprints of located base stations (BSs) such as RSS or channel state information (CSI) in some known locations—called reference points (RPs)—are gathered and saved in a database. These BSs are also known as access points (APs) in the literature [9]. Some of the most frequently used signals are Wi-Fi [10, 11], Bluetooth [12], ZigBee [13], and light [14]. Among these, Wi-Fi is the most popular one [15, 16] because of its accessibility in most environments, including offices, buildings, shopping malls, and museums [17]. Also, most of the portable devices provide this standard communication technology [1, 3]. After the training phase, the trained model can map the online (test) RSS or CSI data to location coordinates.

Recently, researchers propose to use the CSI as fingerprints of RPs for accuracy enhancement of location estimation [18, 19]. CSI is stable [19] and provides more information than RSS [18]; however, measuring this information needs special network interface controllers and software tools [1, 20]. Thus, the RSS is still the preferred solution, especially when it comes to practical aspects of popular applications. Fingerprint-based localization methods can be non-probabilistic such as k-nearest neighbors (KNN) [3], support vector regression (SVR) [21] or can be probabilistic-based ones such as Gaussian process regression (GPR) [22], KL divergence [23], and expectation-maximization (EM) [24]. In probabilistic methods, besides the location estimation, it is possible to compute the confidence interval [25] and Cramer-Rao lower bound (CRLB) [22] to be used in realistic applications and to determine the theoretical accuracy, respectively.

The data collection phase is labor-intensive, and many researchers have contributed to solve this issue. The synthetic noise-free fingerprints generated by model-based approaches (i.e., path-loss model) can be used to remove the data collection in the offline phase. Unlike the obtained distances by path-loss model for trilateration or triangulation methods [3], the NLOS condition can be detected in the offline phase for generating synthetic fingerprints. In the online

phase of localization, the RSS samples suffer from small-scale fading and shadowing effects. The small-scale fading can be removed by averaging out the RSS samples; however, the shadowing noise cannot easily be alleviated. Therefore, the shadowing noise remains in the online phase.

Conventional GPR (CGPR) cannot accurately estimate the confidence interval for synthetic noise-free fingerprints. In this paper, we propose a GPR-based algorithm called enhanced GPR (EGPR) to intensify the accuracy level in estimating the confidence interval. EGPR consists of two optimization and tuning parts to estimate the 2σ confidence interval more accurately. Besides, most of the existing works do not consider the theoretical analysis of their proposed methods. Here, we demonstrate that a GPR-based algorithm is a minimum variance unbiased (MVU) and an efficient estimator. It means that the practical accuracy can reach the CRLB. To the best of our knowledge in fingerprint-based localization, this is the first time that it is proven whether the practical accuracy can reach the theoretical bound (CRLB) or not. To summarize, the main contributions of this paper can be listed as follows:

- We propose a Gaussian process regression-based algorithm, named EGPR, to increase the accuracy of the confidence interval estimation.
- We demonstrate that a Gaussian process regression-based algorithm is MVU and efficient.
- We provide a complexity analysis for the proposed EGPR algorithm and demonstrate that its complexity at the test (online) phase is the same as CGPR.

The rest of this paper is organized as follows: In section 2, the related works are described. In section 3 the system model and proposed localization algorithm are described in detail. Theoretical and complexity analysis is provided in section 4. Experimental results and corresponding discussions are presented in section 5, and finally, section 6 concludes the paper.

In this paper we use lower-case letters to denote scalars (e.g., a), boldface

lower-case letters for vectors (e.g., $\mathbf{a}$), and boldface capital letters to present matrices (e.g., $\mathbf{A}$). Moreover, a_i shows the i^{th} element of vector $\mathbf{a}$ and a_{ij} denotes the element within the i^{th} row and j^{th} column of matrix $\mathbf{A}$. Also $\mathbf{a}_i$ shows the i^{th} row of matrix $\mathbf{A}$. The symbols $(\widetilde{\cdot})$ and $(\widehat{\cdot})$ are used to indicate the train and test data, respectively.

2. Related Works

Although several accurate algorithms for fingerprint-based indoor localization are proposed in the literature, most of them are not yet appropriate for practical scenarios due to their critical need for costly data-gathering processes in the training phase and high level of complexity.

In [22], the researchers tried to remove the need for data gathering step in the training phase by generating synthetic simulated data using a model-based approach (i.e., path-loss model). They found that conventional GPR (CGPR) cannot reliably estimate the confidence interval for this noise-free training phase. Therefore, they proposed a numerical approximation GPR (NaGPR) algorithm, which significantly increases the computational complexity at the online phase of localization. Authors in [26] used generative adversarial networks (GANs) to generate massive synthetic training data out of a small set of really gathered data for the cases in which limited training data is available. However, this technique forces a considerable computational cost at the training phase due to the usage of GANs for each class, separately. Authors in [24] used supporting sets that are subsets of available fingerprints similar to the test samples and a probabilistic-based EM algorithm estimates the users' locations. The proposed method forces a high calculation cost to the localization process, since the EM algorithm is implemented in the test (online) phase. Also, the ability of EM to estimate the confidence interval is not considered by [24]. In [10], the authors proposed a localization method that employs the GPR to learn the distribution of available fingerprints. Then, the obtained distribution is used for generating synthetic training data to be added to existing gathered training data, and

weighted KNN is then used for location estimation procedure. However, the confidence interval cannot be estimated in this method due to the nature of the implemented technique. Authors in [27] proposed a crowdsourcing-based probabilistic approach in which active participants in unfixed data points update the available dataset. The proposed method needs at least a robot to move in known track lines of the environment, and this forces huge equipment costs in the offline phase. Also, the crowdsourcing-based algorithms cannot guarantee the desired performance as the gathered data are very noisy, and this leads to inaccurate relations between RSS vectors and locations [28].

When localization systems are supposed to be implemented on the user side (mostly because of privacy reasons), NaGPR is no longer feasible due to its computational complexity in the test (online) phase, whereas, our proposed EGPR algorithm does not increase the complexity in the test phase, as can be seen in section 4.

3. System Model

In this section, we describe the underlying architecture of the proposed EGPR-based localization, which can be used to estimate the confidence interval accurately.

3.1. preliminaries

RSS fingerprint-based methods measure and store the received signal strength (called fingerprint) from APs in specific coordinates called reference points (RPs), as depicted in Fig. 1. These RSS vectors are used in the offline phase of the localization process. Among the fingerprint-based algorithms, the probabilistic ones can estimate 2σ confidence interval in addition to the coordinates themselves. The 2σ confidence interval shows a region in which the user is located with a probability of 95%.

Two types of fingerprints are used for the training phase, including noise-free and real noisy fingerprints. The real noisy fingerprint is one that we measure in a

real environment. Although this scenario considers all complexities of the indoor area, it is labor-intensive, and the collected fingerprints need to be updated periodically due to the environment changes or displacement of access points [29]. Small-scale fading and shadowing noise are the main concerns in real data gathering scenario [22]. To mitigate the small-scale fading effect, we can average out the RSS samples obtained from multiple times [30], while the shadowing noise is space-dependent and the spatial averaging is needed to reduce the effect of this noise. The spatial averaging is impossible for the test phase because the test location is not available in the online phase [31]. However, we can simulate synthetic noise-free data using the path-loss model for the training phase [22, 31]. In this scenario, the train data does not have shadowing noise, while the noise is added to the test data. To construct a map in the noise-free scenario, we only need to know the APs' and walls' locations, which consumes fewer efforts than RSS data collection. By doing so, we can consider the NLOS condition, which is the utmost concern of trilateration or triangulation methods [3]. The trained algorithm can easily be updated in the noise-free scenario based on the environment variations such as APs' locations displacement.

In the training phase, we should obtain two functions for location estimation (μ_x and μ_y) and two functions for variance estimation (φ_x and φ_y) using the training dataset. For the training phase, we consider a dataset consists of RPs' locations and their corresponding fingerprints as follows

$$\tilde{\mathbf{R}} = [\tilde{\mathbf{r}}_1, \tilde{\mathbf{r}}_2, \cdots, \tilde{\mathbf{r}}_{\tilde{N}}]^T, \qquad \begin{matrix} \tilde{\mathbf{x}} = [\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_{\tilde{N}}]^T \\ \tilde{\mathbf{y}} = [\tilde{y}_1, \tilde{y}_2, ..., \tilde{y}_{\tilde{N}}]^T, \end{matrix} \tag{1}$$

where $\tilde{\mathbf{R}} \in \mathbb{R}^{\tilde{N} \times M}$ is the RSS matrix called fingerprints of RPs, $\tilde{\mathbf{r}}_i$ is the vector of the received signal from all APs in the i^{th} RP, $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$ are Cartesian coordinates of the RPs, M is the number of APs, and $\tilde{N}$ is the number of RPs. The location estimation functions map RSS vectors $\mathbf{r}_i$ to the two-dimensional (2D) Cartesian

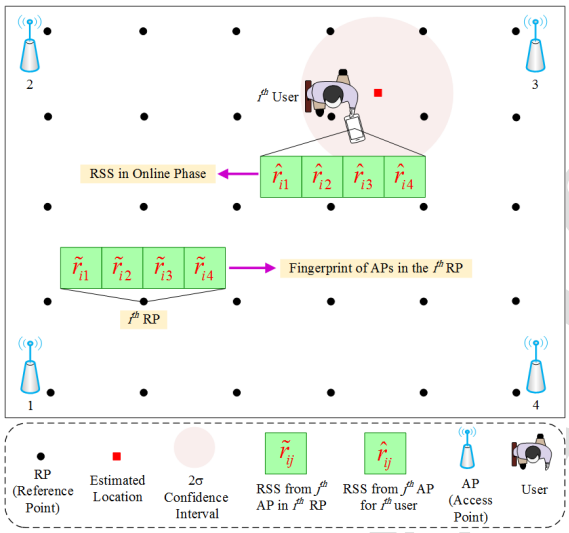


Figure 1: Schematic of fingerprint-based localization in which the RSS vectors received from APs (dBm) are recorded at RPs in the offline phase.

coordinates as described in (2)

$$\begin{aligned} &x_i = \mu_x(\mathbf{r}_i) + \varepsilon_x \quad \text{and} \quad y_i = \mu_y(\mathbf{r}_i) + \varepsilon_y \qquad \forall \mathbf{r}_i \in \{\tilde{\mathbf{r}}_i, \hat{\mathbf{r}}_i\} \\ &\varepsilon_x \sim \mathcal{N}(0, \sigma_n^x) \quad \text{and} \quad \varepsilon_y \sim \mathcal{N}(0, \sigma_n^y). \end{aligned} \tag{2}$$

Both x and y coordinates are estimated with separate optimized functions μ_x and μ_y. The functions φ_x and φ_y estimate the variance of ε_x and ε_y, respectively. Then, the 2σ confidence interval can be estimated by $\mu_x \pm 2\sqrt{\varphi_x}$ and $\mu_y \pm 2\sqrt{\varphi_y}$. Since the optimization process is the same for both coordinates, without loss of generality, we use f to represent x or y, and the coordinate vectors in (1) (i.e. $\tilde{\mathbf{x}}$ or $\tilde{\mathbf{y}}$) can be considered as $\tilde{\mathbf{f}} = [\tilde{f}_1, \tilde{f}_2, ..., \tilde{f}_{\tilde{N}}]^T$. Also, Eq. (2) can be re-written as follows

$$f_i = \mu(\mathbf{r}_i) + \varepsilon \text{ and } \varepsilon \sim \mathcal{N}(0, \sigma_n). \tag{3}$$

The $\tilde{f}_i^{th}$ in training data can be considered to have a joint Gaussian distribution. For simplicity and without loss of generality, we assume a zero-mean Gaussian process, and therefore, the vector $\tilde{\mathbf{f}} = [\tilde{f}_1, \tilde{f}_2, ..., \tilde{f}_{\tilde{N}}]^T$ has the following distri-

Table 1: Kernel functions that can be used to capture the similarity between two pairs of RSS vectors.

Kernel Name	Function	Hyper-parameter(s)
Squared Exponential (SE)	$\mathcal{K}_{SE}(\mathbf{r}_i, \mathbf{r}_j) = \gamma^2 \exp\left(-\frac{d^2(\mathbf{r}_i, \mathbf{r}_j)}{l^2}\right)$	γ and l
Linear (Lin)	$\mathcal{K}_{Lin}(\mathbf{r}_i, \mathbf{r}_j) = \gamma^2 \omega(\mathbf{r}_i, \mathbf{r}_j)$	γ
Noise (n)	$\mathcal{K}_n(\mathbf{r}_i, \mathbf{r}_j) = \sigma_n^2 \delta_{ij}$ $\delta_{ij} = \{1 \text{ if } i = j,\ 0 \text{ otherwise}\}$	σ_n

bution [32]

$$\tilde{\mathbf{f}} \sim \mathcal{GP}(\mathbf{0}, \mathbf{C}), \tag{4}$$

where $\mathbf{C} \in \mathbb{R}^{\tilde{N} \times \tilde{N}}$ is the covariance matrix of the training data. Each element of this matrix demonstrates the similarity between two elements of the vector $\tilde{\mathbf{f}}$. This similarity can be captured by different kernel functions such as linear, squared exponential, and Noise kernel, as depicted in Table 1 [32].

In Table 1, $\mathcal{K}_{SE}$ is the squared exponential kernel that captures non-linear dependencies of RSS samples, $\mathcal{K}_{Lin}$ is the linear kernel that captures linear dependencies of RSS samples, and $\mathcal{K}_n$ is the Noise kernel that models the variance of ε in (3). The Noise kernel is independent of inputs, however, it increases the similarity of diagonal elements and plays a pivotal role in estimating the confidence interval. Finally, $d(.,.)$ and $\omega(.,.)$ represent the Euclidean distance and inner product, respectively, which are defined as follows

$$d(\mathbf{r}_i, \mathbf{r}_j) = \sqrt{(r_{1i} - r_{1j})^2 + ... + (r_{Mi} - r_{Mj})^2}, \tag{5}$$

$$\omega(\mathbf{r}_i, \mathbf{r}_j) = (r_{1i} \cdot r_{1j}) + ... + (r_{Mi} \cdot r_{Mj}), \tag{6}$$

where $\mathbf{r}_i$ and $\mathbf{r}_j$ are two RSS vectors, and M is the number of APs in the environment. A single kernel can only capture one characteristic of RSS samples

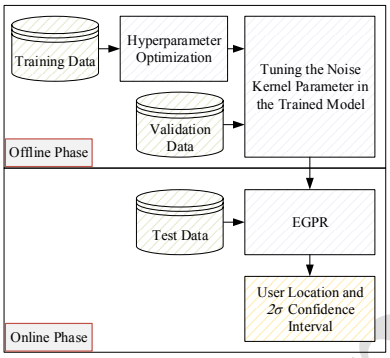


Figure 2: Architecture of proposed EGPR in offline and online phases.

[10], and therefore, we have preferred to use the combination of different kernels.

The architecture of the proposed fingerprinting-based EGPR is depicted in Fig. 2. As can be seen, the training data is employed to optimize the hyperparameters of kernel functions. In the next step, the Noise kernel parameter is tuned by using the validation data to estimate the confidence interval accurately. After the offline phase, the system is ready to be utilized in the online phase to determine the location and its corresponding confidence interval for the new input, also known as "test data".

3.2. Training Phase

GPR-based algorithms are specified by kernel functions with their hyperparameters where should be optimized in the training phase [10]. In this section, we present the hyperparameter optimization procedure for a combination of three different kernels of Table 1 to capture linear and non-linear dependencies of RSS distribution

$$\begin{aligned} c_{ij} &= \mathcal{K}_{SE}(\tilde{\mathbf{r}}_i, \tilde{\mathbf{r}}_j) + \mathcal{K}_{Lin}(\tilde{\mathbf{r}}_i, \tilde{\mathbf{r}}_j) + \mathcal{K}_n(\tilde{\mathbf{r}}_i, \tilde{\mathbf{r}}_j) \\ &= \gamma_1^2 \exp\left(-\frac{d^2(\tilde{\mathbf{r}}_i, \tilde{\mathbf{r}}_j)}{l^2}\right) + \gamma_2^2 \omega(\tilde{\mathbf{r}}_i, \tilde{\mathbf{r}}_j) + \sigma_n^2 \delta_{ij}, \\ &\text{where}, \delta_{ij} = \{1 \text{ if } i = j,\ 0 \text{ otherwise}\}, \end{aligned} \tag{7}$$

In (7), c_{ij} is the element at the i^{th} row and j^{th} column of $\mathbf{C}$. The vector $\boldsymbol{\theta} = [\gamma_1, \gamma_2, l, \sigma_n]^T$ contains all hyperparameters that should be optimized in the training phase. The hyperparameters in $\boldsymbol{\theta}$ can be estimated from an optimization problem to maximize the log-likelihood of the multivariate probability density function of training samples or to minimize the negative ones as below

$$\tilde{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \log(p(\tilde{\mathbf{f}})) = \arg\min_{\boldsymbol{\theta}}(-\log(p(\tilde{\mathbf{f}}))), \tag{8}$$

where $p(\tilde{\mathbf{f}})$ is the multivariate probability density function defined in (9)

$$p(\tilde{\mathbf{f}}) = \frac{1}{(2\pi)^{\tilde{N}/2}|\mathbf{C}|^{1/2}} exp(-\frac{1}{2}\tilde{\mathbf{f}}^T\mathbf{C}^{-1}\tilde{\mathbf{f}}). \tag{9}$$

Therefore, the objective function which should be optimized in (8) can be written as follows

$$\mathcal{L}(\boldsymbol{\theta}) = -\log p(\tilde{\mathbf{f}}) = \frac{1}{2}\log|\mathbf{C}| + \frac{\tilde{N}}{2}\log(2\pi) + \frac{1}{2}\tilde{\mathbf{f}}^T\mathbf{C}^{-1}\tilde{\mathbf{f}}. \tag{10}$$

The Eq. (8) shows a non-convex optimization problem; however, we can use the gradient-based optimizers (e.g., conjugate gradient [33] or gradient descent [34]) to solve the problem for a locally optimum point. The gradient of $\mathcal{L}(\boldsymbol{\theta})$ for each hyperparameter in optimization process can be calculated as follows

$$\begin{aligned} \nabla\mathcal{L}(\theta_j) &= \frac{\partial(-\log p(\tilde{\mathbf{f}}))}{\partial\theta_j} \\ &= -\frac{1}{2}\mathrm{tr}((\mathbf{s}\mathbf{s}^T - \mathbf{C}^{-1})\frac{\partial\mathbf{C}}{\partial\theta_j}) \ \text{ where } \ \mathbf{s} = \mathbf{C}^{-1}\tilde{\mathbf{f}}, \end{aligned} \tag{11}$$

where $\frac{\partial\mathbf{C}}{\partial\theta_j} \in \mathbb{R}^{\tilde{N}\times\tilde{N}}$ is a symmetric matrix, and each element of this matrix is calculated by the gradient of kernel function with respect to the i^{th} hyperparameter. After calculating $\nabla\mathcal{L}(\theta_i)$ for the i^{th} hyperparameter, it can be fed into the i^{th} element of a vector $\boldsymbol{\vartheta}$, called gradient vector. The hyperparameters in $\boldsymbol{\theta}$ can be updated until reaching a convergence by using the gradient descent or conjugate gradient algorithm. Here, we use the conjugate gradient algorithm as

provided in Appendix B.

Considering σ_n in (7) as a learning parameter can capture the noise when real noisy data is used for the training phase, because the noise extracted from training data can be used in the test phase to estimate the confidence interval. However, this scheme cannot be used in the noise-free scenario, because unlike the test observations, the train observations are not noisy and the noise cannot be estimated with the training dataset. In section 3.4, we propose the tuning procedure to overcome this problem.

3.3. Test Phase

We need to find the posterior distribution to estimate the users' locations and their confidence intervals. The GPR is able to provide both information at the same time. First, the joint distribution of the train and test samples can be written as follows

$$\begin{bmatrix} \tilde{\mathbf{f}} \\ \hat{\mathbf{f}} \end{bmatrix} \sim \mathcal{N}\left[\begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} \mathbf{C} & \mathbf{C}(\tilde{\mathbf{r}}, \hat{\mathbf{r}}) \\ \mathbf{C}(\hat{\mathbf{r}}, \tilde{\mathbf{r}}) & \mathbf{C}(\hat{\mathbf{r}}, \hat{\mathbf{r}}) \end{pmatrix} \right], \tag{12}$$

where $\mathbf{C} \in \mathbb{R}^{\tilde{N} \times \tilde{N}}$ is the covariance matrix calculated from the train observations, $(\mathbf{C}(\tilde{\mathbf{r}}, \hat{\mathbf{r}}))^T = \mathbf{C}(\hat{\mathbf{r}}, \tilde{\mathbf{r}}) \in \mathbb{R}^{\hat{N} \times \tilde{N}}$ is the covariance matrix calculated from the test and train observations, and $\mathbf{C}(\hat{\mathbf{r}}, \hat{\mathbf{r}}) \in \mathbb{R}^{\hat{N} \times \hat{N}}$ is the covariance matrix calculated from the test observations. All elements of covariance matrices in (12) are calculated with kernel functions obtained from (8). Also, $\hat{\mathbf{f}}$ is the vector of real test values (users' locations) that should be estimated with conditioning over the $\tilde{\mathbf{f}}$ as follows [32]

$$\begin{aligned} &\hat{\mathbf{f}}|\tilde{\mathbf{f}} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Phi}) \\ &\boldsymbol{\mu} = \mathbf{C}(\hat{\mathbf{r}}, \tilde{\mathbf{r}})\, \mathbf{C}^{-1} \tilde{\mathbf{f}} \\ &\boldsymbol{\Phi} = \mathbf{C}(\hat{\mathbf{r}}, \hat{\mathbf{r}}) - \mathbf{C}(\hat{\mathbf{r}}, \tilde{\mathbf{r}}) \mathbf{C}^{-1} \mathbf{C}(\tilde{\mathbf{r}}, \hat{\mathbf{r}}), \end{aligned} \tag{13}$$

where $\boldsymbol{\mu}$ is the vector of estimated coordinates and $\boldsymbol{\Phi}$ is the corresponding covariance matrix for all users. The diagonal elements of $\boldsymbol{\Phi}$ represent the variance

of estimated locations in $\boldsymbol{\mu}$. Therefore, the 2σ confidence interval for the i^{th} user can be calculated by $\mu_i \pm 2\sqrt{\varphi_{ii}}$, where μ_i is the i^{th} element of $\boldsymbol{\mu}$ and φ_{ii} is the i^{th} diagonal element of $\boldsymbol{\Phi}$.

The optimized hyperparameters in (8) and the training data in (1) can be delivered to smartphones or other portable devices to estimate the location and confidence interval from (13). Note that when the location estimation is performed on the user side, $\boldsymbol{\mu}$ and $\boldsymbol{\Phi}$ have only one element, and each user performs the (13) separately. However, we explained the test phase of the Gaussian process in (13) for a general form that can calculate all users' locations and corresponding covariance matrix simultaneously. The diagonal elements of $\boldsymbol{\Phi}$ for synthetic noise-free fingerprints are small and cannot be used to determine the 2σ confidence interval accurately [15]. In section 3.4, we propose tuning procedure to estimate the 2σ confidence interval for noise-free scenarios.

3.4. Tuning The Parameter of Noise Kernel $\mathcal{K}_n$

The parameter of Noise kernel $\mathcal{K}_n$ is chosen during the optimization of (8). However, experimental results show that it is not accurate to estimate the 2σ confidence interval for the synthetic noise-free training fingerprints [15]. In this section, we explain the tuning procedure of the Noise kernel parameter σ_n to overcome the small estimation of the confidence interval. To tune the Noise kernel parameter, we need to find the relation between $\boldsymbol{\Phi}$ and σ_n^2. In other word, the effect of σ_n^2 on the $\boldsymbol{\Phi}$ should be known when it increases. First, by using the mathematical formulation, we show that by adding σ_n^2 to diagonal elements of $\mathbf{C}$ and $\mathbf{C}(\hat{\mathbf{r}}, \hat{\mathbf{r}})$ in (13), the variance of posterior distribution $\boldsymbol{\Phi}$ will increase. Then, we propose a method to choose the proper value of σ_n. Considering the added parameter to diagonal elements of $\mathbf{C}$ and $\mathbf{C}(\hat{\mathbf{r}}, \hat{\mathbf{r}})$ in (13), we can write

$$\boldsymbol{\Phi} = \sigma_n^2\mathbf{I} + \mathbf{C}(\hat{\mathbf{r}}, \hat{\mathbf{r}}) - \mathbf{C}(\hat{\mathbf{r}}, \tilde{\mathbf{r}})\,(\sigma_n^2\mathbf{I} + \mathbf{C})^{-1}(\mathbf{C}(\hat{\mathbf{r}}, \tilde{\mathbf{r}}))^T. \tag{14}$$

The second term $\mathbf{C}(\hat{\mathbf{r}}, \hat{\mathbf{r}})$ already exists on the $\boldsymbol{\Phi}$ in (13) and does not change the diagonal elements of $\boldsymbol{\Phi}$. It is clear that $\sigma_n^2\mathbf{I}$ in the first term, is positive and

increases the diagonal elements of $\mathbf{\Phi}$. Expansion of the third term in (14) is as follows

$$\begin{aligned}
&\mathbf{C}(\hat{\mathbf{r}},\tilde{\mathbf{r}})\,(\sigma_n^2\mathbf{I}+\mathbf{C})^{-1}(\mathbf{C}(\hat{\mathbf{r}},\tilde{\mathbf{r}}))^T \\
&= \mathbf{C}(\hat{\mathbf{r}},\tilde{\mathbf{r}})\,(\sigma_n^2\,\mathbf{C}^{-1}\mathbf{C}+\mathbf{C}\mathbf{C}^{-1}\mathbf{C})^{-1}(\mathbf{C}(\hat{\mathbf{r}},\tilde{\mathbf{r}}))^T \\
&= \mathbf{C}(\hat{\mathbf{r}},\tilde{\mathbf{r}})((\sigma_n^2\,\mathbf{C}^{-1}+\mathbf{C}\mathbf{C}^{-1})\mathbf{C})^{-1}(\mathbf{C}(\hat{\mathbf{r}},\tilde{\mathbf{r}}))^T \\
&= \mathbf{C}(\hat{\mathbf{r}},\tilde{\mathbf{r}})\mathbf{C}^{-1}(\sigma_n^2\,\mathbf{C}^{-1}+\mathbf{I})^{-1}(\mathbf{C}(\hat{\mathbf{r}},\tilde{\mathbf{r}}))^T \\
&\overset{\text{ts}}{=} \mathbf{C}(\hat{\mathbf{r}},\tilde{\mathbf{r}})\,\mathbf{C}^{-1}(\mathbf{I}-\sigma_n^2\,\mathbf{C}^{-1}+(\sigma_n^2\,\mathbf{C}^{-1})^2-...)(\mathbf{C}(\hat{\mathbf{r}},\tilde{\mathbf{r}}))^T \\
&\simeq \mathbf{C}(\hat{\mathbf{r}},\tilde{\mathbf{r}})\mathbf{C}^{-1}(\mathbf{I}-\sigma_n^2\,\mathbf{C}^{-1})(\mathbf{C}(\hat{\mathbf{r}},\tilde{\mathbf{r}}))^T \\
&= \mathbf{C}(\hat{\mathbf{r}},\tilde{\mathbf{r}})\mathbf{C}^{-1}(\mathbf{C}(\hat{\mathbf{r}},\tilde{\mathbf{r}}))^T-\sigma_n^2\,\mathbf{C}(\hat{\mathbf{r}},\tilde{\mathbf{r}})\mathbf{C}^{-1}\,(\mathbf{C}^{-1}\mathbf{C}(\hat{\mathbf{r}},\tilde{\mathbf{r}}))^T,
\end{aligned} \tag{15}$$

where "ts" in the fifth line of (15) is the Taylor series. In the last line of (15) the first term $\mathbf{C}(\hat{\mathbf{r}},\tilde{\mathbf{r}})\mathbf{C}^{-1}(\mathbf{C}(\hat{\mathbf{r}},\tilde{\mathbf{r}}))^T$ already exists in (13) and does not change the diagonal elements of $\mathbf{\Phi}$. We need to show that diagonal elements in the second term $\sigma_n^2\,\mathbf{C}(\hat{\mathbf{r}},\tilde{\mathbf{r}})\mathbf{C}^{-1}\,(\mathbf{C}^{-1}\mathbf{C}(\hat{\mathbf{r}},\tilde{\mathbf{r}}))^T$ are positive. The σ_n^2 is positive and therefore, we just consider $\mathbf{C}(\hat{\mathbf{r}},\tilde{\mathbf{r}})\mathbf{C}^{-1}\,(\mathbf{C}^{-1}\mathbf{C}(\hat{\mathbf{r}},\tilde{\mathbf{r}}))^T$. By assuming that $\mathbf{A}=\mathbf{C}(\hat{\mathbf{r}},\tilde{\mathbf{r}})\mathbf{C}^{-1}$, it can be shown that diagonal elements of $\mathbf{A}\mathbf{A}^T$ are positive, and therefore by adding σ_n^2 to diagonal elements of $\mathbf{C}$ and $\mathbf{C}(\hat{\mathbf{r}},\hat{\mathbf{r}})$ the estimated variances in $\mathbf{\Phi}$ will increase. To select the suitable value of σ_n, we carry the σ_n over the values between σ_n^{down} and σ_n^{up} with steps of ς and compute the true estimate region percentage (TERP). The values of σ_n^{down}, σ_n^{up}, and the step size ς depend on the data and are chosen experimentally. The TERP is calculated as follows

$$\begin{aligned}
&\text{TERP}=\left(\frac{1}{\hat{N}}\sum_{i=1}^{\hat{N}}\Gamma_i\right), \\
&\text{where, }\ \Gamma_i=\begin{cases}1 & |\hat{f}_i-\mu_i|\le 2\sqrt{\varphi_{ii}}\\ 0 & \text{otherwise.}\end{cases}.
\end{aligned} \tag{16}$$

The TERP criterion shows the user presence for the estimated 2σ confidence interval. After calculating the TERP values for σ_n in the interval $[\sigma_n^{\text{down}},\sigma_n^{\text{up}}]$, we can choose a proper value for σ_n. We hold all values of σ_n on a vector

$\boldsymbol{\psi}$, where the TERP is between $\mathrm{TERP_d}$ and $\mathrm{TERP_u}$, and then the median of $\boldsymbol{\psi}$ is selected. This process means that the probability of user presence is between $\mathrm{TERP_d}$ and $\mathrm{TERP_u}$. We choose the upper and lower TERP such that $0.9 \leq \mathrm{TERP_d} < \mathrm{TERP_u} \leq 0.99$. Therefore, the values of vector $\boldsymbol{\psi}$ is as follows

$$\boldsymbol{\psi} = \{\sigma_n | \mathrm{TERP_d} \leq \mathrm{TERP}(\sigma_n) \leq \mathrm{TERP_u}\}, \tag{17}$$

where $\boldsymbol{\psi}$ is a vector consists of all values for σ_n in which the TERP is between $\mathrm{TERP_d}$ and $\mathrm{TERP_u}$. The σ_n is selected as below

$$[\sigma_n]_{\mathrm{sel}} = \mathrm{median}\{\boldsymbol{\psi}\}. \tag{18}$$

4. Theoretical Analyses

4.1. CRLB and MVU estimator

The CRLB demonstrates the lowest possible mean squared error (MSE) for an estimator [35]. There are two types of CRLB in the literature for Gaussian processes that already have been derived. One of them discusses the lower bound of errors for estimated hyperparameters $\tilde{\boldsymbol{\theta}}$ [36], and the other one for the estimated locations $\boldsymbol{\mu}$ [22]. Here, the CRLB refers to the lower bound of estimated locations. Considering the existing works on the theoretical analysis of GPR, it has not been proven whether the MSE of GPR can reach the CRLB or not. In this section, we show that the GPR-based algorithms are MVU and efficient estimators. It means that the MSE performance of GPR-based algorithms can reach the CRLB. We use two lemmas as described below.

Lemma 1 (MVU estimator in vector form): *Assume that* $\boldsymbol{\mu} = \hat{\mathbf{f}} + \mathbf{w}$ *where* $\hat{\mathbf{f}}$ *is the vector of real outputs,* $\boldsymbol{\mu}$ *is the vector of predicted outputs, and* $\mathbf{w}$ *is white Gaussian noise. The probability distribution of vector* $\boldsymbol{\mu}$ *is* $p(\boldsymbol{\mu}; \hat{\mathbf{f}}, \boldsymbol{\Phi})$ *where* $\boldsymbol{\Phi}$ *is the covariance matrix. If regularity condition* $E[\frac{\partial log(p(\boldsymbol{\mu}; \hat{\mathbf{f}}, \boldsymbol{\Phi}))}{\partial \hat{\mathbf{f}}}] = 0$ *is satisfied,*

then the covariance matrix of any unbiased estimator must satisfy

$$\bar{\mathbf{\Phi}} \geq \frac{1}{\mathbf{I}(\hat{\mathbf{f}})} \Rightarrow \bar{\mathbf{\Phi}} - \mathbf{I}^{-1}(\hat{\mathbf{f}}) \geq 0,$$
$$\text{where, } \mathbf{I}(\hat{\mathbf{f}}) = \frac{1}{-\mathrm{E}\left[\dfrac{\partial^2 \log(\mathrm{p}(\boldsymbol{\mu}; \hat{\mathbf{f}}, \mathbf{\Phi}))}{\partial \hat{\mathbf{f}}^2}\right]}, \tag{19}$$

In (19), $\bar{\mathbf{\Phi}} = E[(\boldsymbol{\mu} - \hat{\mathbf{f}})(\boldsymbol{\mu} - \hat{\mathbf{f}})^T]$*,* $\mathbf{I}(\hat{\mathbf{f}})$ *is the Fisher information matrix, and* $\bar{\mathbf{\Phi}} - \mathbf{I}^{-1}(\hat{\mathbf{f}}) \geq 0$ *means that the resulting matrix is positive semidefinite. The average of diagonal elements in* $\bar{\mathbf{\Phi}}$ *shows the MSE and the average of diagonal elements in* $\mathbf{I}^{-1}(\hat{\mathbf{f}})$ *shows the CRLB. Also, an unbiased estimator maybe found that attains to the CRLB if and only if* $\frac{\partial \log p(\boldsymbol{\mu}; \hat{\mathbf{f}}, \mathbf{\Phi})}{\partial \hat{\mathbf{f}}} = \mathbf{I}(\hat{\mathbf{f}})(g(\boldsymbol{\mu}) - \hat{\mathbf{f}})$*, where the mentioned estimator* $g(\boldsymbol{\mu})$ *is MVU.*

Lemma 2 (Efficient estimator in vector form): *If the g function is a linear transformation of* $\boldsymbol{\mu}$ *in the Lemma 1 (i.e.* $g(\boldsymbol{\mu}) = \mathbf{A}\boldsymbol{\mu} + \mathbf{b}$ *where* $\mathbf{A} \in \mathbb{R}^{R \times \hat{N}}$ *and* $\mathbf{b} \in \mathbb{R}^{R \times 1}$*), then the* $g(\boldsymbol{\mu})$ *is an efficient estimator.*

Based on Lemma 1 and Lemma 2, we prove that GPR-based algorithms are MVU and efficient. First, MSE for estimated locations is defined as follows

$$\begin{aligned} \mathrm{MSE} &= \frac{1}{\hat{N}} \mathrm{tr}(\bar{\mathbf{\Phi}}^x + \bar{\mathbf{\Phi}}^y) \\ &= \frac{1}{\hat{N}} \left(\sum_{i=1}^{\hat{N}} (\hat{x}_i - \mu_i^x)^2 + (\hat{y}_i - \mu_i^y)^2 \right). \end{aligned} \tag{20}$$

According to Lemma 1, the CRLB is available if the regularity condition is satisfied

$$E\left[\frac{\partial \log p(\boldsymbol{\mu}; \hat{\mathbf{f}}, \mathbf{\Phi})}{\partial \hat{\mathbf{f}}}\right] = 0, \tag{21}$$

where $\hat{\mathbf{f}}$ is a vector that consists of real outputs, $\boldsymbol{\mu}$ is the predicted values by GPR algorithm, $\mathbf{w}$ is white Gaussian noise, and the expectation is taken with

respect to $p(\boldsymbol{\mu}; \hat{\mathbf{f}}, \boldsymbol{\Phi})$. Therefore, we can write

$$\begin{aligned} &\int_{-\infty}^{\infty} \frac{\partial \log p(\boldsymbol{\mu}; \hat{\mathbf{f}}, \boldsymbol{\Phi})}{\partial \hat{\mathbf{f}}} p(\boldsymbol{\mu}; \hat{\mathbf{f}}, \boldsymbol{\Phi}) \; d\boldsymbol{\mu} \\ &= k' \int_{-\infty}^{\infty} \underbrace{(\boldsymbol{\mu} - \hat{\mathbf{f}}) \, exp\left(-(\boldsymbol{\mu} - \hat{\mathbf{f}})^T \boldsymbol{\Phi}^{-1} (\boldsymbol{\mu} - \hat{\mathbf{f}})\right)}_{\text{odd function}} \; d\boldsymbol{\mu} = 0, \end{aligned} \tag{22}$$

where $k' = \boldsymbol{\Phi}^{-1}/(2\pi)^{\frac{\hat{N}}{2}} |\boldsymbol{\Phi}|^{\frac{1}{2}}$, which can be brought out from the integral operator as it does not depend on $\boldsymbol{\mu}$. We can see that the regularity condition is satisfied, and therefore, the CRLB is available. The CRLB can be derived as below

$$\begin{aligned} &\Rightarrow E[(\boldsymbol{\mu} - \hat{\mathbf{f}})(\boldsymbol{\mu} - \hat{\mathbf{f}})^T] \geq \text{CRLB}, \\ &\text{CRLB} = \mathbf{I}(\hat{\mathbf{f}})^{-1} = \frac{1}{-E\left[\dfrac{\partial^2 (\log p(\boldsymbol{\mu}; \hat{\mathbf{f}}, \boldsymbol{\Phi}))}{\partial \hat{\mathbf{f}}^2}\right]}, \\ &\text{where,} \\ &\log p(\boldsymbol{\mu}; \hat{\mathbf{f}}, \boldsymbol{\Phi}) = \\ &\quad -\frac{\hat{N}}{2} \log(2\pi) - \frac{1}{2} \log(|\boldsymbol{\Phi}|) - \frac{1}{2} (\boldsymbol{\mu} - \hat{\mathbf{f}})^T \boldsymbol{\Phi}^{-1} (\boldsymbol{\mu} - \hat{\mathbf{f}}), \\ &\mathbf{I}(\hat{\mathbf{f}}) = -E\left[\frac{\partial^2 (\log p(\boldsymbol{\mu}; \hat{\mathbf{f}}, \boldsymbol{\Phi}))}{\partial \hat{\mathbf{f}}^2}\right] = \boldsymbol{\Phi}^{-1}, \\ &\Rightarrow \; \text{MSE} \geq \frac{1}{\hat{N}} \text{tr}(\boldsymbol{\Phi}), \end{aligned} \tag{23}$$

where $\mathbf{I}(\hat{\mathbf{f}})$ is the Fisher information matrix and for the two-dimensional scenario we have

$$\text{MSE} \geq \frac{1}{\hat{N}} \left(\text{tr}(\boldsymbol{\Phi}^x + \boldsymbol{\Phi}^y)\right) \tag{24}$$

Now we show that GPR is MVU and efficient

$$\begin{aligned}\frac{\partial(\log p(\boldsymbol{\mu};\hat{\mathbf{f}},\boldsymbol{\Phi}))}{\partial\hat{\mathbf{f}}} &= \frac{\partial\left(-\frac{1}{2}(\boldsymbol{\mu}-\hat{\mathbf{f}})^T\boldsymbol{\Phi}^{-1}(\boldsymbol{\mu}-\hat{\mathbf{f}})\right)}{\partial\hat{\mathbf{f}}}\\ &= -\frac{1}{2}\frac{\partial\left((\boldsymbol{\mu}-\hat{\mathbf{f}})\cdot(\boldsymbol{\Phi}^{-1}(\boldsymbol{\mu}-\hat{\mathbf{f}}))\right)}{\partial\hat{\mathbf{f}}}\\ &= -\frac{1}{2}\left(-\boldsymbol{\Phi}^{-1}(\boldsymbol{\mu}-\hat{\mathbf{f}})-\boldsymbol{\Phi}^{-1}(\boldsymbol{\mu}-\hat{\mathbf{f}})\right)\\ &= \boldsymbol{\Phi}^{-1}(\boldsymbol{\mu}-\hat{\mathbf{f}}) = \mathbf{I}(\hat{\mathbf{f}})(\boldsymbol{\mu}-\hat{\mathbf{f}}).\end{aligned} \tag{25}$$

Therefore, $\boldsymbol{\mu}$ is an MVU estimator and as $\boldsymbol{\mu}$ is in the linear form (according to Lemma 2), this MVU estimator is also efficient and can be attained to the lower bound.

Note that we report the root mean squared error (RMSE) in section **??**, therefore without loss of generality the CRLB for RMSE performance can be defined by CRLB $= \sqrt{1/\hat{N}\,(\mathrm{tr}(\boldsymbol{\Phi}^x+\boldsymbol{\Phi}^y))}$. The CRLB, and TERP (see Eq. (16)) depend on diagonal elements of $\boldsymbol{\Phi}$. Also in section 3.4, we showed that the diagonal elements of $\boldsymbol{\Phi}$ are affected by Noise kernel. Therefore, the Noise kernel parameter plays a key role in both CRLB and TERP. Hence, the selected values for $\mathrm{TERP_d}$ and $\mathrm{TERP_u}$ in (17) should not be in such a way that the TERP exceeds 95%. Otherwise, the CRLB is wrongly placed upper than that of RMSE.

4.2. Complexity Analysis

Here, we use the notation $\mathcal{O}$ to demonstrate the order of complexity. In the first step, we will present the complexity of the conventional GPR method in both of the training and test phases. Then, we discuss the complexity of our proposed EGPR. The following explanations are based on the assumption that $M \ll \tilde{N}$ and $\hat{N} \ll \tilde{N}$. We note that $\hat{N} = 1$ when the algorithm is performed on the user side.

Training phase: The conjugate gradient algorithm just needs the first-order gradient $\nabla\mathcal{L}(\boldsymbol{\theta})$ to optimize the hyperparameters, where the complexity of the first-order gradient is $\mathcal{O}(\tilde{N}^3)$, because the inversion matrix has $\mathcal{O}(\tilde{N}^3)$

complexity in (11). The other operations such as calculation of all elements in $\frac{\partial \mathbf{C}}{\partial \theta_j}$ and $\mathbf{C}$ need $\mathcal{O}(M\tilde{N}^2)$ computations. Therefore, the complexity of GPR in training phase is equal to $\mathcal{O}(\tilde{N}^3)$.

Test phase: The complexity of calculating the elements of $\mathbf{C}$ and $\mathbf{C}(\hat{\mathbf{r}}, \tilde{\mathbf{r}})$ matrices in (13) is equivalent to $\mathcal{O}(M\tilde{N}^2)$ and $\mathcal{O}(M\tilde{N}\hat{N})$, respectively. To calculate $\boldsymbol{\mu}$, it needs $\mathcal{O}(\tilde{N}^3)$, $\mathcal{O}(\tilde{N}^2)$ and $\mathcal{O}(\tilde{N}\hat{N})$ operations for $\mathbf{A}_1 = \mathbf{C}^{-1}$, $\mathbf{A}_2 = \mathbf{A}_1\tilde{\mathbf{f}}$ and $\mathbf{C}(\hat{\mathbf{r}}, \tilde{\mathbf{r}})\mathbf{A}_2$, respectively. Therefore, the complexity order of calculating $\boldsymbol{\mu}$ is $\mathcal{O}(\tilde{N}^3)$. Also, calculation of $\boldsymbol{\Phi}$ needs operations with the orders of $\mathcal{O}(\tilde{N}^3)$, $\mathcal{O}(\tilde{N}^2\hat{N})$, and $\mathcal{O}(\hat{N}^2\tilde{N})$ for $\mathbf{A}_1 = \mathbf{C}^{-1}$, $\mathbf{A}_2 = \mathbf{A}_1(\mathbf{C}(\hat{\mathbf{r}}, \tilde{\mathbf{r}}))^T$, and $\mathbf{A}_3 = \mathbf{C}(\hat{\mathbf{r}}, \tilde{\mathbf{r}})\mathbf{A}_2$, respectively. Therefore, the complexity of calculating $\boldsymbol{\Phi}$ is $\mathcal{O}(\tilde{N}^3)$.

Complexity of EGPR and NaGPR: To tune the Noise kernel parameter in the proposed EGPR method, we need to estimate the outputs' mean and variance for $L = \frac{\sigma_n^{\text{up}} - \sigma_n^{\text{down}}}{\varsigma}$ times, and we know that the estimations of $\boldsymbol{\mu}$ and $\boldsymbol{\Phi}$ are in the order of $\mathcal{O}(\tilde{N}^3)$. Therefore, the complexity of this process is $\mathcal{O}(L\tilde{N}^3)$ in the offline phase. The online phase of EGPR is the same as the online phase of CGPR, therefore, the complexity of EGPR in the online phase is $\mathcal{O}(\tilde{N}^3)$. NaGPR [22] calculates the outputs' mean and variance for S times in the online phase, where S is the number of iteration in which the shadowing noise is added to the RSS vectors. Thus, the complexity of NaGPR equals to $\mathcal{O}(S\tilde{N}^3)$ in the online phase. The S is a user-defined value, and the broader S leads to a more accurate confidence interval estimation.

5. Results and discussion

In this section, we perform the experiments on both simulated test data and really-collected test samples. In section 5.1, we present the performed simulations for both LOS and NLOS conditions, and in section 5.2, we perform the experiment on the really-collected test samples. Using the noise-free dataset in the training phase has three advantages, especially for large areas: a) it removes the data collection stage that is labor-intensive. b) the trained algorithm can

easily be replaced according to the environment changes such as APs' locations displacement. c) the shadowing noise is mitigated from training data.

5.1. Numerical results

We generate RSS values from 25 APs on 500 RPs in a $100m \times 100m$ area. The relationship between RSS value and distance is as follows [37]

$$P_r = P_0 - 10\eta \log_{10}(d/d_0) - \sum_{i=1}^{W} WAF_i + \mathcal{X}(\sigma_s), \tag{26}$$

where P_r is the received power at distance d from the transmitter, P_0 is the received signal power at reference distance d_0, the parameter η is chosen based on the environment, $\mathcal{X}(\sigma_s)$ is zero-mean Gaussian noise with σ_s deviant that models the shadowing noise, and WAF_i is the wall attenuation factor of the i^{th} wall, and W is the number of walls between the transmitter and receiver. We set the parameters with proposed values in [38] that have been summarized in Table 2.

Table 2: Simulation parameters in the noise-free scenario.

System parameters	Value
Path-loss parameters [38]	$d_0 = 5m$ $P_0 = -49\,\text{dBm}$ $\eta = \begin{cases} 5 \text{ if d} \leq 5\text{m} \\ 4 \text{ if d} > 5\text{m} \end{cases}$
Wall attenuation factor	$WAF = 10$ dBm
Receiver sensitivity	-95 dBm

First, we implement our proposed EGPR method in a noise-free scenario under a LOS condition. The shadowing noise $\mathcal{X}(\sigma_s)$ is only added to the test data; because, the shadowing noise is a random factor that affects the RSS training samples. The simulated environment is depicted in Fig. 3.a for the LOS condition and 20 test points are randomly generated in this figure. Also, 20 other points as validation data are generated to tune the Noise kernel parameter. Here we compare the proposed EGPR with NaGPR [22] and CGPR [22]. The mean RMSE (MRMSE) performance of the mentioned methods is depicted in

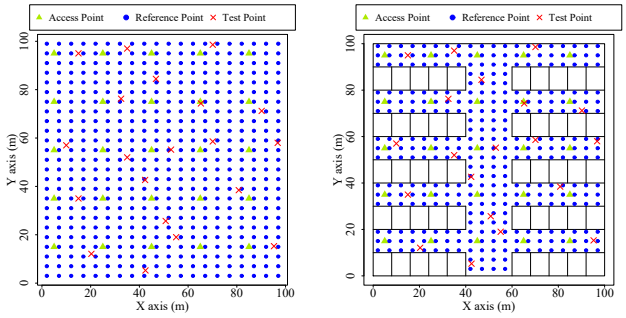


Figure 3: The simulated environment in a noise-free scenario under (a) LOS and (b) NLOS conditions.

Fig. 4.a which is defined as follows

$$\text{MRMSE} = \frac{1}{Z}\sum_{i=1}^{Z}\text{RMSE}_i, \tag{27}$$

where RMSE_i is the RMSE for the i^{th} iteration in which the noise is added to the test samples (in this experiment $Z = 200$) and prediction is performed. As can be seen, the EGPR has better performance and is very close to the CRLB. The CGPR does not show a realistic 2σ confidence interval. The TERP of this LOS scenario for x dimension has been depicted in Fig. 5.a. The horizontal line shows the 95% confidence interval. As shown, both EGPR and NaGPR provide a realistic confidence interval for each level of noise deviant, whereas the CGPR fails to estimate the confidence interval.

For the NLOS condition, the WAF is added to both of the test and train samples, because this value is not a random factor for each point, while the shadowing noise is only added to the test samples due to the randomness properties of this term. Each segment wall in simulation is described by two points and the NLOS condition can be detected by considering that the segment line between transmitter and receiver is intersected with the segment wall or not [1].

[1] Please refer to Appendix A for more discussion.

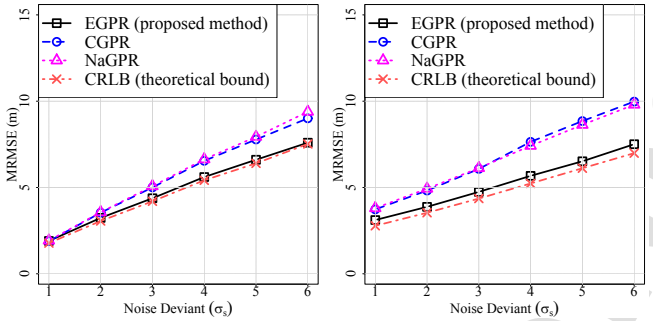


Figure 4: MRMSE performance of EGPR (proposed algorithm), CGPR [22], NaGPR [22], and theoretical bounds under (a) LOS and (b) NLOS conditions. The Crame-Rao lower bound (CRLB) shows the lowest possible RMSE in theoretical analysis.

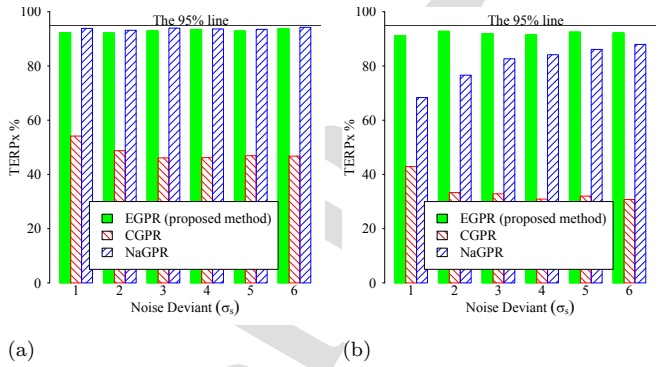


Figure 5: TERP performance of EGPR (proposed algorithm), CGPR [22], and NaGPR [22] in x dimension under (a) LOS and (b) NLOS conditions. The horizontal line shows the 95% confidence interval accuracy.

The simulated environment is depicted in Fig. 3.b. The MRSME performances of the CGPR [22], NaGPR [22], and EGPR have been plotted in Fig. 4.b. As can be seen, the proposed EGPR outperforms the MRMSE compared with CGPR and NaGPR. The TERP performance of the NaGPR under the NLOS condition in low shadowing noise is small as depicted in Fig. 5.b, while the proposed EGPR can estimate the confidence interval accurately in each level of noise deviant.

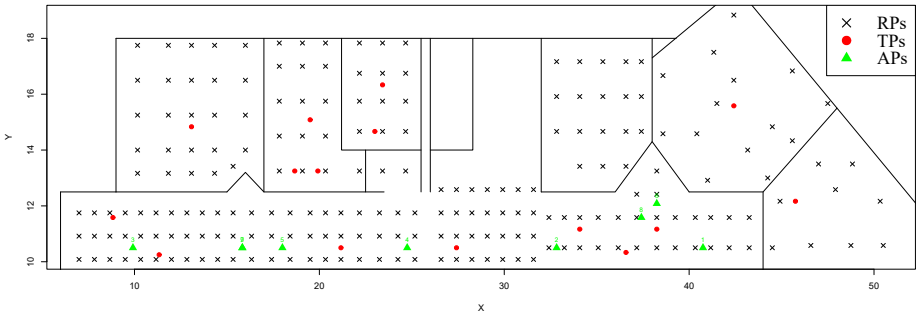


Figure 6: The Schematic of the 2nd floor of Cyberspace Research Institute at Shahid Beheshti University, where the experiment is conducted.

Table 3: Performances of algorithms in terms of RMSE and TERP in the real-world environment depicted in Fig. 6.

	EGPR	CGPR	NaGPR
RMSE	2.72	4.65	3.93
TERP	93.33%	66.66%	86.66%

5.2. Experimental Results

In this section, we use synthetic noise-free samples in the training phase and really-collected data for the test phase. The test data has been collected from the 2nd floor of Cyberspace Research Institute at Shahid Beheshti University, as depicted in Fig. 6. There are 9 APs[2] in the environment, which their locations are known. At each TP, 100 RSS samples have been recorded, and the average of these samples is used for the test. Also, we set -95 dBm for the RSS values of unavailable APs at TPs. Each wall is described by two points, and the NLOS condition is considered in the offline phase of localization. The RSS samples at RPs are generated by the path-loss model. However, the test data (red dots) is collected with a smartphone. Table 3 shows the performances of algorithms in terms of RMSE and TERP. As can be seen, the proposed method achieves better results in terms of RMSE and TERP. It shows the robustness of the pro-

[2]Note that two APs (numbered as 7 and 9) are at the same locations in a 2D environment; however, their floor are different.

Table 4: Comparison of probabilistic-based algorithms. The top and bottom arrows show a high and low level, respectively. Also, the green and red arrows demonstrate a desired and undesired state, respectively.

	Complexity		Validation Data	Noise Prior Knowledge	Confidence Interval Accuracy			
					LOS		NLOS	
	Online	Offline			LS	HS	LS	HS
CGPR	↓	↓	No	No	↓	↓	↓	↓
NaGPR	↑	↓	No	Yes	↑	↑	↓	↑
Proposed EGPR	↓	↑	Yes	No	↑	↑	↑	↑

posed EGPR compared with counterparts in the real-world environment.

In Table 4, we have summarized characteristics of probabilistic-based GPR algorithms in terms of online/offline complexity, the need to validation data, prior knowledge about shadowing noise in the test data, and accuracy of confidence interval estimation under LOS/NLOS condition in low/high shadowing noise (LS/HS). The complexity level of calculations in CGPR is low for both online and offline phases; however, it cannot truly estimate the 2σ confidence interval. The NaGPR implicates the complexity in the online phase with prior knowledge about the shadowing noise of the test data, while it does not need validation data. The proposed EGPR implicates the complexity in the offline phase to estimate the noise of the data, while there is no need for prior knowledge about shadowing noise.

6. Conclusion

We proposed a GPR-based algorithm named enhanced GPR (EGPR), which can estimate the 2σ confidence interval with higher accuracy particularly compared with other methods for synthetic noise-free fingerprints, while it does not increase the computational complexity in the online phase. Experimental results showed that when noise-free data is used in the training phase, the conventional GPR (CGPR) method cannot estimate the 2σ confidence interval. The NaGPR can calculate the confidence interval via high accuracy with prior knowledge about shadowing noise; however, it increases the complexity in the online phase that is not acceptable for real-world scenarios. We also theoretically analyzed the lower bound of errors in terms of CRLB and proved that the GPR-based

algorithms are MVU and efficient estimators, which means that they can reach the CRLB. However, to estimate a reliable CRLB, the Noise kernel parameter should be tuned carefully, since both CRLB and confidence intervals are directly affected by Noise kernel. The proposed tuning process is independent of the optimization process and transports the online phase complexity to the offline phase. To reduce the offline phase complexity, one can consider how to learn the noise of validation data in the optimization process, as a future work.

We have performed all simulations for the 2D environment, however, the proposed method can be extended for the 3D environment to more realistically estimate the users' locations. This would possibly enhance the estimation accuracy, because, noise-free fingerprints are generated based on a 3D distance from the APs. Since the proposed method needs a small number of validation data for tuning the noise kernel parameter, the validation data also can be utilized for better choosing the path-loss parameters.

Appendix A. NLOS condition

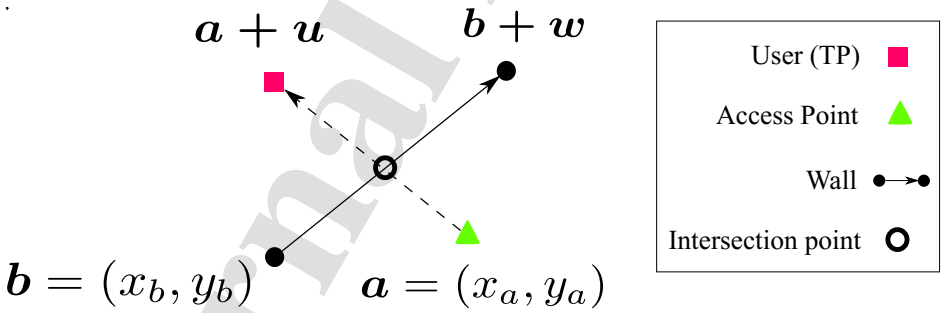


Figure A.7: The locations of user, access point, and wall for detecting NLOS condition.

As depicted in Fig. A.7, the NLOS condition can be detected when the segment line between a pair of AP and user is intersected by a walls. The user is under the NLOS condition if we can find t and q such that

$$\boldsymbol{a}+t\boldsymbol{u}=\boldsymbol{b}+q\boldsymbol{w}, \tag{A.1}$$

by multiplying both sides of (A.1) to $\boldsymbol{w}$ we have

$$(\boldsymbol{a} + t\boldsymbol{u}) \times \boldsymbol{w} = (\boldsymbol{b} + q\boldsymbol{w}) \times \boldsymbol{w}, \tag{A.2}$$

where $\times$ is cross product[3] and $\boldsymbol{w} \times \boldsymbol{w} = 0$. Simplifying the above equality, we have

$$t = \frac{(\boldsymbol{b} - \boldsymbol{a}) \times \boldsymbol{w}}{\boldsymbol{u} \times \boldsymbol{w}}, \tag{A.3}$$

similarly for q, we have

$$q = \frac{(\boldsymbol{a} - \boldsymbol{b}) \times \boldsymbol{u}}{\boldsymbol{w} \times \boldsymbol{u}}, \tag{A.4}$$

There are four possible situations:

1- If $(\boldsymbol{b} - \boldsymbol{a}) \times \boldsymbol{w} = 0$ and $\boldsymbol{u} \times \boldsymbol{w} = 0$, two segments are colliner.

2- If $(\boldsymbol{b} - \boldsymbol{a}) \times \boldsymbol{w} = 0$ and $\boldsymbol{u} \times \boldsymbol{w} \neq 0$, two segments are parallel.

3- If $\boldsymbol{u} \times \boldsymbol{w} \neq 0$, $0 \leq t \leq 1$, and $0 \leq q \leq 1$, two segments are intersect at the point $\boldsymbol{a} + t\boldsymbol{u} = \boldsymbol{b} + q\boldsymbol{w}$.

4- Otherwise, the two segments are not parallel, however, they do not intersect either.

References

[1] F. Zafari, A. Gkelias, K. K. Leung, A survey of indoor localization systems and technologies, IEEE Communications Surveys & Tutorials 21 (3) (2019) 2568–2599.

[2] P. M. Ghari, R. Shahbazian, S. A. Ghorashi, Maximum entropy-based semi-definite programming for wireless sensor network localization, IEEE Internet of Things Journal 6 (2) (2019) 3480–3491.

[3] A. Khalajmehrabadi, N. Gatsis, D. Akopian, Modern WLAN fingerprinting indoor positioning methods and deployment challenges, IEEE Communications Surveys & Tutorials 19 (3) (2017) 1974–2002.

[3] If $\boldsymbol{a} = (x_a, y_a)$ and therefore $\boldsymbol{b} = (x_b, y_b)$, the cross product for these two vector equals to $\boldsymbol{a} \times \boldsymbol{b} = x_a y_b - y_a x_b$

[4] M. Nabati, S. A. Ghorashi, R. Shahbazian, Joint coordinate optimization in fingerprint-based indoor positioning, IEEE Communications Letters 25 (4) (2021) 1192–1195.

[5] E. Xu, Z. Ding, S. Dasgupta, Source localization in wireless sensor networks from signal time-of-arrival measurements, IEEE Transactions on Signal Processing 59 (6) (2011) 2887–2897.

[6] J. J. Pérez-Solano, S. Ezpeleta, J. M. Claver, Indoor localization using time difference of arrival with uwb signals and unsynchronized devices, Ad Hoc Networks 99 (2020) 102067.

[7] C. Yang, H. Shao, WiFi-based indoor positioning, IEEE Communications Magazine 53 (3) (2015) 150–157.

[8] L. Chen, I. Ahriz, D. Le Ruyet, AoA-aware probabilistic indoor location fingerprinting using channel state information, IEEE Internet of Things Journal (2020) 1–1.

[9] E. Homayounvala, M. Nabati, R. Shahbazian, S. A. Ghorashi, V. Moghtadaiee, A novel smartphone application for indoor positioning of users based on machine learning, in: Adjunct Proceedings of the ACM International Joint Conference on UbiComp/ISW, 2019, p. 430–437.

[10] W. Sun, M. Xue, H. Yu, H. Tang, A. Lin, Augmentation of fingerprints for indoor WiFi localization based on gaussian process regression, IEEE Transactions on Vehicular Technology 67 (11) (2018) 10896–10905.

[11] J. Jun, L. He, Y. Gu, W. Jiang, G. Kushwaha, V. A., L. Cheng, C. Liu, T. Zhu, Ieee transactions on mobile computing 17 (3) (2018) 590–603.

[12] S. Tomažič, D. Dovžan, I. Škrjanc, Ieee transactions on industrial electronics 66 (3) (2019) 2015–2024.

[13] K. Zheng, H. Wang, H. Li, W. Xiang, L. Lei, J. Qiao, X. S. Shen, Energy-efficient localization and tracking of mobile devices in wireless sensor net-

works, IEEE Transactions on Vehicular Technology 66 (3) (2017) 2714–2726.

[14] X. Guo, S. Shao, N. Ansari, A. Khreishah, Indoor localization using visible light via fusion of multiple classifiers, IEEE photonics journal 9 (6) (2017) 1–16.

[15] X. Tian, W. Li, Y. Yang, Z. Zhang, X. Wang, Optimization of fingerprints reporting strategy for WLAN indoor localization, IEEE Transactions on Mobile Computing 17 (2) (2018) 390–403.

[16] A. Ravi, A. Misra, Practical server-side wifi-based indoor localization: Addressing cardinality & outlier challenges for improved occupancy estimation, Ad Hoc Networks 115 (2021) 102443.

[17] M. Zhao, D. Qin, R. Guo, G. Xu, Research on crowdsourcing network indoor localization based on co-forest and bayesian compressed sensing, Ad Hoc Networks 105 (2020) 102176.

[18] Y. Wang, C. Xiu, X. Zhang, D. Yang, WiFi indoor localization with CSI fingerprinting-based random forest, Sensors 18 (9) (2018) 2869.

[19] X. Wang, L. Gao, S. Mao, S. Pandey, CSI-based fingerprinting for indoor localization: A deep learning approach, IEEE Transactions on Vehicular Technology 66 (1) (2017) 763–776.

[20] X. Tian, S. Zhu, S. Xiong, B. Jiang, Y. Yang, X. Wang, Performance analysis of Wi-Fi indoor localization with channel state information, IEEE Transactions on Mobile Computing 18 (8) (2019) 1870–1884.

[21] W. Kim, J. Park, J. Yoo, H. J. Kim, C. G. Park, Target localization using ensemble support vector regression in wireless sensor networks, IEEE transactions on cybernetics 43 (4) (2013) 1189–1198.

[22] K. N. R. S. V. Prasad, E. Hossain, V. K. Bhargava, Machine learning methods for RSS-based user positioning in distributed massive MIMO, IEEE Transactions on Wireless Communications 17 (12) (2018) 8402–8417.

[23] P. Mirowski, H. Steck, P. Whiting, R. Palaniappan, M. MacDonald, T. K. Ho, Kl-divergence kernel regression for non-gaussian fingerprint based localization, in: International Conference on Indoor Positioning and Indoor Navigation, 2011, pp. 1–10.

[24] X. Guo, L. Li, F. Xu, N. Ansari, Expectation maximization indoor localization utilizing supporting set for internet of things, IEEE Internet of Things Journal 6 (2) (2019) 2573–2582.

[25] S. He, S. . G. Chan, Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons, IEEE Communications Surveys & Tutorials 18 (1) (2016) 466–490.

[26] M. Nabati, H. Navidan, R. Shahbazian, S. A. Ghorashi, D. Windridge, Using synthetic data to enhance the accuracy of fingerprint-based localization: A deep learning approach, IEEE Sensors Letters 4 (4) (2020) 1–4.

[27] Q. Jiang, Y. Ma, K. Liu, Z. Dou, A probabilistic radio map construction scheme for crowdsourcing-based fingerprinting localization, IEEE Sensors Journal 16 (10) (2016) 3764–3774.

[28] W. Cui, L. Zhang, B. Li, J. Guo, W. Meng, H. Wang, L. Xie, Received signal strength based indoor positioning using a random vector functional link network, IEEE Transactions on Industrial Informatics 14 (5) (2018) 1846–1855.

[29] J. Lim, W. Jang, G. Yoon, D. Han, Radio map update automation for WiFi positioning systems, IEEE Communications Letters 17 (4) (2013) 693–696.

[30] A. Zanella, Best practice in RSS measurements and ranging, IEEE Communications Surveys & Tutorials 18 (4) (2016) 2662–2686.

[31] K. N. R. S. V. Prasad, E. Hossain, V. K. Bhargava, S. Mallick, Analytical approximation-based machine learning methods for user positioning in distributed massive MIMO, IEEE Access 6 (2018) 18431–18452.

[32] C. K. Williams, C. E. Rasmussen, Gaussian processes for machine learning, Vol. 2, MIT press Cambridge, MA, 2006.

[33] J. Nocedal, S. Wright, Numerical optimization, Springer Science & Business Media, 2006.

[34] S. Boyd, S. P. Boyd, L. Vandenberghe, Convex optimization, Cambridge university press, 2004.

[35] S. M. Kay, Fundamentals of statistical signal processing, Prentice Hall PTR, 1993.

[36] S. Kumar, R. M. Hegde, N. Trigoni, Gaussian process regression for fingerprinting based localization, Ad Hoc Networks 51 (2016) 1–10.

[37] M. Lott, I. Forkel, A multi-wall-and-floor model for indoor radio propagation, in: IEEE VTS 53rd Vehicular Technology Conference, Spring 2001. Proceedings (Cat. No. 01CH37202), Vol. 1, IEEE, 2001, pp. 464–468.

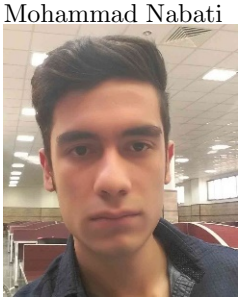
[38] Atreyi Bose, Chuan Heng Foh, A practical path loss model for indoor WiFi positioning enhancement, in: International Conference on Information, Communications Signal Processing, 2007, pp. 1–5.

Mohammad Nabati received his M.Sc. degree in Electrical Engineering from the Shahid Beheshti University. He also has spent three years 2017-2020 as a research assistant at the cognitive radio laboratory in the Shahid Beheshti University. His main research interests include applications of machine learning and deep learning algorithms in communication systems, internet of things, and indoor localization.

Seyed Ali Ghorashi received his B.Sc. and M.Sc. degrees in Electrical Engineering from the University of Tehran, Iran and his Ph.D. degree from Kings College London, UK. He has worked for Samsung Electronics (UK) Ltd, Shahid Beheshti University and University of East London. He is a senior member of IEEE, holds international patents and has published over 120 technical papers mainly related to the applications of optimization, artificial intelligence and machine learning in localization, internet of things and wireless communications.

Reza Shahbazian has received his B.Sc. and M.Sc. degree in Telecommunications from the Iran University of Science and Technology (IUST), Tehran, Iran, in 2008 and 2011, respectively. He has received a Ph.D. in Telecommunications Engineering from Shahid Beheshti University in 2017 and a Ph.D. in computer science from the University of Calabria, Italy, in 2020. He has served as a postdoc researcher at Shahid Beheshti University and University of Calabria in machine learning applications. He has been the author and co-author of 5 books, several journals, and conference papers. He has served as executive chair of two international conferences and the reviewer for many credible journals.

Mohammad Nabati



Seyed Ali Ghorashi

Reza Shahbazian

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: