EURASIP Journal on Wireless Communications and Networking

RESEARCH

Open Access

Improving the performance of a radio-frequency localization system in adverse outdoor applications

Marcelo N. de Sousa[1*], Ricardo Sant'Ana[2], Rigel P. Fernandes[2], Julio Cesar Duarte[2], José A. Apolinário Jr.[2] and Reiner S. Thomä[1]

*Correspondence: marcelo.nogueira@tu-ilmenau.de
[1] Electronic Measurements and Signal Processing, University of Technology of Ilmenau, Helmholtzplatz 2, 98693 Ilmenau, Germany
Full list of author information is available at the end of the article

Abstract

In outdoor RF localization systems, particularly where line of sight can not be guaranteed or where multipath effects are severe, information about the terrain may improve the position estimate's performance. Given the difficulties in obtaining real data, a ray-tracing fingerprint is a viable option. Nevertheless, although presenting good simulation results, the performance of systems trained with simulated features only suffer degradation when employed to process real-life data. This work intends to improve the localization accuracy when using ray-tracing fingerprints and a few field data obtained from an adverse environment where a large number of measurements is not an option. We employ a machine learning (ML) algorithm to explore the multipath information. We selected algorithms random forest and gradient boosting; both considered efficient tools in the literature. In a strict simulation scenario (simulated data for training, validating, and testing), we obtained the same good results found in the literature (error around 2 m). In a real-world system (simulated data for training, real data for validating and testing), both ML algorithms resulted in a mean positioning error around 100 ,m. We have also obtained experimental results for noisy (artificially added Gaussian noise) and mismatched (with a null subset of) features. From the simulations carried out in this work, our study revealed that enhancing the ML model with a few real-world data improves localization's overall performance. From the machine ML algorithms employed herein, we also observed that, under noisy conditions, the random forest algorithm achieved a slightly better result than the gradient boosting algorithm. However, they achieved similar results in a mismatch experiment. This work's practical implication is that multipath information, once rejected in old localization techniques, now represents a significant source of information whenever we have prior knowledge to train the ML algorithm.

Keywords: RF localization, Wireless positioning, Hybrid positioning, Machine learning, Ray tracing fingerprints

1 Introduction

The localization problem in communication systems has been on research spot for many years. Node position estimation enables a large set of location-based services [1], boosts biological studies [2], and improves defense capabilities [3].

In a multipath scenario, the primary signal of an emitter and/or its reflections arrives at the sensors, resulting in a different number of possible targets. The non-line-of-sight (NLOS) between the emitter and the sensors affects the position estimation in an outdoor scenario [4, 5], because only reflections can be used to estimate the emitter localization.

However, it is still possible to use these signals for transmitter localization and tracking, if multipath exploitation is performed and information about the environment is available [6]. In [7], multipath information is used with the image theory to locate the emitter using only one sensor. The hybrid approach presented in [8], uses multipath information, machine learning, and propagation simulation tools to enhance the performance of outdoor TDOA systems [9]. The work presented in [10], uses evolutionary algorithms to optimize target localization in a wireless sensor network. A two-stage position estimation system is presented in [11]: the first step is to ignore the additional path error and estimate an initial position; the second step is to include the additional path error in the estimation problem to enhance the results using a variable projection method. The research presented in [12], describes a method to estimate the node location using deep learning in outdoor environments, more specifically in a heterogeneous network environment.

In this work, we consider an area of 788 m $\times$ 736 m $=$ 579,968 m^2. In this area, we need to locate a target (for example, a cell phone from hostile target—a non-collaborative emitter). Since we assume an adverse (sometimes hostile) area, it is not possible to obtain real data for the location with NLOS. However, it is possible to obtain real data obtained by a reconnaissance patrol in a known position. It is also possible to obtain simulated data based on ray-tracing for the region of interest. This scenario will define three datasets: Simulation Dataset which contains simulated data from the ray-tracing simulation, Emitter Dataset, which contains real data from a reconnaissance patrol in known positions and Target Dataset, which contains real data from a target. We will refer to this scenario as outdoor adverse position localization. To our knowledge, none of related bibliography addressed this type of scenario.

Many machine learning methods can be used for node localization; however, it should be noted that random forest and gradient boosting are ensemble methods having several advantages that make them suitable for outdoor fingerprinting localization:

- Random Forest uses decision trees, which are very prone to success [13]; typically, it is used to achieve higher accuracy, based on different sets of attributes and samples. Random forest is known for fast training, matching speed, high classification accuracy and good performance for high-dimensional input data [14, 15]. An approach of position fingerprints using Random Forest can be found in [16]. An algorithm and a synthetic volume cross-correlation (VCC) function to extract the multipath features from the TDOAs measurements are available in [17].

- Gradient Boosting is a boosting method built on weak classifiers. It was applied to indoor localization and performed equally or better than k-nearest neighbors (kNN) in [18]. It is based on weak learners (high bias, low variance). The base predictors or weak learners are shallow trees, sometimes even as small as decision stumps (trees with only one level of decision). The boosting approaches reduce error mainly by reducing bias and by aggregating the output from many models.

Finally, both algorithms in [19, 20] can handle the discrepancies in data quality, which cause over-fitting and under-representative data observations. However, a radio-map is also required to allow multipath exploitation; yet, it is not always possible to build the dataset with the real-world signals, and a simulation tool can be used. Therefore, some mismatches may occur between the dataset produced with simulation tools and real-world measurements provided by sensors. Also, the machine learning engine can produce a few outliers in the position estimation, and the tuning parameters cannot deal with intense noise in the real world measurements setup.

For this reason, it is essential to improve the estimations even when we have to deal with the differences caused by noise and mismatching between synthetic and real-world data in an outdoor hostile position localization.

The contributions of this work are listed in the following.

1 Enhancement of the machine learning model with few real world data, a common requirement for law-enforcement agencies and defense forces;
2 The use of synthetic data and limited amount of real world data to train the model and to predict real world data;
3 Performance of Gradient Boosting and Random Forest models for mismatching and noisy measurements of multipath fingerprints in outdoor hostile position localization; and
4 Guidelines to optimize the hyperparameters of machine learning algorithms applied to estimate emitter position in outdoor hostile scenario, using mainly simulation data.

The rest of the paper is organized as follows: Sect. 2 shows the methods in localization systems using multipath exploitation and machine learning. Section 3 presents the main approaches when using localization methods and the effect of the NLOS between emitter and sensors, and explains how the ray-tracing simulation tool is used to produce the dataset employed by the machine learning algorithms. Section 4 presents the main aspects of a multipath in localization systems. Section 5 presents how to apply machine learning in localization problems and describes the main aspects of the machine learning fingerprint framework, explaining the general characteristics of methods random forest and gradient boosting. In the following, Sect. 6.1 details the proposed scenario, presents and analyses the results obtained for both random forest and gradient boosting. This section also presents the results from two experiments to evaluate how the machine learning algorithms behave under noisy environment and mismatch measurements. Finally, Sect. 7 summarizes conclusions and perspectives for future work.

2 Machine learning fingerprints in localization problems

This section aims to explain the method to include multipath information in an outdoor localization system, using simulation data and signal processing of real measurements. The ray-tracing electromagnetic simulation tool can extract the channel impulse response (CIR) of the signal that arrives in each TDOA sensor [21]. We used the AWE WinProp ray-tracing to extract CIR fingerprints. We created a fingerprint using the same design employed in [22] and used machine learning algorithms to predict the target position. The dataset was created with amplitudes and delays of wavefront components arriving on the sensors. We used two other Datasets from real-world data: (1) an Emitter dataset corresponding to 1000 measurements using four sensors with emitter from a fixed know position, and (2) a target dataset corresponding to measurements using four sensors with emitter in different positions [22].

In Algorithm 1, we have an overview of the machine learning training process: the dataset generated by synthetic volume cross-correlation (VCC) produced by ray-tracing (simulation dataset) was the only data used for training the model. In contrast, the real-world data (emitter dataset) was used for the validation step (to set hyperparameters). When hyperparameters are selected, both datasets are used to generate the final model.

```
Algorithm 1: Machine Learning training
input  : Features from each sensor $S_1$, $S_2$, $S_3$ and $S_4$
output: Model for predicting $(X_e, Y_e)$
1) From the Emitter Dataset:
· $\alpha_i, \Delta_{\tau,i}$ for each of the 4 sensors;
· Emitter position vector $\mathbf{x}$ ;
2) From the Simulation Dataset:
· $\alpha_i, \Delta_{\tau,i}$ for each of the 4 sensors;
· Position vector $\mathbf{x}$ for each point from 10 to 10 meters;
· Buildings descriptions;
3) Machine Learning training process:
· Select model's hyperparameters;
· Define loss function as RMSE;
· Use data from Simulation Dataset;
· Use data from emitter Dataset for validation;
4) Build a Regression Model:
· 20 $(\alpha_i, \Delta_{\tau,i})$ for each sensor;
· 2 Outputs: $[X_s, Y_s]^{\mathsf{T}}$;
5) Tunning Hyperparameters:
· Use data from Emiter dataset to evaluate performance;
· 20 $(\alpha_i, \Delta_{\tau,i})$ for each sensor;
· Use data from Simulation dataset to evaluate performance;
· 20 $(\alpha_i, \Delta_{\tau,i})$ for each sensor;
· Evaluate performance. Adjust hyperparameters;
Repeat Step 3 to 5 until performance is considered
  suitable;
Obtain Final Model;
· Use both Simulation and Emitter Datasets and
  hyperparameters selected to generate a model.;
```

After generating a model using random forest and gradient boosting, we used the target dataset to predict locations ($\mathbf{p}_t = [x_t, y_t]^T$) and compared the performance of each model. Two simulations were performed: one evaluating the model response to the addition of Gaussian noise to the features of the target dataset; and another one estimating the models' response when cancelling features from the target dataset.

Figure 1 presents the suggested framework. First, the machine learning engine uses the Simulation Dataset, optimizes the hyperparameters using the Emitter Dataset, and creates the prediction model. Later, both machine learning models (gradient boosting or random forest) are used to perform position prediction. In parallel, we add noise and create an artificial mismatch on the features of the target dataset. Meanwhile, we evaluate the prediction error using the Euclidean distance between the predicted and the actual positions.

3 Using NLOS in localization systems

This section shows the existing approaches to deal with NLOS measurements in outdoor localization systems; most do not use multipath corrupted position estimations. Others only include multipath fingerprints using real measurements data.

Traditional TDOA techniques are strongly affected by reflected and diffracted rays in the environment; when there are measurements with NLOS paths, the location errors can be substantial. The performance of the localization systems depends on the signal processing algorithms and the channel characteristics severely affected by NLOS rays [23].

There are several approaches to deal with multipath in localization systems; the classical TDOA localization approaches have relied only on line-of-sight propagation with degradation in performance whenever NLOS rays occurred. Regarding antenna simplicity, small size, weight and power requirements (SWaP) [24], TOA and TDOA techniques are the most popular schemes used for emitter localization in wireless networks.

NLOS multipath propagation introduces an inherent error into the localization because it alters the propagation paths and adds additional channel delays. It is possible to obtain the propagation information to deal with NLOS using "radio frequency fingerprinting," either performing extensive measurement campaigns or using ray-tracing to simulate the environment.

Ray-tracing models the signal propagation finding the ray propagation trajectories in a defined scenario, with geometric optics approximation.

In [25], a multipath database characterization was created with a grid of possible emitter positions, where the angle of arrival (azimuth and elevation) and the time of arrival were recorded, giving a signature for each possible transmitter location in an area of interest that is populated via ray-tracing simulations. The received signal with NLOS components is compared with the values in this database, to estimate the emitter position with the same multipath information.

The approach described in [26] uses multipath characteristics of the scenario to build a database of NLOS rays, applying a clustering procedure to match the real-world measurement with the simulated one to locate the emitter. In [27], the authors presented different localization fingerprints using algorithms received signal strength (RSS) and the K-nearest neighbour (KNN). There are different types of "Fingerprints," as discussed in [28] and [29], where a performance improvement was observed when the position estimation was evaluated using the channel state information (CSI) for long term evolution (LTE). In [30], several approaches for indoor and outdoor multipath fingerprint enhancement localization systems in 5G and IoT are presented. A modified version of the random forest algorithm [22] is used to implement a localization system with the information of Wi-Fi access points in indoor scenarios; the system gives the target position as a classification processing.

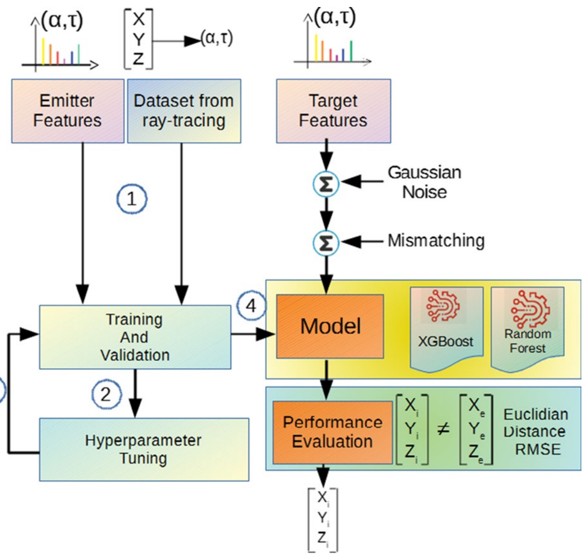


Fig. 1 Overview of the proposed approach

4 Ray tracing to exploit multipath in localization systems

The machine learning fingerprint framework presented in [16] used a ray-tracing simulation tool to extract multipath features for all possible emitter positions in the scenario. Sometimes, due to operational or physical restrictions, it is challenging to build a dataset with real world measurements; thus some studies, like those in [8], use simulation tools. In [21], the term "ray tracing" fingerprint relates to a "radio map" of received signal strength (RSS) in a coverage prediction, that takes into account the output power of the emitter to perform the position estimation [31].

Figure 2 shows a typical performance of a TDoA localization system in outdoor scenario, as described in [16]. We can see how the estimations differ from the actual emitter position, multipath and NLOS effect degrading the system performance severely.

Figure 3 shows that the multipath fingerprints also give information about the propagation mechanism (reflection, diffraction or scattering). The simulation output draws each path from each point in a scenario grid, which describes amplitude, delay (α_i, τ_i), reflection points and angular information of each ray. They represent the emitter-sensor interactions, and it is the basis for the visibility matrix where each interaction is considered as a layer in a multilayer scheme. Therefore, the ray tracing (RT) software gives information about each ray path, describing the edges and walls touched by the rays in the emitter-sensor path.

The performance is highly dependable on the details given in the scenario setup. In practical outdoor implementations, buildings are only represented by simple structures, where details such as windows and doors are not present. For a suburban outdoor scenario with simple buildings, the ray-tracing does give reasonable information about the main specular components in the propagation channel.

A ray tracing simulation provides the site-specific channel impulse response, which means that, as soon the position of each sensor is defined, it is possible to obtain the multipath information of each point of the scenario.

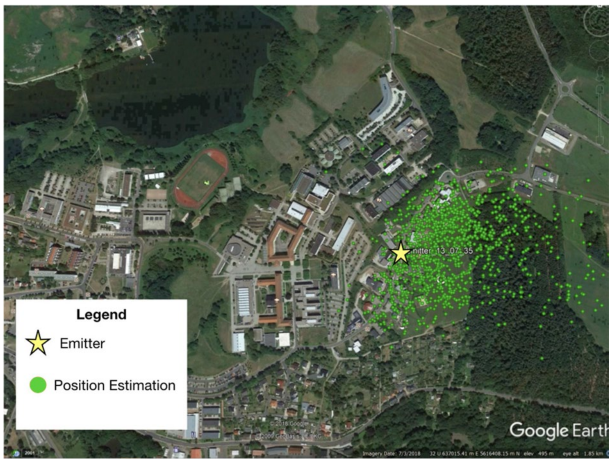


Fig. 2 Performance of a Localization System in Outdoor Scenario [16]

The specular components path including all reflection and diffraction points are available at the end of the ray-tracing simulation. Depending on the desired number of ray interactions: the image theory allows to identify the reflection points, and the virtual nodes, all possible rays in the simulation domain can be estimated.

Following the approach of [32], the ray-tracing scenario can be decomposed into walls and edges; the "view tree" represents the emitter-sensor interactions, and it is the basis for the visibility matrix, where each interaction is considered a layer in a multilayer scheme as depicted in Fig. 4.

With the output file of the ray tracing software, it is possible to search for a given position where the main reflectors are those the rays bounce before arriving at the sensor.

The inputs of the localization problem are the positions of the sensors, usually known, the received signal and the scenario description or characterization. With this information, it is possible to improve the performance of the localization system by adding a multipath fingerprint using NLOS patterns.

In this point, the approximation, not only in the scenario description but also in the RT multipath information, can play an essential role in the machine learning framework. For this reason, the rays description should be good enough to establish the model, but can not be as precise that loses the generalization features. Generalization occurs when we try to describe the learning of the target function from simulation training data.

The data model and the position estimation problem is a generalization of the approach introduced by [33]:

$$\mathbf{r} = \mathbf{f}(\mathbf{x}) + \mathbf{n}, \tag{1}$$

where $\mathbf{r}$ is the measurements vector, $\mathbf{x}$ is the vector with the unknown source position that we want to estimate, $\mathbf{f}(x)$ is a non-linear function that maps the position vector into the measurements, and $\mathbf{n}$ is a vector that describes the zero mean noise that corrupts the measurements. TDOA location is carried out using the range differences assuming that the received signals are synchronized.

When the source emits a signal at instant t_0 (unknown), the lth sensor receives the signal at time t_l, $l = 1, 2, \ldots L$. It is possible to obtain $L(L-1)/2$ distinct TDOAs. If there

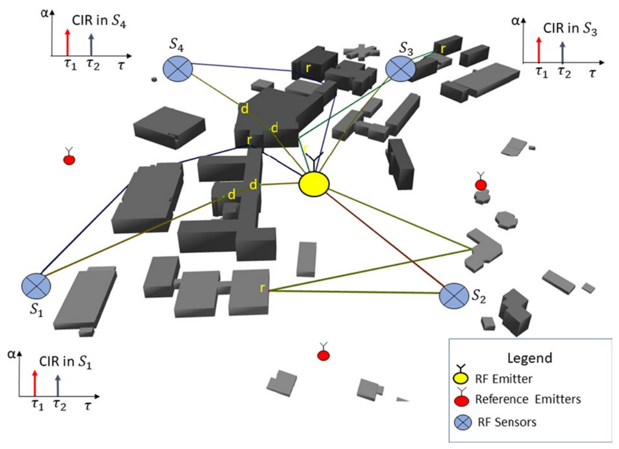


Fig. 3 Extraction of CIR fingerprints using ray tracing

are four sensors, we would have six delays from which only three are usually employed: τ_{21}, τ_{31}, and τ_{41}. The time differences and range differences are related by a constant, the speed of light. Using the range difference formulation with the TDOA values, we have:

$$r_{\mathrm{TDOA}} = d_{l,1} + n_{\mathrm{TDOA}}, \quad l = 2, 3, \ldots, L, \tag{2}$$

where the term $d_{l,1} = d_l - d_1$, and n_{TDOA} is the error in the range differences. It is possible to use the following compact matrix notation.

$$\mathbf{r}_{\mathrm{TDOA}} = \left[r_{\mathrm{TDOA},2}, r_{\mathrm{TDOA},3} \ldots r_{\mathrm{TDOA},L}\right]^T$$
$$\mathbf{n}_{\mathrm{TDOA}} = \left[n_{\mathrm{TDOA},2}, n_{\mathrm{TDOA},3} \ldots n_{\mathrm{TDOA},L}\right]^T$$

$$\mathbf{f}_{\mathrm{TDOA}}(X) = \begin{bmatrix} \sqrt{(x-x_2)^2 + (y-y_2)^2} - \sqrt{(x-x_1)^2 + (y-y_1)^2} \\ \sqrt{(x-x_3)^2 + (y-y_3)^2} - \sqrt{(x-x_1)^2 + (y-y_1)^2} \\ \vdots \\ \sqrt{(x-x_L)^2 + (y-y_L)^2} - \sqrt{(x-x_1)^2 + (y-y_1)^2} \end{bmatrix} \tag{3}$$

$$\begin{aligned} p(\mathbf{r}_{\mathrm{TDOA}}) =& \frac{1}{(2\pi)^{(L-1)/2}|\mathbf{C}_{\mathrm{TDOA}}|^{1/2}} \times \\ & \exp\left(-\frac{1}{2}(\mathbf{r}_{\mathrm{TDOA}} - \mathbf{d}_1)^T \mathbf{C}_{\mathrm{TDOA}}^{-1}(\mathbf{r}_{\mathrm{TDOA}} - \mathbf{d}_1)\right) \end{aligned} \tag{4}$$

The position estimation is, therefore, a process to deal with the nonlinear formulation of vector $\mathbf{f}$ using either least square (LS) or weighted least squares (WLS) formulation to evaluate the error between the estimated and the actual positions,

$$\mathbf{e}_{\mathrm{nonlinear}} = \mathbf{e}_{\mathrm{noise}} = \mathbf{r} - \mathbf{f}(\tilde{x}). \tag{5}$$

In case of multipath, time differences or range differences present an extra error caused by the NLOS signals:

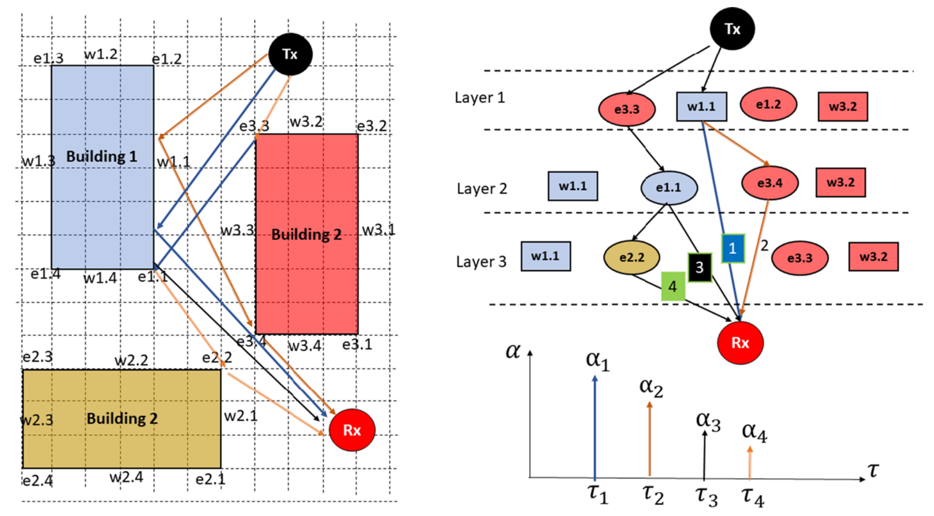


Fig. 4 Visibility tree from building walls and edges in ray tracing simulation, adapted from [32]

$$\mathbf{e}_{\text{nonlinear}} = \mathbf{e}_{\text{noise}} + \mathbf{e}_{\text{NLOS}}. \tag{6}$$

The effects of the multipath are included in the signal data model and in the Cramér–Rao bound, presented in [23], as an “extra error” in the estimation. The TDOA in an NLOS scenario is, therefore, a standard system but with an extra noise in the estimation performance that leads to an inaccurate position estimate.

5 Machine learning algorithms

Machine learning algorithms make an approximation to find a position as a regression process (*f*). This function *f* should be able to map input variables (α_i, τ_i) to an output variable $\mathbf{p}_t = [x_t, y_t]^T$, the target position of an observation,

$$\mathbf{p}_t = f(\alpha_i, \tau_i). \tag{7}$$

Assuming there is not too much fluctuation in z, we use only x and y for our simulations.

Several machine learning algorithms can be used to solve the proposed problem. We focus on meta algorithms based on ensemble methods to cover different areas of the problem, which, through a voting scheme, tend to provide better solutions. Originally developed to reduce the variance and then to improve the accuracy, ensemble methods have since been successfully used to address a variety of machine learning problems. We have selected two known machine learning algorithms based on ensemble methods: Random Forest and Gradient Boosting.

Random Forest was introduced in [34], based on an earlier work described in [35], and uses decision trees and bagging [36]. Random forest can be used for either categorical labels (classification) or continuous labels (regression).

Bootstrap aggregating, or bagging model, is a method for fitting multiple versions of a prediction model and then combining them into an aggregated prediction (ensemble model) [36]. In bagging, b bootstrap copies of the original training data are created, the regression or classification algorithm is applied to each bootstrap sample and, in the regression context, new predictions are made by averaging the predictions from individual regressors. The bagged prediction $\tilde{f}_{\text{bag}}$ is given as

$$\tilde{f}_{\text{bag}} = \tilde{f}_1(X) + \tilde{f}_2(X) + \tilde{f}_3(X) + \cdots + \tilde{f}_b(X), \tag{8}$$

where X is the data for which we want to generate a prediction and $\tilde{f}_1(X), \tilde{f}_2(X) \ldots \tilde{f}_b(X)$ are the predictions from the individual regressor. Because of the aggregation process, bagging effectively reduces the variance of an individual regressor but does not always improve upon an individual base learner. Since each base learner is completely independent of one another, we could run them in parallel. Figure 5 shows a bagging example in which random subsets of the original dataset are drawn with replacement as random subsets of the samples. Random forest's base estimators are built on subsets of both samples and features from original dataset.

According to [37], gradient boosting is a class of the machine learning methods based on the idea that a combination of simple classifiers, obtained by a weak learner, can perform better than any of the simple classifiers alone. A weak learner is a learning algorithm capable of producing classifiers with the probability of error strictly (but only slightly) lower than that of random guessing. The same idea could be extended to the regression task. Gradient boosting produces a model based on weak learners (typically decision trees) in a stage-wise fashion like other boosting methods, but it identifies the shortcomings of weak learners by using gradients in the loss function. In most cases, the decision trees used on gradient boosting is composed by one internal node (the root) which is immediately connected to the terminal nodes or leaves. These smalls decision trees are called stumps.

The main idea of boosting is to add new models to the ensemble sequentially; it approaches the bias-variance trade-off by starting with a weak learner and sequentially boosts its performance by continuing to build new trees (from weak learners), where each new tree in the sequence tries to fix up where the previous one made the biggest mistakes. Figure 6 shows this approach.

Gradient boosting may use a decision tree as a weak learner, and since each decision tree training depends on later results from the last decision tree, it is not possible to parallel the training process; it is, therefore, a sequential process.

Random Forest models mostly depend on the number of estimators, which is the number of trees that will be used to fit. We have chosen the same approach used in [16], keeping most of the hyperparameters as default values and changing the number of estimators. By default, random forest trains fully grown trees which could be done in a parallel way; the model size is therefore limited by computer available memory.

Gradient Boosting models based on decision trees also mostly depends on the number of estimators, i.e. the number of trees which will be used in the fitting process. But since this only creates weak learners, we can limit the maximum depth of the trees, avoiding to increase the maximum depth, which could makes the model more complex and more likely to overfit.

Random Forest is especially attractive when using noisy real-world data, while gradient boosting is more sensitive to overfitting if the data is noisy.

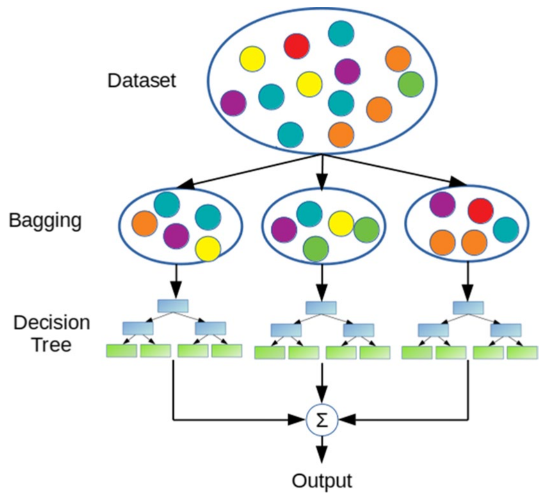


Fig. 5 Bagging example (based on [36])

6 Results and discussion

In this Section we present the results from two scenarios: a simulation scenario, which uses only simulation dataset, and a real scenario, which uses simulation and real-world datasets. Then, results from noise and mismatch experiments from the real-world scenario are presented.

6.1 Datasets

In this work, we have three different datasets available.

- The **simulation dataset**: containing 127,000 observations, each consisting of 20 pairs of amplitude and delay (α_i, τ_i) for each 4 simulated receiver, summing up 160 features and their respective localizations $\mathbf{p}$ (labels).
- The **target dataset** containing 2973 real world data observations, each consisting of 20 pairs of amplitude and delay (α_i, τ_i) for each of the 4 receivers, summing up 160 features and their respective localizations $\mathbf{p}$ (labels).
- The **emitter dataset**: containing 1000 measurements using 4 sensors with emitter from a know position, each observation consists of 20 pairs of amplitude and delay (α_i, τ_i) for each of the 4 receivers, summing up 160 features and their respective localizations, $\mathbf{p}$ (labels).

6.2 Simulation and real scenarios

We have settled two main scenarios to apply machine learning methods to disclose the results of real and simulated environment:

1 **Simulation scenario**: in this scenario, we have used just simulation dataset during training, validation and test processes for both random forests and gradient boosting models. The training dataset, which consists of 64% of the samples from simulation

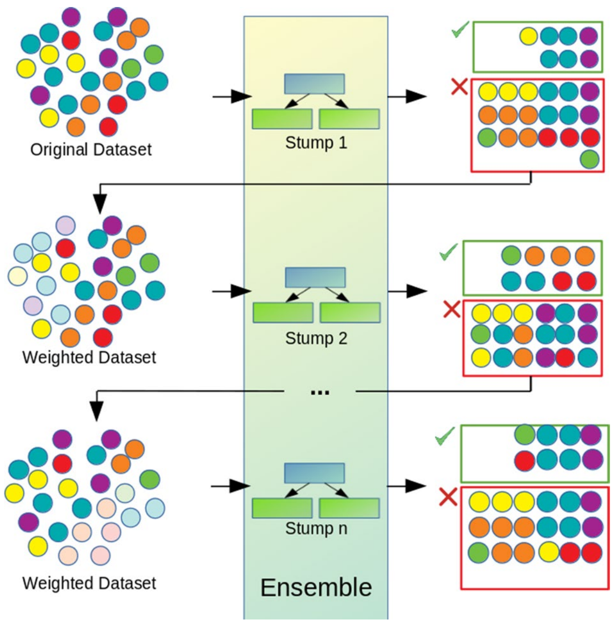


Fig. 6 Ensemble sequentially in boosting (source [38])

dataset, is used to fit the parameters. The validation dataset, which consists of 16% of the samples from simulation dataset, is used to tune the hyperparameters. Finally, the test dataset, which consists of 20% of the samples from Simulation dataset, is used to to assess the performance. The goal was to understand how the machine learning algorithms would perform on ideal conditions. Figure 7 presents the machine learning training process for the simulation scenario;

2 **Real-world scenario**: in this scenario, we have used simulation dataset and emitter dataset during training and validation processes to optimize the machine learning hyperparameters and real world data from target dataset to evaluate the model's performance. The main goal was to estimate the performance of both random forest and gradient boosts under real world environment conditions, where we could use Ray Tracing fingerprint (simulation dataset) to generate most of the data and a small amount of real world data from a fixed located source (emitter dataset) for training and validation processes. Finally, we used real world data (target dataset) for performance evaluation. Figure 8 presents the machine learning training process for the real scenario. Using the real scenario, we can evaluate how both random forest and gradient boosting would behave under **noisy environment** and **mismatch measurements** where a subset of features is nullified.

The loss function used to measure the model performance was the mean squared error (MSE) between the actual and the predicted localization in both scenarios; Eq. (9) presents its definition:

$$\mathrm{MSE} = \frac{1}{n}\Sigma_{i=1}^{n}\left|\left|\mathbf{p}_i - \tilde{\mathbf{p}}_i{}^2\right|\right|, \tag{9}$$

where $\mathbf{p}_i$ is the actual localization of emitter i and $\tilde{\mathbf{p}}_i$ is the predicted localization of emitter i.

After hyperparameters tuning, the performance estimation were evaluated for both scenarios. Figure 9 presents (a) how to evaluate performance using test subset from simulation dataset in the simulation scenario and (b) how to evaluate performance using target dataset in the real-world scenario.

6.3 Results from simulation scenario

Since we want to obtain minimum mean square distance error from predicted and actual localization using simulation dataset, we have tried to range the number of base estimators for both random forest and gradient boosting[1] using the training and validation datasets.

Depicted in Fig. 10, is the relationship between the number of base estimators (from 10 to 100) and the mean distance error achieved using validation dataset and random forest model. We have chosen the number of estimators as 80. This configuration does not yield the smallest error, but the difference of error between 80 and 100 base estimators is about 0.2 m, which is very small and could be ignored. But the time required to generate a model with 80 base estimators are around 70% of the time required to generate a model with 100 base estimators. The mean distance error in the Test Dataset using 80 base estimators was 1.38 m.

Figure 11 presents how the mean distance error from predictions and true value for Validation Dataset for a range from 10 to 400 base estimators for XGBoost. We have chosen the number of estimators as 100 and a mean distance error value of 3 m, which was a trade-off between number estimators and mean distance error. When using, for instance, 400 estimators, we obtained a mean distance error value of 2.5 m, but the XGBoost model using 400 estimators will take more than 8 times the amount of time to generate a model than a XGBoost model using only 100 base estimators. The mean distance error in the Test Dataset using 80 base estimators was 2.52 m.

The results from simulation scenario shows that both random forest and gradient boosting (implemented as XGBoost) obtains suitable performance evaluated as mean distance error from true value. These results, 1.38 m for random forest and 2.52 m for gradient boosting, are consistent with results that used only simulation data as found in [25, 39–41].

So, the conclusion is that when using just simulation data, both random forest and XGBoost obtained very good results. These results could be applied when there are real data avaliable for traning, validation and test. But, as presented, when considering outdoor hostile position localization, there is no real data avaliable for the region of interest.

6.4 Results from the real scenario

In the real scenario, the simulation dataset was used for training the algorithm and the emitter dataset was used for validation (hyperparameters tuning) and later, when generating final model. For the random forest model, we used a range of base estimators

[1] We used a XGBoost implementation of Gradient Boosting.

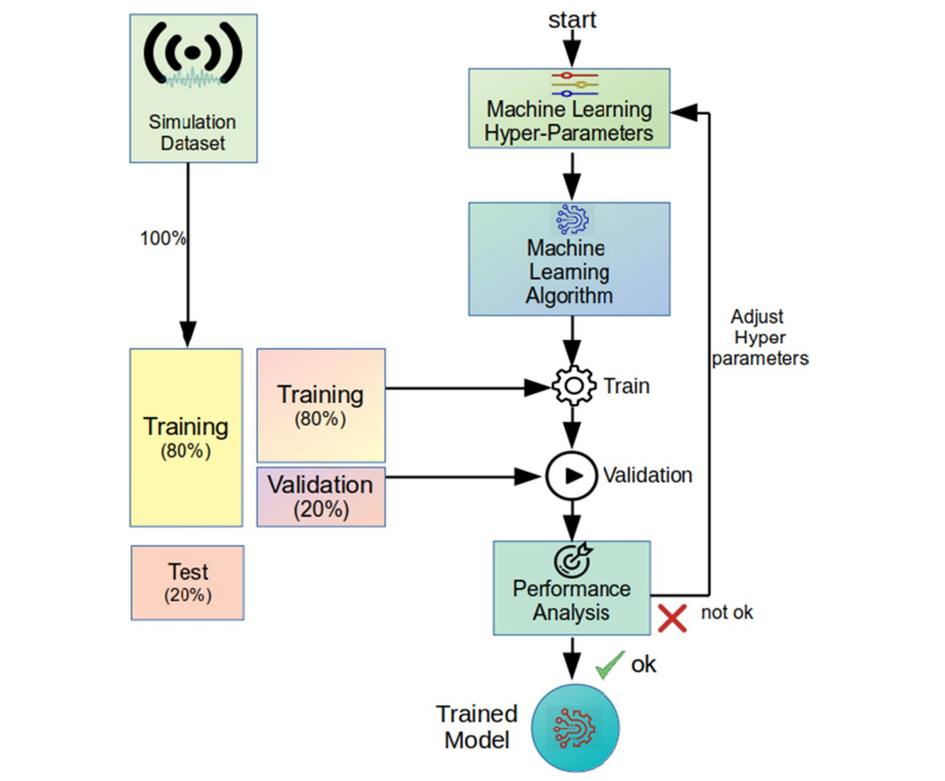


Fig. 7 Training process using only simulation dataset

from 10 to 1200 and the best trade off between mean distance error and number of base estimators was obtained with 80 estimators and a mean distance error of 120.98 m. By using 1200 base estimators, we obtain a mean distance error of 120.49 m, a very modest improvement over 80 estimators. Figure 12 presents the mean distance error evaluated considering the number of base estimators for random forest.

For XGBoost, the best mean distance error, 87.83 m, was obtained using 80 base estimators. Figure 13 presents the mean distance error evaluated considering the number of base estimators for XGBoost.

Since we have defined the number of base estimators from both random forest and gradient boosting (XGBoost), a new model was trained using both simulation dataset and emitter dataset. The generated model was then used for estimating the target position $\mathbf{p}_t$ from real world data using target dataset. For random forest, the mean distance error obtained when applying the Target Dataset was 148.51 ± 115.84 m and for XGBoost, the mean distance error when applying the target dataset was 145.63 ± 121.20 m. If we just used simulation data for traning and validation and apply the generated model to evaluate the target dataset, the the mean distance error obtained was 237.54 ± 138.78 when using random forest model and 219.31 ± 134.95 when using XGBoost model. So, simulation data (simulation dataset) and real data from one known position (emitter dataset) we could notice and mean distance error improvement of 89 m for Random Forest and 74 m for XGBoost.

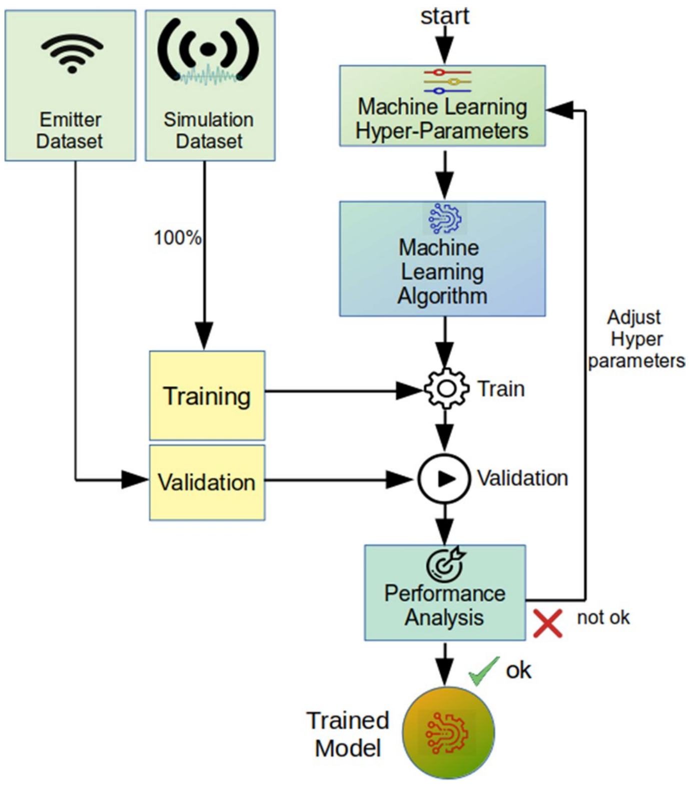


Fig. 8 Training process using simulation dataset and emitter dataset

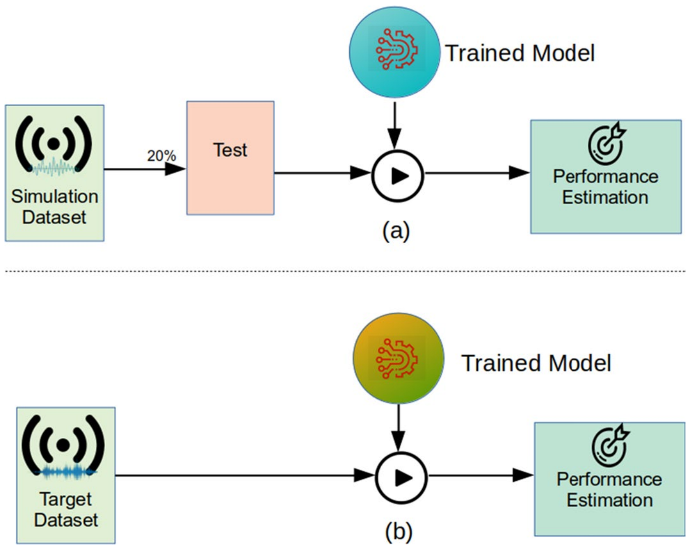


Fig. 9 Estimating model's error on target dataset

For sure, when considering the real scenario, it is obvious that the mean distance errors are bigger than the mean distance errors from the simulation scenario. Anyway, when using just 1000 measurements from a known position (emmiter dataset)

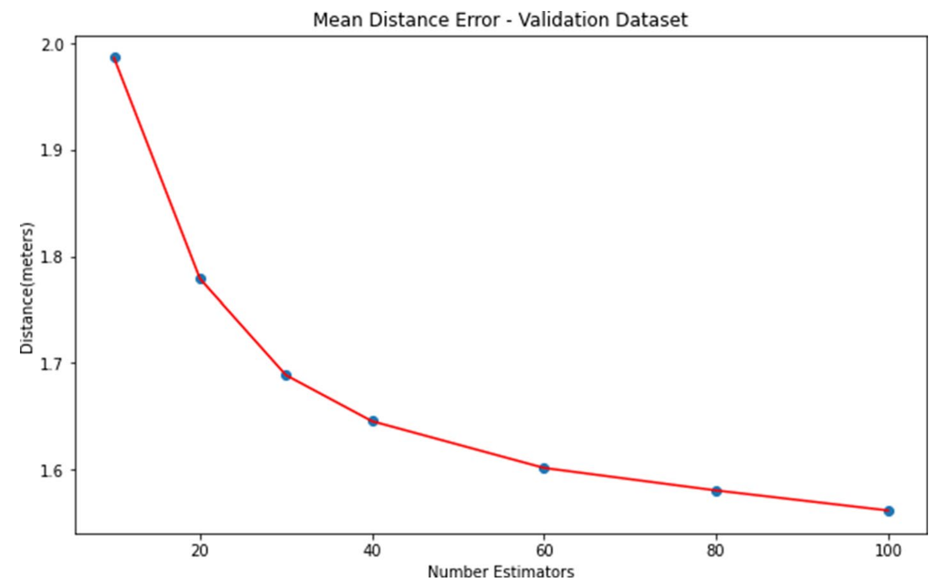


Fig. 10 Estimating model's error on test dataset for random forest

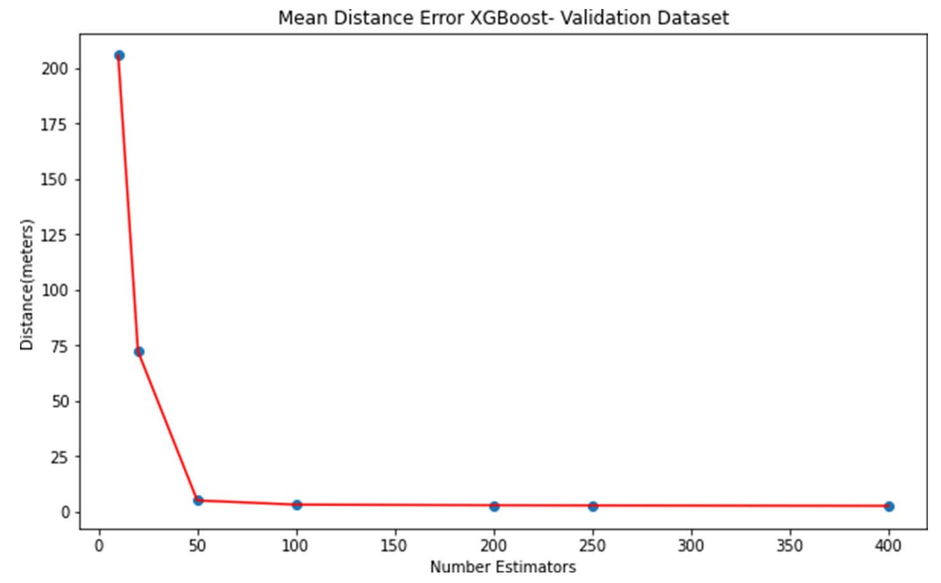


Fig. 11 Estimating model's error on test dataset for XGBoost

in training process we obtained a considerable improvement when using just simulation data to predict real data. Still, if we consider the Simulation Dataset's total area (788 m $\times$ 736 m $=$ 579,968 m^2), the random forest model obtained a mean distance error value of 148.51 m which defined an area $A = \pi r^2$ of 69,288.5 m^2 which represents 11.94% of total area. For XGBoost model, we obtained a mean distance error value of 145.63 m which defines an area $A = \pi r^2$ of 66,627.20 m^2 that represents 11.48% of the total area. So, in both cases, the target's search area are reduced by the amount of 9 times the total area.

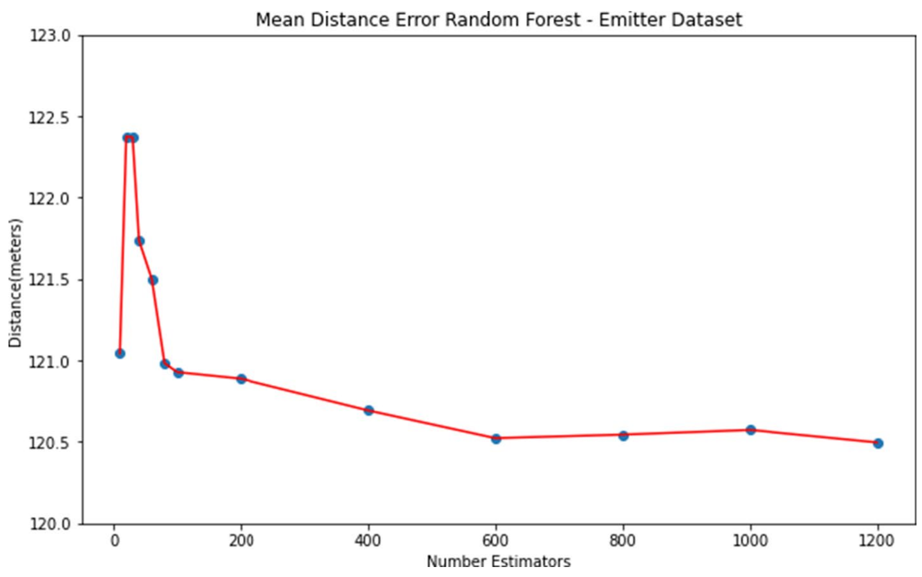


Fig. 12 Estimating model's error on emitter dataset for random forest

In both cases, about 46% of all predictions had mean distance error less than 75 m, which would reduces the target's search area by almost 33 times the total area. Figure 14 shows an example of a 75 m radius search area bounded by the simulation region (black-edged rectangle). The red region defines the circle with a radius of 75 m and the red point in the center defines the actual target position, usually not known. When using just simulation data to predict real data, 46% of all predictions had mean distance error of 127 m, represented as the external blue circle.

When considering model's time and CPU's performance in the Real Scenario, we come to the following observations:

- It is clear that the random forest algorithm used much more computer resources; on the other hand, it allows parallel execution due to the fact that the internal structure of the decision trees is independent. It should be noted that a Random Forest model with 80 estimators used more than 6 GB of RAM during training, and it took 30 min to be created using *Google Colaboratory*'s basic default CPU configuration;
- Gradient boosting uses less RAM and, in this particular case, it takes less time to create the model (mostly because the base estimators are stumps). The gradient boosting model with 80 base estimators used about 2.5 GB RAM, and it took 10 min to the generate the model on *Google Colaboratory*'s basic default CPU configuration.

Therefore, gradient boosting implementation (XGBoost) achieved better time and CPU performance. The main reason seems to be that the amount of base estimators of both models were not big enough to allow the parallel training from random forest to surpass gradient boost serial training.

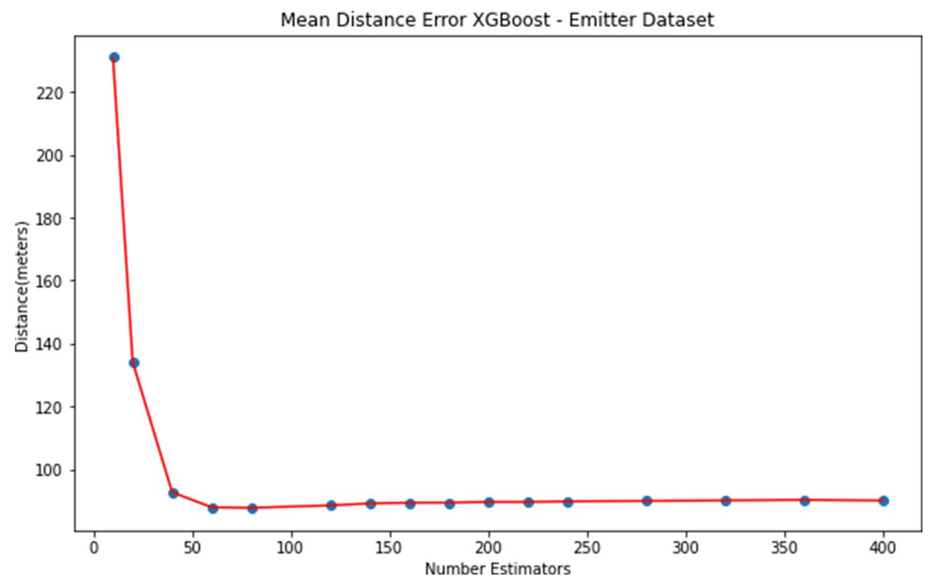


Fig. 13 Estimating model's error on emitter dataset for XGBoost

6.4.1 Experiment with noisy features

To evaluate how random forest and XGboost algorithms would behave in outdoor hostile position localization scenario when adding noise, we have added to all 160 features of the Target Dataset, a zero mean Gaussian noise with γ standard deviation. Each feature has its own mean and standard deviation, so we define each the standard deviation γ_i as

$$\gamma_i = level \times \sigma_i, \tag{10}$$

where σ_i is the standard deviation of a given feature and i is from range 1 to 160. Since there are 160 features, the noise applied to each feature will be proportional (*level*) to its own standard deviation.

Thus, for each feature i, the generated Gaussian noise will have the following distribution:

$$P(x) = \frac{1}{\gamma_i\sqrt{2\pi}} e^{-(x-\mu)^2/2\gamma^2}. \tag{11}$$

In this experiment, the range of the factor *level* used was: from 0.1 to 1.0, with an increment of 0.05 and from 1 to 9, with an increment of 1. Figure 15 shows the noise experiment process.

Figure 16 shows the results of this experiment. The main red line is the average Euclidean distance error between actual and predicted positions of all 2973 observations from the target dataset using random forest. Assuming μ as mean and σ as the standard deviation of the random forest positioning errors the red area defines $\mu - \sigma$ and $\mu + \sigma$. The main green line is the average Euclidean distance error between actual and predicted positions of all 2973 observations from target dataset using gradient boosting. The green area defines its $\mu - \sigma$ and $\mu + \sigma$ limits.

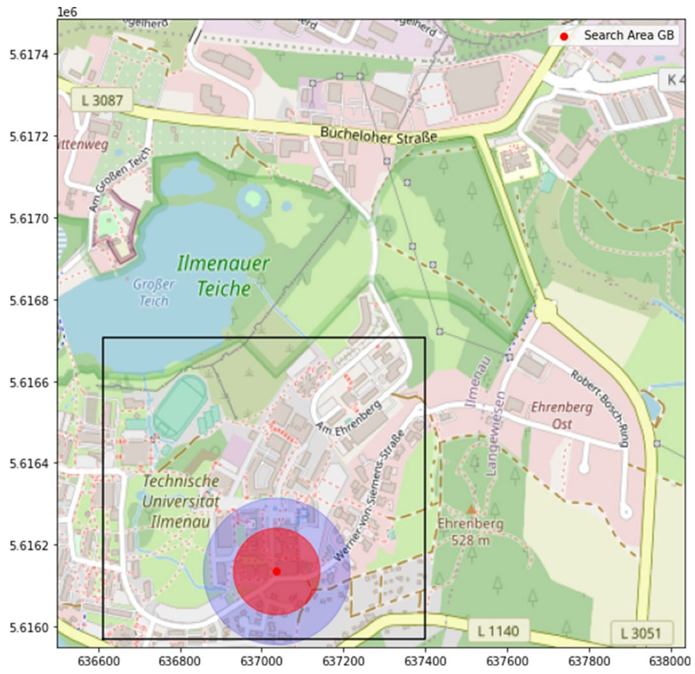


Fig. 14 An example of a 75 m radius search area from the proposed method and a 127 m radius search area when using just simulation data to predict real data

Evaluating both models with features corrupted by the same type of noise, random forest and gradient boosting obtained about the same results when the level of the noise was low. Random Forest obtained slightly better results when the noise level was greater than 2. So, the first important finding is that, in a noisier environment, one should give preference to random forest models. This is expected, because random forest uses bagging as a ensemble method and the predictions of a single tree of the random forest are highly sensitive to noise, the average of many trees is not, since the trees are not correlated.

Still from Fig. 16, we notice that the standard deviations of the Euclidean distance errors are high, which means that the methods are not robust to noise. The main cause of these high values is that both models were generated using mainly simulation data and on real world data (another data distribution). However, several locations were obtained with error lower than 50 meters. In terms of the localization task, 50-m error may be considered as satisfactory because the model was created using synthetic data and optimized using only a small amount of real world data.

Figure 17 presents the performance for both (a) random forest and (b) gradient boosting model when the noise variance is set to 2. The yellow points are all from the target dataset. The blue points are position estimations with random forest error less than 50 m, adding up to 172 points and the red points are position estimation with gradient boosting error less than 50 m, adding up to 133 points.

The second finding is that, considering all target observations with error less than 50 m, there is no clear zone where Random Forest is better than gradient boosting or vice-versa.

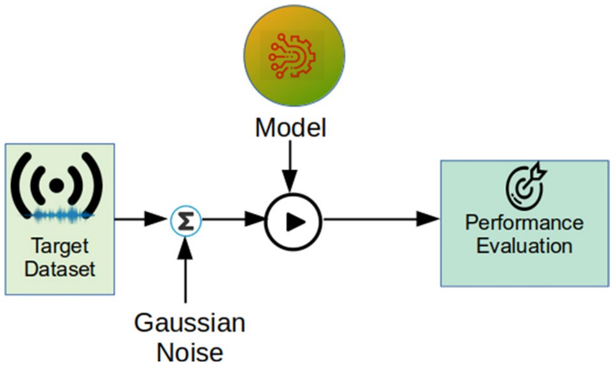


Fig. 15 Noise experiment process

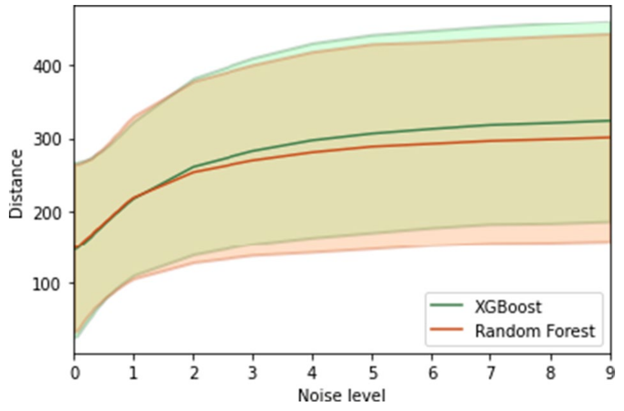


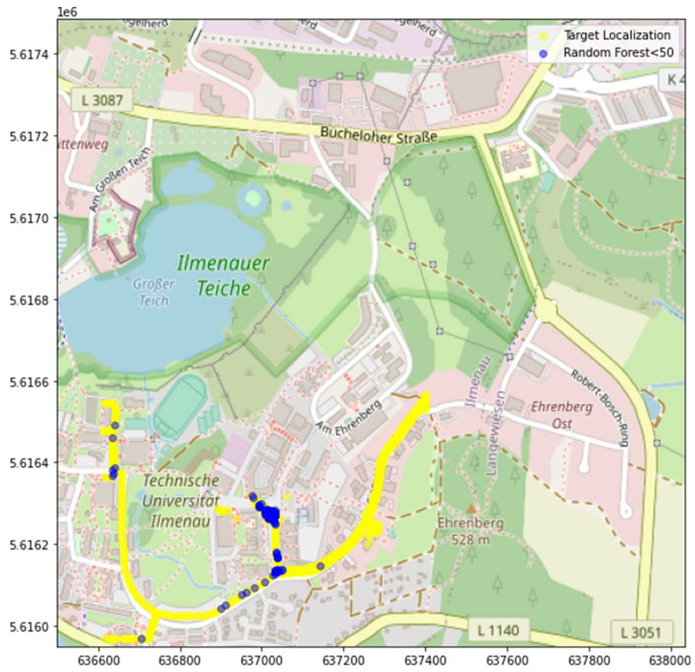
Fig. 16 Noise effects over euclidean distance error for random forest model and gradient boosting model

6.4.2 Mismatching experiment

In the mismatching experiment, a subset of features is nullified and the performance of the model is evaluated. The selected noise level (variance) for this experiment was set to 2—the value in which both models have comparable errors when using the emitter dataset in training process. Figure 18 presents this process.

Since the 4 sensors receive 20 rays (parameters pairs of the amplitude and the delay—α_i, τ_i), for each step we have selected the n most significant pairs of features, where n is in the range from 1 to 20. Figure 19 presents the results. The main red line is the average Euclidean distance error between actual and predicted positions of all 2973 observations using random forest. The red area defines the limits $\mu - \sigma$ to $\mu + \sigma$. The main green line is the average Euclidean distance error between actual and predicted positions of all 2,973 observations using gradient boosting. The green area defines the region between $\mu - \sigma$ and $\mu + \sigma$.

Both models did a good job in dealing with mismatching because the models have not suffered significant variations in performance (mean error) with the variation of the number of available characteristics. One reason for that is that many of the features from the simulation dataset are already zero.



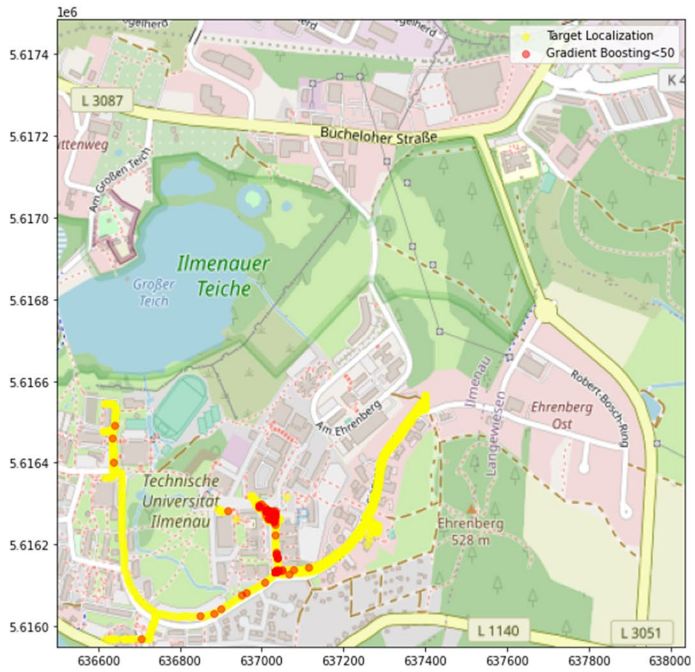


Fig. 17 Position estimation experiment (with noise level equal to 2) where the error was less than 50 m for **a** random forest (blue) and **b** gradient boosting (red), the yellow line is the positions assumed by the target

Figure 20 presents the performance for both random forest and gradient boosting model when the number of feature pairs is set to 1 (one amplitude and one delay), which is the worst case for NLOS scenario. All Yellow points are from the target dataset. The Red points

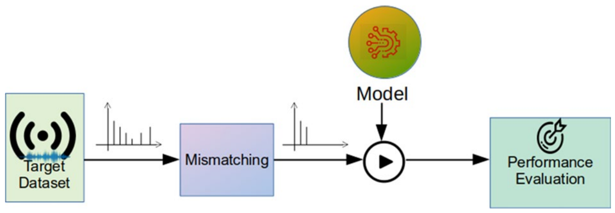


Fig. 18 Mismatching experiment process

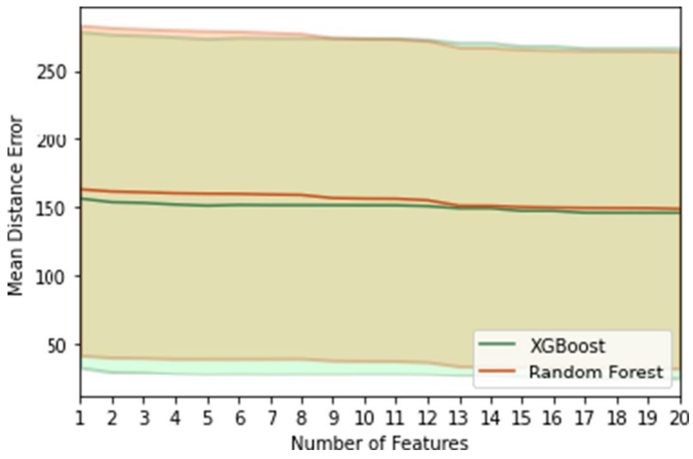


Fig. 19 Mismatching effects over euclidean distance error for random forest model and for gradient boosting model

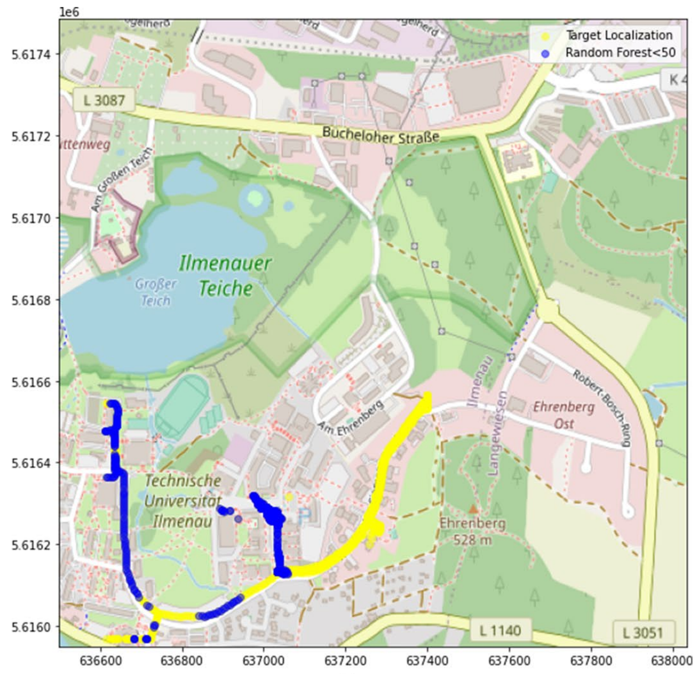
are positions where random forest error was less than 50 m, and the Blue points are location where Gradient Boosting error was less than 50 meters.

Another finding, similarly to the noisy experiment when considering all target observations where the error is less than 50 m, some localizations are better modelled by Random Forest and others by gradient boosting. Again, there is no clear domain where Random Forest is better than gradient boosting or vice-versa. Finally, both random forest and XGBoost achieved almost no loss due to the decrease in the number of parameters pair available, indicating that most of the information are in the first parameter pair of the amplitude and the delay—α_1, τ_1 from the 4 sensors.

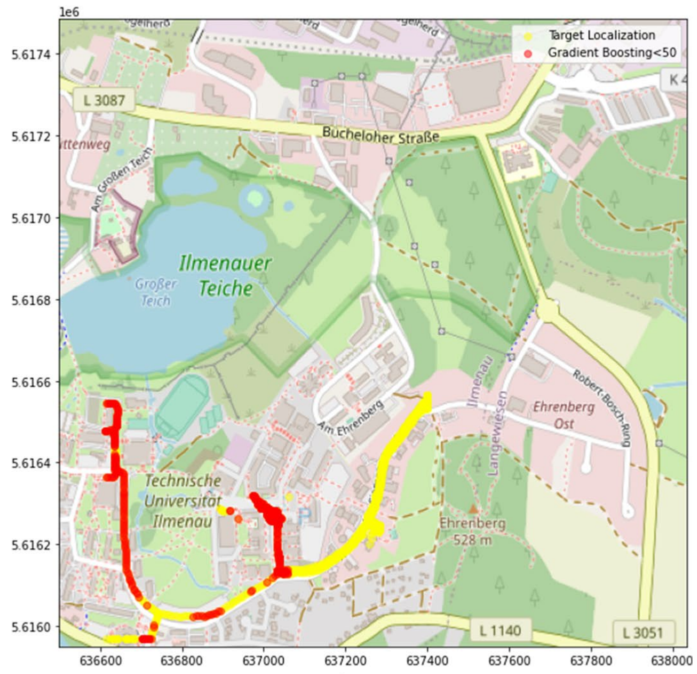
7 Conclusion

This paper presents a comparison of algorithms random forest and gradient boosting when employed to enhance a kernel-based machine learning localization scheme using TDOA fingerprinting in an outdoor hostile position localization problem.

The results presented herein can serve as guidelines in similar problems to adjust the number of estimators and to help in the definition of which machine learning



a



b

Fig. 20 Position estimation experiment where the error was less than 50 m for **a** random forest (blue) and **b** gradient boosting (red) for the mismatching experiment

implementation is more suitable and how to use it when there is limited real data for the area of interest, similar to outdoor hostile position localization problem.

Fingerprinting localization methods can deal with measurement errors by continuously improving the estimation based on the real-world measurements available in the area of interests. So, in this work, we have used a fixed located real-world data to help improving machine learning performance when estimating real-world measurements. To validate the proposed algorithms, we have evaluated them in two scenarios: simulation scenario, which uses just simulation data, and real scenario, which uses both simulation and limited real-world data for predicting actual positions from real-world data. The latter scenario was much more challenging.

For real scenario, which we called outdoor hostile position localization and using only 4 passive sensors, a trained model from simulated data from an area of 579,968 m^2 and real world data from a fixed located emitter, it was possible to estimate the position of a moving target $\mathbf{p}_t = [x_t, y_t]^T$ by reducing the search area by 9 times for both Random Forest and XGBoost. Still, in almost 50% of cases, the reduction was about 33 times, which is very considerable. Hence, the proposed algorithms and method make our approach very appealing for practical applications in NLOS propagation environments.

Geo-information is a promising field in signal processing for localization, because it can reduce error of the radio-map created by simulation tools. We consider, for future work, to build a data fusion engine with cartographic database and signal processing; and to use optimization tools to deal with the raw information produced by multipath reflection in a TDOA system deployment.

Abbreviations

AOA: Angle of arrival; CART: Classification and regression trees; CIR: Channel impulse response; CRB: Cramer–Rao bound; CSI: Channel state information; FDOA: Frequency difference of arrival; KNN: K-nearest neighbor; LTE: Long-term evolution; NLOS: Non-line-of-sight; RSS: Received signal strength; RT: Ray-tracing; TDOA: Time-difference of arrival; VCC: Volume cross-correlation; WSN: Wireless sensor network.

Acknowledgements

We thank the editor and anonymous reviewers for their helpful comments and valuable suggestions.

Authors' contributions

All authors took part in the discussions of the work described herein and should be considered co-correspondent authors. MNS conducted a research and investigation process in multipath fingerprints, specifically performing the simulations, developed the machine learning algorithm for position estimation using out the experimental setup including the measurement campaign. JAAJr and RST oversight and leadership responsibility for the research activity planning and execution, including mentorship external to the core team, the idea of multipath fingerprints is aligned with his ongoing research activities in Passive Cooperative Localization in future communication systems. RS and RPF conducted a research in applying machine learning for position estimation, developed the machine learning architecture and applied it into available multipath datasets. JCD conducted research, investigation process and mentoring in applying machine learning for position estimation, beyond analysis of the results. All authors read and approved the final manuscript.

Funding

Open Access funding enabled and organized by Projekt DEAL. We thank the Brazilian Agency CAPES for partial funding of this work.

Availability of data and materials

The datasets used and/or analyzed during the current study are available upon request to the corresponding author.

Declarations

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Electronic Measurements and Signal Processing, University of Technology of Ilmenau, Helmholtzplatz 2, 98693 Ilmenau, Germany. [2]Military Institute of Engineering, Praça Gen Tibúrcio 80, Rio de Janeiro 22.290-270, Brazil.

Received: 10 November 2020 Accepted: 29 April 2021
Published online: 11 May 2021

References

1. J. Schiller, A. Voisard, *Location-Based Services* (Elsevier, Amsterdam, 2004).
2. M. van der Sluis, Y. de Haas, B. de Klerk, T.B. Rodenburg, E.D. Ellen, Assessing the activity of individual group-housed broilers throughout life using a passive radio frequency identification system-a validation study. Sensors **20**(13), 3612 (2020). https://doi.org/10.3390/s20133612
3. S. Kunze, A. Weinberger, R. Poeschl, A software defined radio based implementation for the radio frequency analysis of signals from unmanned aerial systems. In: *2019 29th International Conference Radioelektronika (RADIOELEKTRONIKA)*, pp. 1–6 (2019). IEEE
4. M. Liu, L. Cheng, K. Qian, J. Wang, J. Wang, Y. Liu, Indoor acoustic localization: a survey. Hum.-Centr. Comput. Inf. Sci. **10**(1), 2 (2020)
5. W. Li, Z. Chen, X. Gao, W. Liu, J. Wang, Multimodel framework for indoor localization under mobile edge computing environment. IEEE Internet Things J. **6**(3), 4844–4853 (2018)
6. L.B. Fertig, M.J. Baden, J.C. Kerce, D. Sobota, Localization and tracking with multipath exploitation radar. In: *2012 IEEE Radar Conference*, pp. 1014–1018 (2012). https://doi.org/10.1109/RADAR.2012.6212286
7. Setlur, P., Devroye, N.: Bayesian and Cramer–Rao bounds for single sensor target localization via multipath exploitation. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5845–5849 (2013). https://doi.org/10.1109/ICASSP.2013.6638785
8. M.N. de Sousa, R.S. Thomä, Enhancement of localization systems in NLOS urban scenario with multipath ray tracing fingerprints and machine learning. Sensors **18**(11), 4073 (2018)
9. A. Haniz, G.K. Tran, K. Sakaguchi, J. Takada, D. Hayashi, T. Yamaguchi, S. Arata, Hybrid fingerprint-based localization of unknown radios: measurements in an open field. In: *2017 IEEE Asia Pacific Microwave Conference (APMC)*, pp. 865–868 (2017). https://doi.org/10.1109/APMC.2017.8251586
10. N. Liu, J.-S. Pan, J. Wang et al., An adaptation multi-group quasi-affine transformation evolutionary algorithm for global optimization and its application in node localization in wireless sensor networks. Sensors **19**(19), 4112 (2019)
11. G.-Y. Chen, M. Gan, C.P. Chen, L. Chen, A two-stage estimation algorithm based on variable projection method for GPS positioning. IEEE Trans. Instrum. Meas. **67**(11), 2518–2525 (2018). https://doi.org/10.1109/TIM.2018.2826798
12. L. Wu, C.-H. Chen, Q. Zhang, A mobile positioning method based on deep learning techniques. Electronics **8**(1), 59 (2019). https://doi.org/10.3390/electronics8010059
13. S. Jhaveri, I. Khedkar, Y. Kantharia, S. Jaswal, Success prediction using random forest, CatBoost, XGBoost and AdaBoost for kickstarter campaigns. In: *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 1170–1173 (2019)
14. Y. Wang, C. Xiu, X. Zhang, D. Yang, Wifi indoor localization with CSI fingerprinting-based random forest. Sensors **18**(9), 2869 (2018)
15. T. Janssen, R. Berkvens, M. Weyn, Comparing machine learning algorithms for rss-based localization in lpwan. In: *International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, pp. 726–735 (2019). Springer
16. M.N. de Sousa, R.S. Thomä, Applying random forest and multipath fingerprints to enhance TDOA localization systems. IEEE Antennas Wirel. Propag. Lett. **18**(11), 2316–2320 (2019)
17. Y. Li, H. Huang, Z. Wu, Animal sound recognition based on double feature of spectrogram. Chin. J. Electron. **28**(4), 667–673 (2019). https://doi.org/10.1049/cje.2019.04.005
18. M. Luckner, B. Topolski, M. Mazurek, Application of XGBoost algorithm in fingerprinting localisation task. In: *IFIP International Conference on Computer Information Systems and Industrial Management*, pp. 661–671 (2017). Springer
19. C. Dong, J. Chen, W. Guo, J. Zou, A machine-learning-based hardware-trojan detection approach for chips in the internet of things. Int. J. Distrib. Sens. Netw. (2019). https://doi.org/10.1177/1550147719888098
20. S. Subedi, J.-Y. Pyun, Practical fingerprinting localization for indoor positioning system by using beacons. J. Sens. (2017). https://doi.org/10.1155/2017/9742170
21. V. Degli-Esposti, Ray tracing propagation modelling: future prospects. In: *The 8th European Conference on Antennas and Propagation (EuCAP 2014)*, pp. 2232–2232 (2014). https://doi.org/10.1109/EuCAP.2014.6902256
22. R. Górak, M. Luckner, Modified random forest algorithm for Wi–fi indoor localization system. In: *International Conference on Computational Collective Intelligence*, pp. 147–157 (2016). https://doi.org/10.1007/978-3-319-45246-3-14
23. H. Miao, K. Yu, M.J. Juntti, Positioning for NLOS propagation: algorithm derivations and Cramer–Rao bounds. IEEE Trans. Veh. Technol. **56**(5), 2568–2580 (2007). https://doi.org/10.1109/TVT.2007.899948
24. M.B. Lecocke, M.A. Koets, J.L. Alvarez, L.T. McDaniel, M.R. Darnell, M.D. Lillywhite, A configurable timing and communications engine for radio positioning with implementations for an FPGA or an ASIC. In: *2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers*, pp. 759–763 (2009). https://doi.org/10.1109/ACSSC.2009.5469957
25. I. Vin, D.P. Gaillot, P. Laly, M. Lienard, P. Degauque, Multipath component distance-based fingerprinting technique for non-cooperative outdoor localization in NLOS scenarios. IEEE Trans. Antennas Propag. **62**(9), 4794–4798 (2014). https://doi.org/10.1109/TAP.2014.2327142
26. B.R. Phelan, E.H. Lenzing, R.M. Narayanan, Source localization using unique characterizations of multipath propagation in an urban environment. In: *2012 IEEE 7th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pp. 189–192 (2012). https://doi.org/10.1109/SAM.2012.6250463
27. A.L. Paiva, W.C. Freitas, I.M. Guerreiro, H.B. Nascimento, Indoor localization algorithm based on fingerprint using a single fifth generation wi-fi access point. IEEE Latin Am. Trans. **16**, 2020–2026 (2018). https://doi.org/10.1109/TLA.2018.8447371
28. G. Pecoraro, S. Di Domenico, E. Cianca, M. De Sanctis, CSI-based fingerprinting for indoor localization using LTE signals. EURASIP J. Adv. Signal Process. (2018). https://doi.org/10.1186/s13634-018-0563-7
29. X. Guo, N. Ansari, L. Li, H. Li, Indoor localization by fusing a group of fingerprints based on random forests. IEEE Internet Things J. **5**(6), 4686–4698 (2018). https://doi.org/10.1109/JIOT.2018.2810601
30. J. del Peral-Rosado, G. Granados, R. Raulefs, E. Leitinger, S. Grebien, T. Wilding, D. Dardari, E. Lohan, H. Wymeersch, J. Floch, et al. Whitepaper on new localization methods for 5g wireless systems and the internet-of-things (2018)

31. M. Raspopoulos, C. Laoudias, L. Kanaris, A. Kokkinis, C.G. Panayiotou, S. Stavrou, 3d ray tracing for device-independent fingerprint-based positioning in WLANs. In: *2012 9th Workshop on Positioning, Navigation and Communication*, pp. 109–113 (2012). https://doi.org/10.1109/WPNC.2012.6268748
32. F. Fuschini, H. El-Sallabi, V. Degli-Esposti, L. Vuokko, D. Guiducci, P. Vainikainen, Analysis of multipath propagation in urban environment through multidimensional measurements and advanced ray tracing simulation. IEEE Trans. Antennas Propag. **56**(3), 848–857 (2008)
33. R. Zekavat, R.M. Buehrer, *Handbook of Position Location: Theory, Practice and Advances*, vol. 27 (Wiley, Hoboken, 2011).
34. L. Breiman, Random forests. Mach. Learn. **45**(1), 5–32 (2001)
35. Y. Amit, D. Geman, Shape quantization and recognition with randomized trees. Neural Comput. **9**(7), 1545–1588 (1997)
36. L. Breiman, Bagging predictors. Mach. Learn. **24**(2), 123–140 (1996)
37. A.J. Ferreira, M.A.T. Figueiredo, In: Zhang, C., Ma, Y. (eds.) Boosting Algorithms: A Review of Methods, Theory, and Applications, pp. 35–85. Springer, Boston (2012). https://doi.org/10.1007/978-1-4419-9326-7-2
38. M. Sammer, Boosting is a general ensemble method (2019). https://mbaskills.in/boosting/. Accessed 30 Dec 2019
39. B.Y. Shikur, T. Weber, TDOA/AOD/AOA localization in NLOS environments. In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6518–6522 (2014)
40. J. Gante, G. Falcão, L. Sousa, Deep learning architectures for accurate millimeter wave positioning in 5G. Neural Process. Lett. (2019). https://doi.org/10.1007/s11063-019-10073-1
41. D. Corte-Valiente, J.M. Gómez-Pulido, O. Gutiérrez-Blanco, J.L. Castillo-Sequera et al., Localization approach based on ray-tracing simulations and fingerprinting techniques for indoor-outdoor scenarios. Energies **12**(15), 2943 (2019)

Publisher's Note