Comparison of Random Forest and Gradient Boosting Fingerprints to Enhance an Outdoor Radio-frequency Localization System

Marcelo Nogueira de Sousa (marcelo.nogueira@tu-ilmenau.de)
Technische Universitat Ilmenau https://orcid.org/0000-0002-4002-3499
Ricardo Sant'Ana
Military Institute of Engineering: Instituto Militar de Engenharia
Riegel P. Fernandes
Military Institute of Engineering: Instituto Militar de Engenharia
Julio Cesár Duarte
Military Institute of Engineering: Instituto Militar de Engenharia
José A. Aploinário
Military Institute of Engineering: Instituto Militar de Engenharia
Reiner S. Thomä
Ilmenau University of Technology: Technische Universitat Ilmenau

Research

Keywords: Wireless Positioning, Hybrid Positioning, Machine Learning, Ray Tracing Fingerprints

Posted Date: November 19th, 2020

DOI: https://doi.org/10.21203/rs.3.rs-108739/v1

RESEARCH

Comparison of Random Forest and Gradient Boosting Fingerprints to Enhance an Outdoor Radio-frequency Localization System

Marcelo N. de Sousa[??*†], Ricardo Sant'Ana[??], Julio Cesar Duarte[??], José A. Apolinário Jr.[??] and Reiner S. Thomä[??]

Abstract

Machine Learning framework adds a new dimension to the localization estimation problem; it tries to find the most likely position using processed features in a radio map. This paper compares the performance of two machine learning tools, Random Forest (RF) and XGBoost, in exploiting the multipath information for outdoor localization problem. The investigation was carried out in a noisy outdoor scenario, where non-line-of-sight between target and sensors may affect the location of a radio-frequency emitter strongly. It is possible to improve the position system performance by using fingerprints techniques that employ multipath information in a Machine Learning framework, which operate a dataset generated by ray-tracing simulation. Usually, real measurements produce the fingerprints localization features, and there is mismatching with the simulated data. Another drawback of NLOS features extraction is the noise level that occurs in position processing. Random Forest algorithm uses fully grown decision trees to classify possible emitter position, trying to achieve error mitigation by reducing variance. On the other hand, XGBoost approach uses weak learners, defined by high bias and low variance. The results of the simulation performed aims to be used as a design parameter to perform hyperparameter refinements in similar multipath localization problems.

Keywords: 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, and it is possible to extract the radio frequency emitter position using a wireless sensor network (WSN). Processing techniques can be applied for computing the Angle of Arrival (AOA), the Time of Arrival in each position or even take the Time Difference of Arrival (TDOA) or Frequency Doppler Difference of Arrival (FDOA) in each combination pair sensors [**?**].

In an outdoor scenario, the non-line-of-sight situation between the emitter and the sensors affects the position estimation [**?**]. The multipath is as a limitation in position processing, but it is possible to use it for localization [**?**], and tracking vehicle where multipath exploitation is performed to track the target. In [**?**], the multipath information is used together with the image theory to locate the emitter, showing the feasibility to perform location using only one sensor.

Some hybrid approaches like [**?**] try to enhance the performance of outdoor TDOA systems, with the multipath information, machine learning and propagation simulation tools. The research [**?**] showed the same idea of position fingerprints, with a random forest (RF) algorithm and a synthetic volume cross-correlation (VCC) function, for extracting the multipath features from the TDOAs measurements. Among several possible machine learning or deep learning approaches, [**?**] and [**?**], Random Forest and XGBoost are two different tools [**?**] applied for position fingerprint approach that can handle the discrepancies in the data quality, which causes of over-fitting, under-representative data samples, and stochastic algorithms.

The Random Forest uses decision trees, which are very prone to [**?**]; typically, it is used to achieve higher accuracy, based on different sets of attributes. On the other hand, XGBoost is a boosting method, which is built on weak classifiers. Both techniques aim to enhance the performance of a location system in

*Correspondence: marcelo.nogueira@tu-ilmenau.de
??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
†Equal contributor

outdoor scenarios, where there are technical restrictions for the deployment in line-of-sight (LoS) conditions.

While RF algorithm is based in fully grown decision trees to classify possible emitter position, the XGBoost is based on weak learners (high bias, low variance). In terms of decision trees, 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, by aggregating the output from many models.

This approach, because in an outdoor environment is not always possible to build the dataset with real signals, and some simulation tool is used to create a radio-map. In this case, some miscalculations may occur between the dataset produced with simulation tools and the actual measurements provided by the sensors. On the other side, the engine in machine learning can produce some outliers in the position estimation, where the tuning parameters can not deal with intense noise in the real measurements setup.

For this reason, it is essential to estimate and compare among several possibilities, which implementation is more robust to deal with differences caused by noise and the mismatching between the synthetic data and the real one.

The main contribution of this work is to compare Boosting and RF models regarding mismatching and the noise in measurements of multipath fingerprints for position localization in outdoor models presented by [**?**] and [**?**]. Another contribution of this work is to show some guidelines to better tune the hyper-parameters in the machine learning approaches for outdoor positioning.

The rest of the paper is organized as follows: Section 2 shows the state-of-the-art in Localization Systems using multipath exploitation. Next, Section 3 defines the localization problem and the effect of the NLOS between the emitter and sensors and explains how the ray-tracing simulation tool is used to produce the used by the machine learning algorithms. Section 4 presents the main aspects of the fingerprint framework with machine learning, explaining the general characteristics of the Random Forest and Gradient Boosting methods. In Section 5, a validation setup for comparing the different model performance, regarding noise and mismatching effects. Finally, the Section 6 show our conclusions and the perspective for future work.

2 Methods

The ray-tracing as an electromagnetic simulation tool for modelling the signal propagation can extract the channel impulse response of the signal that arrives in each TDOA sensor, [**?**]. So, we have used the AWE WinProp ray-tracing to extract the CIR fingerprints. We have created a fingerprint using the same design as [**?**] and using machine learning, and it is possible to predict a target position based on where amplitude and delay which arrives on the sensors. All these pieces of information are part of the Simulation dataset. There are two more data-sets available: (i) emitter data which is a real data from 1000 measurements from same localization using 4 sensors and (ii) a target dataset which is a real data measurement from different position using 4 sensors [**?**].

We have chosen two machine learning algorithms to work out on set-ups for simulation: Random Forest and GBoost. In algorithm list 1 we have an overview of the machine learning training process: it is interesting to note that we use actual data (sender) for the validation step (setting hyper-parameters).

Algorithm 1: Machine Learning trainig process

input : Features from each sensor S_1, S_2, S_3 and S_4
output: Model for predicting (X_e, Y_e)
From Emmiter Dataset:;
· $\alpha_i, \Delta_{\tau,i}$ for each of the 4 sensors;
· Emitter position vector $\vec{x}$;
From Simulation Dataset: ;
· $\alpha_i, \Delta_{\tau,i}$ for each of the 4 sensors;
· Position vector $\vec{x}$ for each point from 10 to 10 meters;
· Buildings descriptions.;
Machine Learning training process;
· Select model's hyperparameters ;
· Define loss function as RMSE ;
· Use data from Simulation Dataset ;
Build a Regression Model:;
· 20 $(\alpha_i, \Delta_{\tau,i})$ for each sensor;
· 2 Outputs: $[X_s, Y_s]^T$;
Tunning Hyperparameters:;
· Use data from Emmiter 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 until performance are considered suitable;
Obtain Final Model;

So, the dataset generated by synthetic VCC produced by ray-tracing (Simulation dataset) was the only data used for training the model. Some real VCC extracted from the TDOA system (Emitter dataset) was used to adjust and define machine learning hyper-parameters.

After generating a model using Random Forest and GBoost, we have used the Target Dataset to predict locations (X_t, Y_t) and compare the performance of each of the models. Two simulations were performed: the first involves evaluating the model response to the addition of Gaussian noise to the features of Target Dataset; the second test evaluates the models' response to when cancelling features from target dataset.

The Fig.1 presents the suggested framework. First, the machine learning engine uses the simulation

scenario dataset, produced by ray-tracing simulation, optimizes the hyper-parameters and creates the prediction model. Later the model and the machine learning implementation (XBoost or Random Forest) uses the model to perform the position prediction. In parallel, we add noise level and impose an artificial mismatching on data provided by the emitter, evaluating the prediction error using the Euclidean distance between the prediction and the real position.

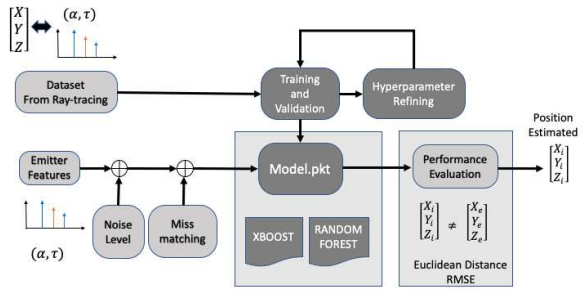

Figure 1 Overview of Proposed Approach

3 State-of-the-art in Localization Systems

Traditional TDOA techniques are strongly affected by reflected and diffracted rays in the environment, when there are measurements with non-line-of-sight (NLOS) paths, the location errors can be substantial. The performance of the localization systems depends on the signal processing algorithms and the channel characteristics, where the non-line-of-sight (NLOS) rays affect the time measurements [?].

There are several approaches to deal with multipath in localization systems; the classical TDOA localization approaches have relied only in on line-of-sight (LoS) propagation where, usually, there is degradation in performance due to the non-line-of-sight (NLoS) rays. Regarding antenna simplicity, small size, weight and power requirements (SWaP) [?], the TOA and TDOA techniques are the most popular schemes used for localization in wireless networks.

NLOS multipath propagation introduces an error inherent into the localization because they alter the propagation paths and add additional delays to the channel. It is possible to obtain the propagation information to deal with NLOS using "RF fingerprinting", either performing extensive measurement campaigns or using ray-tracing simulation software in the environment considered.

Ray tracing is an electromagnetic simulation tool for modelling the signal propagation, assuming the geometric optics approximation to trace the ray propagation paths in a defined scenario. It is possible to use the technique to build a Fingerprint of the likely rays to extract the time difference information. In [?], a Multipath Database characterization was created with a grid of possible emitter position, where the angle of arrival (azimuth and elevation) and time of arrival where recorded, giving a signature for each possible transmitter location in an area of interest that is populated via ray-tracing software simulations. The received signal with NLoS components is compared with the values in this database, to estimate the emitter position that has the same multipath information.

The approach described in [?] uses multipath characteristics of the scenario to build a database of the NLOS rays, applying a clustering procedure to match the real measurements with the simulated one to locate the emitter. In [?], the authors presented different localization fingerprints using Received Signal Strength (RSS) and the K-nearest neighbor (KNN) algorithm. There are different types of "Fingerprints", as discussed in [?] and [?], where the position estimation can be evaluated using the channel state information (CSI) for Long Term Evolution (LTE) and gives a performance improvement.

In [?], several approaches for enhancement localization systems in 5G and IoT are presented where some multipath fingerprint approaches for indoor and outdoor positioning where introduced. A modified version of the Random Forest algorithm, performed by [?], makes a localization system with the information of WIFI access points in indoor scenarios, the system gives the target position as a classification processing. The channel impulse response in an urban scenario can be extracted using ray-tracing, which offers a prediction of the directional and temporal structure of the received multipath.

4 Multipath in Localization Systems

The Machine Learning Fingerprint framework presented in [?] used a ray-tracing simulation tool to extract the multipath features for all the possible emitter positions in the scenario. Sometimes due to operational or physical restrictions, it is challenging to build a dataset with real measurements, that is why some studies, like [?], tried to use propagation simulation tools. Ray-tracing is an electromagnetic simulation tool for modelling the signal propagation, which considers geometric optics approximation. Following [?], the term "ray tracing" fingerprint relates to a "radio map" of Received Signal Strength (RSS) in a coverage prediction, that take into account the output power of the emitter to perform the position estimation, [?].

The Fig.2 shows the performance of a localization system in an outdoor scenario where the NLOS effect

degrades the position estimation. The Fig.3 shows the multipath fingerprints produced by ray-trace simulation from Sensor Position 1. 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.

The Fig.5 and Fig.6 show that the multipath fingerprints also gives information about the propagation mechanism (reflection, diffraction or scattering). They represent the emitter-sensor interactions, and it is the base for the visibility matrix where each interaction is considered as a layer in a multilayer scheme. Therefore, the RT gives the information about each ray path, describing the edges and walls touched by the rays in the emitter-sensor path.

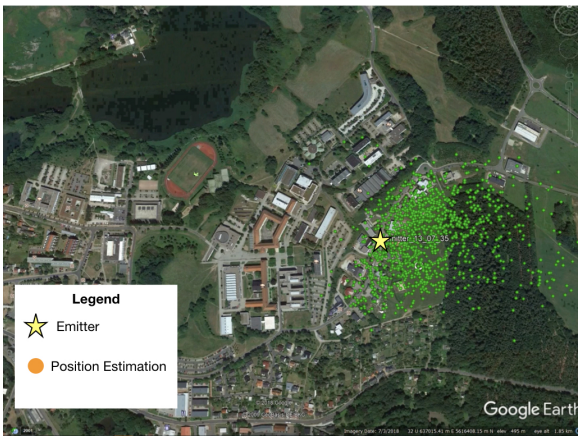

Figure 2 Performance of Localization System in Outdoor Scenario.

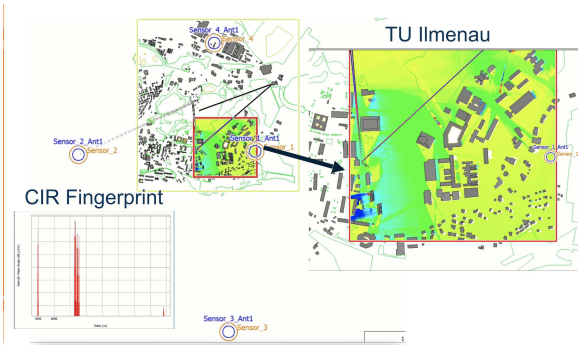

Figure 3 Buildings in Ray Tracing simulation.

The performance is highly dependable on the details given in the scenario setup. In practical outdoor implementations, the buildings are only represented by simpler structures, where windows and doors details are extracted out from the simulation. For a suburban outdoor scenario with simple buildings, the ray-trace does give reasonable information about the main specular components in the propagation channel.

The ray tracing simulation provided the site-specific channel impulse response, what 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.

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

The Fig.3 shows the Multipath Fingerprints produced by Ray Trace simulation from Sensor Position 1.

Following the approach of [**?**], the ray-tracing scenario can be decomposed walls and edges, the "view tree" represents the emitter-sensor interactions, and it is the base for the visibility matrix, where each interaction is considered a layer in a multilayer scheme. Therefore, the RT gives the information about the ray path, describing the edges and wall touched by the rays in the emitter-sensor path.

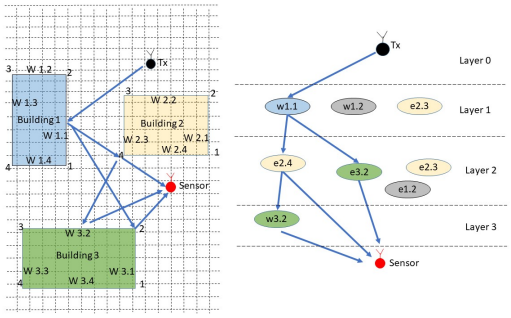

Figure 4 Wall and Edges form the Buildings in Ray Tracing simulation [?]

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

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

In this point, the approximation, not only in the scenario description but also in the in multipath information from ray tracing, can play an essential

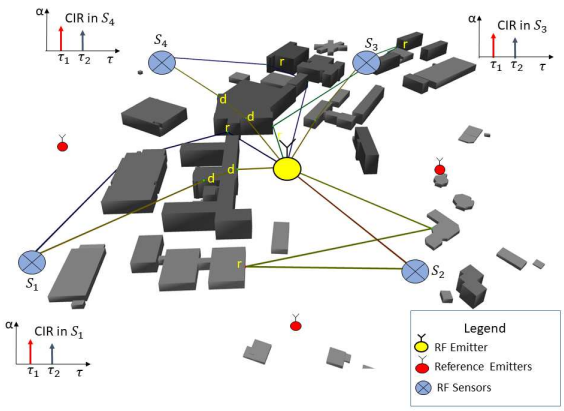

Figure 5 Extraction of CIR fingerprints using Ray Tracing.

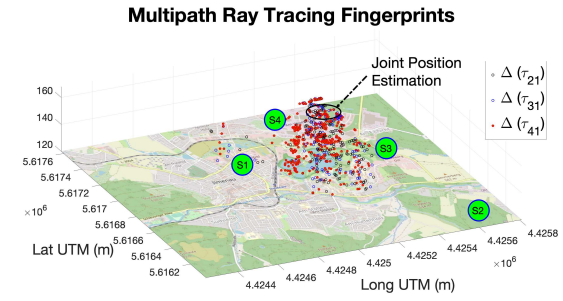

Figure 6 Ray Tracing Multipath Dataset, from [?]

role in the ML 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; it happens because we try to describe the learning of the target function from simulation training data.

4.1 Data Model for TDOA Localization

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

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

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,$\mathbf{n}$ is a vector that describes the zero mean noise that corrupts the measurements. TDOA location is done using the range differences where it assumed that the station is synchronized.

When the source emits a signal at instant t_0 (unknown), the lth sensor receives the signal at time t_l, with $l = 1, 2, ...L$. It is possible to obtain $L(L-1)/2$ distinct TDOAs, if there are four sensors, $\tau_{21}, \tau_{31}, \tau_{41}$. The time differences and range differences are related by the constant of the speed of light, using the range difference formulation withe the TDOA values we have:

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

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 notation in matrix form:

The $\mathbf{f(x)}_{TDOA}$ has the following structure:

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

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

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

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

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

In case of multipath the time differences or range differences presents an extra error caused by the NLoS:

$$\mathbf{e}_{nonlinear} = \mathbf{e}_{noise} + \mathbf{e}_{NLoS}, \tag{6}$$

The effects of the multipath are included in the signal data model and in the CRB, presented in [?], as an "extra error" in the estimation. The TDOA in NLoS scenario is, therefore, a standard system but with an extra noise in the estimation performance that leads to a wrong position estimation.

5 Machine Learning Algorithms for Localization Fingerprint

In machine learning, we usually try to create a model from data of a given distribution (training data) and evaluate the result of this model on data from the same distribution but not used in training.

In this work, we have 3 different data-sets available. The Simulation Dataset: dataset from a ray-tracing simulation tool, which contains 127 thousand samples with each sample have information of 20 pairs of the amplitude and the delay (α_i, τ_i) for each of the four receivers summing up 160 features and its respective localization x, y, z (labels).

The target dataset: a Data set which contains 2973 real data samples and each sample have information of 20 pairs of the amplitude and the delay (α_i, τ_i) received for each of the four receivers, summing up 160 features and its respective localization x, y, z (labels).

The Emitter Dataset: dataset which contains 1000 readings from the same single point where a single reading has information of 20 pairs of the amplitude and the delay (α_i, τ_i) received for each of the four receivers, summing up 160 features and its respective localization x, y, z (labels).

The machine learning algorithm makes an approximation to find a position as a regression process (f). This function should be able to map input variables (α_i, τ_i), to an output variable (X_t, Y_t, Z_t), that is the target position of a sample.

$$[X_t, Y_t, Z_t] = f(\alpha_i, \tau_i), 1 < i < 20. \quad (7)$$

Since there is not too much variance in Z data, we have just selected X and Y for our simulations.

Several Machine Learning algorithms can be used to solve the proposed problem. We, here, focus on meta algorithms, based on ensemble methods to try to cover different areas of the problem using different techniques, which, through a voting scheme, can better solve the problem. 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 were introduced by [**?**] who based on earlier work described in [**?**] and uses Decision Trees and bagging idea [**?**]. Random Forests can be used for either categorical labels (classification) or continuous labels (regression).

Bootstrap aggregating or bagging models is a method for fitting multiple versions of a prediction model and then combining them into an aggregated prediction (ensemble model) [**?**]. 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 together from the individual regressors.

$$\tilde{f}_{bag} = \tilde{f}_1(X) + \tilde{f}_2(X) + \tilde{f}_3(X) + ... + \tilde{f}_b(X) \quad (8)$$

Where X is the data for which we want to generate a prediction, $\tilde{f}_{bag}$ is the bagged prediction, and $\tilde{f}_1(X)$,$\tilde{f}_2(X)$ ... $\tilde{f}_b(X)$ 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 in parallel.

According to [**?**], boosting is a class of 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) less 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.

So, the main idea of boosting is to add new models to the ensemble sequentially and boosting 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. Fig.7 shows this approach.

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

Random Forest models most depend on the number of estimators - which is the number of trees that will be used to fit. So, we have chosen the same approach used in [**?**], where we keep most of the hyperparameters as default values and changed the number of estimators. By default, random forest trains fully grown trees, so, model size is limited by computer available memory.

Gradient Boosting models based on decision trees also most depends on the number of estimators - which is the number of trees which will be used to fit. But

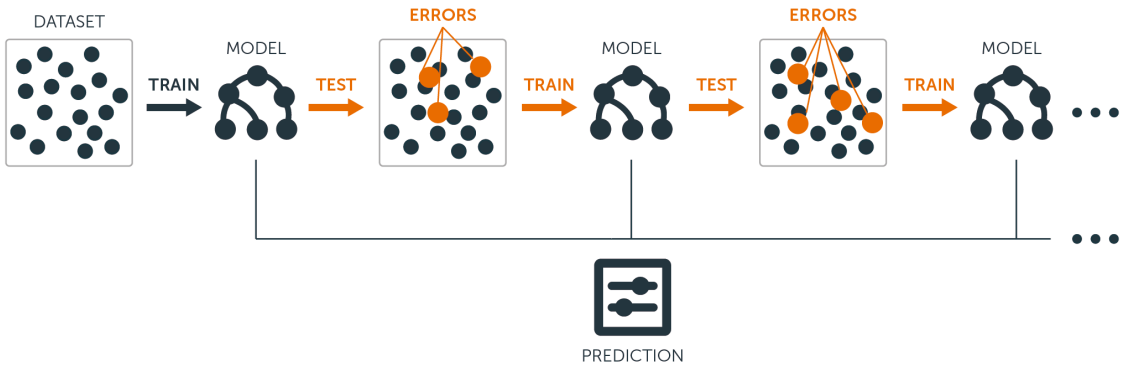

Figure 7 Ensemble sequentially in boosting. Source [?]

since it just has to create weak learners, we can limit the maximum depth of the trees, and avoid increasing maximum depth will make the model more complex and more likely to overfit.

Random Forest is especially attractive when using real-world data which is noisy, and Gradient Boosting is more sensitive to overfitting if the data is noisy.

Since both models are based on the CART algorithm for the decision tree, both solutions are compelling when dealing with missing data (mismatching). CART handles missing values either by imputation with average, either by rough average/mode, either by an averaging/mode based on proprieties.

6 Results and Discussion

Since the model generated should generalize for unseen data (real data) which do not have the same distribution as the training data, it was decided to use emitter dataset in the validation process to allow the adjustment of hyper-parameters. Fig.8 presents the training process for the machine learning algorithm. First, a set of hyper-parameters of the model was selected. Simulation Dataset was divided into Training (80% of the samples) and Testing (20% of the samples) data. Next, the Training dataset was again divided into a new Training Dataset (80% of the samples) and Validation Dataset (20% of the samples). The new training dataset was used to train the models.The Validation and Emitter Data-sets were used to obtain model performance and adjust hyper-parameter to obtain better performance when applied in validation and emitter dataset. The Emitter Dataset was used to set the value of hyper-parameters and the Validation Dataset to estimate model error for simulation data.

The loss function used to measure the model performance was the mean squared error (MSE) between the actual emitter localization and the predicted model localization. Equation 9 presents the formula for MSE:

$$MSE = \frac{1}{n}\Sigma_{i=1}^{n}\left(y_i - \tilde{y}_i\right)^2 \tag{9}$$

Where y_i is the real localization of the emitter i and $\tilde{y}_i$ is the predicted localization of the emitter i.

Finally, with the final model defined, test dataset was used the evaluate final model error (simulation data). Then this final model was applied to the Target Dataset, and the mean squared error for each sample was obtained.

6.1 Hyperparameters Selection

When selecting the number of estimators for random forest model, a range from 1 to 40 was tried. The best result (minimum mean euclidean distance error) were obtained using 24 estimators and the euclidean distance error in validation was 1.76 meters and the euclidean distance error in emitter dataset was 13.40 meters. Fig.9 presents the variation of the median euclidean distance error according to the number of estimators in the random forest model for validation and emitter dataset.

6.2 Analysis

After the simulation with experimental setup, we can come to the following observations:

- It was clear that RF used much more of Computer Power resources; on the other hand, it allows parallel execution because the internal structure of the decision trees is independent. It should be noted that a Random Forest model with 24 estimators used 43Gb RAM, but it took less than

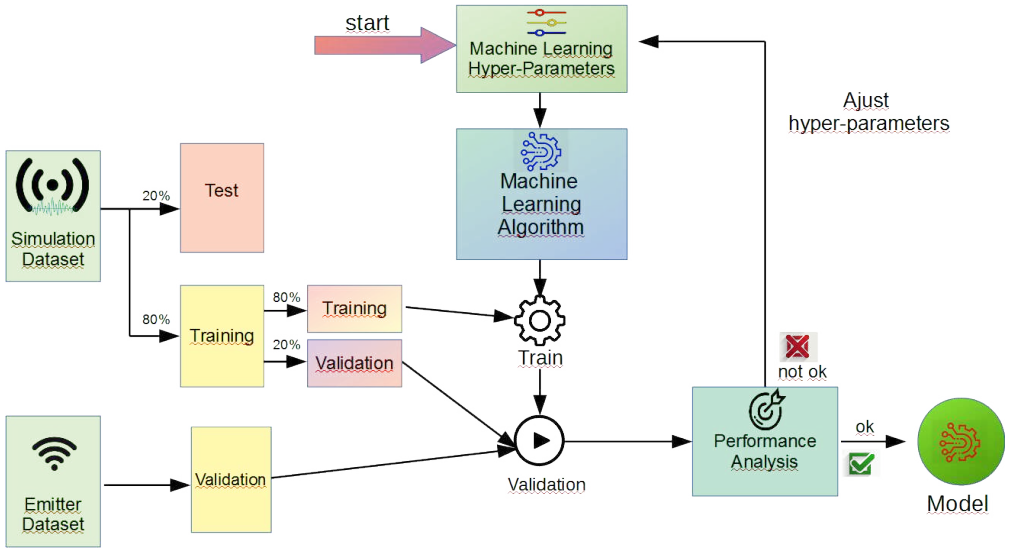

Figure 8 Training process using Simulation Dataset and Emitter Dataset.

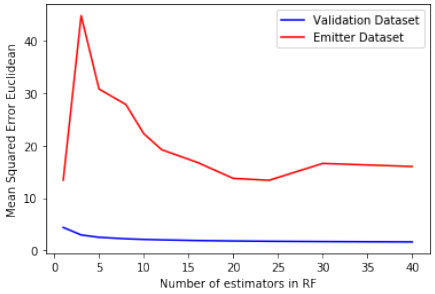

Figure 9 Median Euclidean Distance Error Variation for Validation dataset and Emitter Dataset according to number of estimators in random forest model

10 minutes to be created using 12 threads (number of n_jobs in sci-kit learn library).

- For Gradient Boosting it was used an open-source library called XGBoost where decision trees were used as weak learners. This XGBoost model mostly depends on the number of estimators - number of trees which will be used to fit - and maximum depth - which is the maximum depth of the trees. We have worked out a grid search where the number of estimators is in the range from 100 to 300, and maximum depth are in range 7 to 21. Best results - which means minimum mean euclidean distance error for emitter dataset - were obtained using 280 estimators and 15 maximum depth. All other parameters were set to the default value.
- It was observed that XGBoost used less RAM but it takes more time to be generated, in that case the XGBoost model with 280 estimators and 15 maximum depth used only 60Mb RAM, but it took more than an hour to be generated.

6.3 Noise Experiment

For noise experiment, we have added Gaussian noise in all 160 features with 0 mean and *level*.variance of each feature of all samples (2973) from the Target Dataset. The range of the *level* was from 0.1 to 1.0, with an increment of 0.05, and from 1 to 9, with an increment of 1. Fig. 10 shows the Noise Experiment Process.

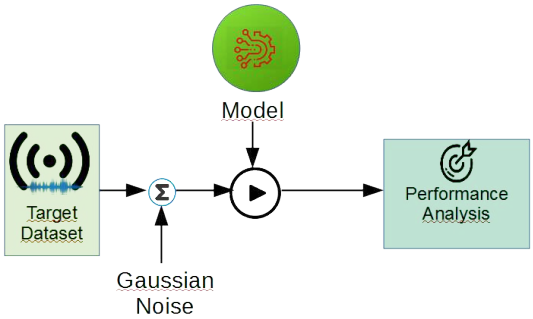

Figure 10 Noise Experiment Process.

Both models suffer from noise and Random Forest got better results when the level of the noise is low, and XGBoost got better results when noise is above 3. So, the first finding is: in a noisy environment give

preference to the XGBoost model and in a low noise environment give preference to the Random Forest model

The Fig.11 shows the results of this experiment. The main red line is the mean Euclidean distance error between real and predicted positions of all 2973 samples using Random Forest. Assuming that μ, as mean and σ, as the standard deviation of the errors in the positioning, The red area define $\mu - \sigma$ and $\mu + \sigma$ limit. The main green line is the mean Euclidean distance error between real and predicted positions of all 2973 samples using XGBoost. The green area defines $\mu - \sigma$ and $\mu + \sigma$ limit.

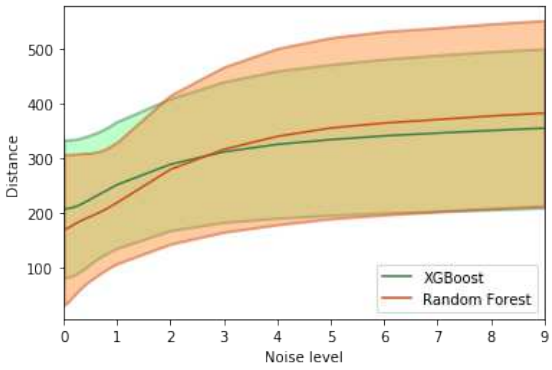

Figure 11 Noise Effects over Euclidean Distance Error for Random Forest model and XGBoost model.

From the Fig.11, we may notice that standard deviation from euclidean distance error are high, which means there lot of position which suffers a lot from noise. The main cause of the high values is that both models were generated using simulation data and executed on real data (another data distribution). However, several locations were obtained with error less than 50 meters[1].

The Fig.12 presents the performance for both Random Forest and XGBoost model when the level of noise is set to 3. The yellow points are all Target Dataset. Red points are Position Estimation where Random Forest error was less than 50 meters, and Blue points are Position Estimation where XGBoost error was less than 50 meters.

The second finding is: when considering all target samples where the error is less than 50 meters, some Position Estimation are better modelled by Random Forest, and XGBoost better models some other localization. There is no clear zone where RF is better than XGBoost or vice-versa.

[1]In terms of the localization task, a 50-meters error is considered by the authors as satisfactory.

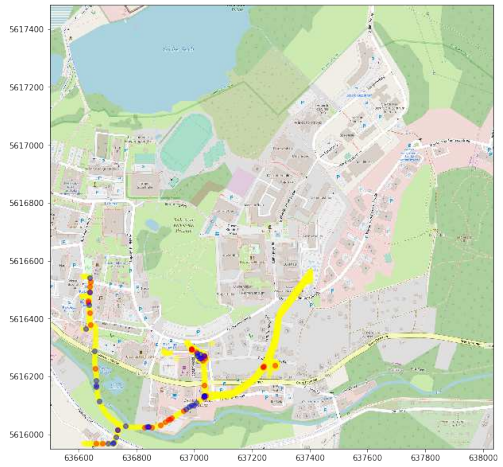

Figure 12 Position estimation where error was less than 50 meters for Random Forest (blue) and XGBoost (red) for noise experiment.

6.4 Mismatching Experiment

In the mismatching experiment, a subset of features is nullified, and the performance of the model is obtained. The selected noise level for this experiment was set to 3 - the value in which both models have comparable errors when using emitter dataset. Fig.13 presents this process.

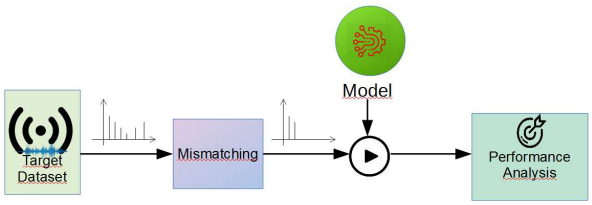

Figure 13 Mismatching Experiment Process.

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 pair of features, where n is in the range from 1 to 20. Fig.14 presents the results: again, the main red line is the mean Euclidean distance error between real and predicted positions of all 2973 samples using Random Forest. The red area defines $\mu - \sigma$ and $\mu + \sigma$ limit. The main green line is the mean Euclidean distance error between real and predicted positions of all 2973 samples using XGBoost. The green area are defines $\mu - \sigma$ and $\mu + \sigma$ limit.

The first finding is that both models did a great job in dealing with mismatching: as can be observed, none of the models suffered significant variations in the average of errors with the variation of the number of

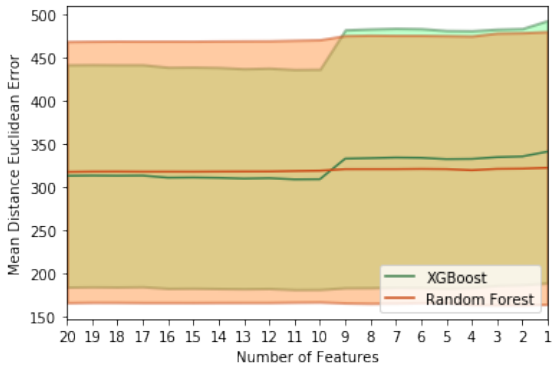

Figure 14 Mismatching effects over Euclidean Distance Error for Random Forest model and XGBoost model.

available characteristics. The reason is that many of the features from simulation dataset are already zero.

Fig.15 presents the performance for both Random Forest and XGBoost model when the number of features is set to 1 (one amplitude and one delay), which is the worst case scenario. All yellow points are from the Target Dataset. Red points are positions where Random Forest error was less than 50 meters, and Blue points are location where XGBoost error was less than 50 meters.

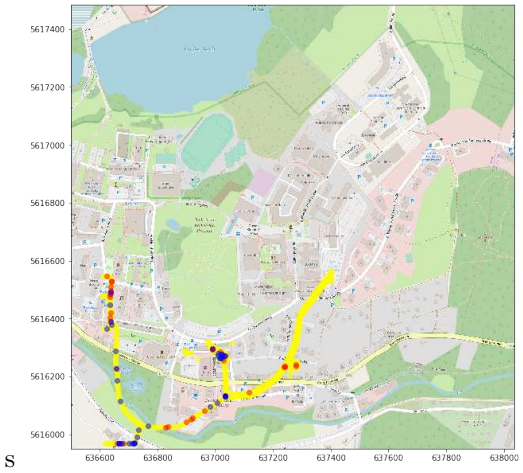

Figure 15 Position estimation where error was less than 50 meters for Random Forest (blue) and XGBoost (red) for mismatching experiment.

The second finding is similar to the noise experiment: when considering all target samples where the error is less than 50 meters, some localization are better modelled by Random Forest, and XGBoost better models some other position estimation. Again, there is no clear domain where RF is better than XGBoost or vice-versa.

7 Conclusion

The paper showed a comparison of the machine learning models for enhancing a kernel-based machine learning localization scheme based on TDOA fingerprinting. In fact, the Position systems using machine learning enhancement based on simulated multipath information face the problem of the difference between the real measurements and the actual channel conditions that can differ from the synthetic data.

The results presented here can serve as guidelines in similar problems to adjust the number of estimators or even to help the definition of which machine learning implementation is more suitable. Localization Fingerprints methods can deal with measurement errors by continuously improving the estimation based on the measurements available in the area of interests. These features 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 improve the overall radio-frequency system performance. In this sense, the scenario features, summed up in the cartographic maps, can give the signal processing an extra dimension of work. We also show that our method is very insensitive and measurement errors from the reference nodes that l are randomly chosen from emitter tracks in the area of interests.

For next steps in the research, the authors would like to consider the use of more features either the signal and the scenario, trying to record more multipath information and patterns to build a data fusion engine with the cartographic database and signal processing. The authors also plan to use optimization tools to be used in Sensor Management and to apply the machine learning approach to deal with the rough information produced by multipath reflection in the TDOA system deployment.

List of abbreviations

Following abbreviations are used in this manuscript:
AOA: Angle of Arrival
CART: Classification and Regression Trees
CRB: Cramer-Rao Bound
CSI: Channel State Information
FDOA: Frequency Difference of Arrival
KNN: K-nearest Neighbor
LTE: Long-term Evolution
ML: Machine Learning
NLOS: Non-Line-of-sight
RF: Random Forest
RSS: Received Signal Strength
RT: Ray-tracing
TDOA: Time-difference of Arrival

VCC: Volume Cross-correlation
WSN: Wireless Sensor Network
XGBoost: X-Gradient Boosting.

Declarations

Availability of data and materials
The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Competing Interest
The authors declare that they have no competing interests.

Funding
There was no funding supporting this Research.

Author's contributions
All authors take part in the discussion of the work described in this and should be considered co-correspondent authors. M.N.S. 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. J.A.A.Jr and R.S.T. 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. R.S. conducted a research in applying machine learning for position estimation, developed the machine learning architecture and applied it into available multipath datasets. J.C.D.conducted a research, investigation process and mentoring in applying machine learning for position estimation, beyond analysis of the results.

Acknowledgements
We thank the editor and anonymous reviewers for their helpful comments and valuable suggestions. I would like to acknowledge all our team members. They contributed equally to this work.

Figures

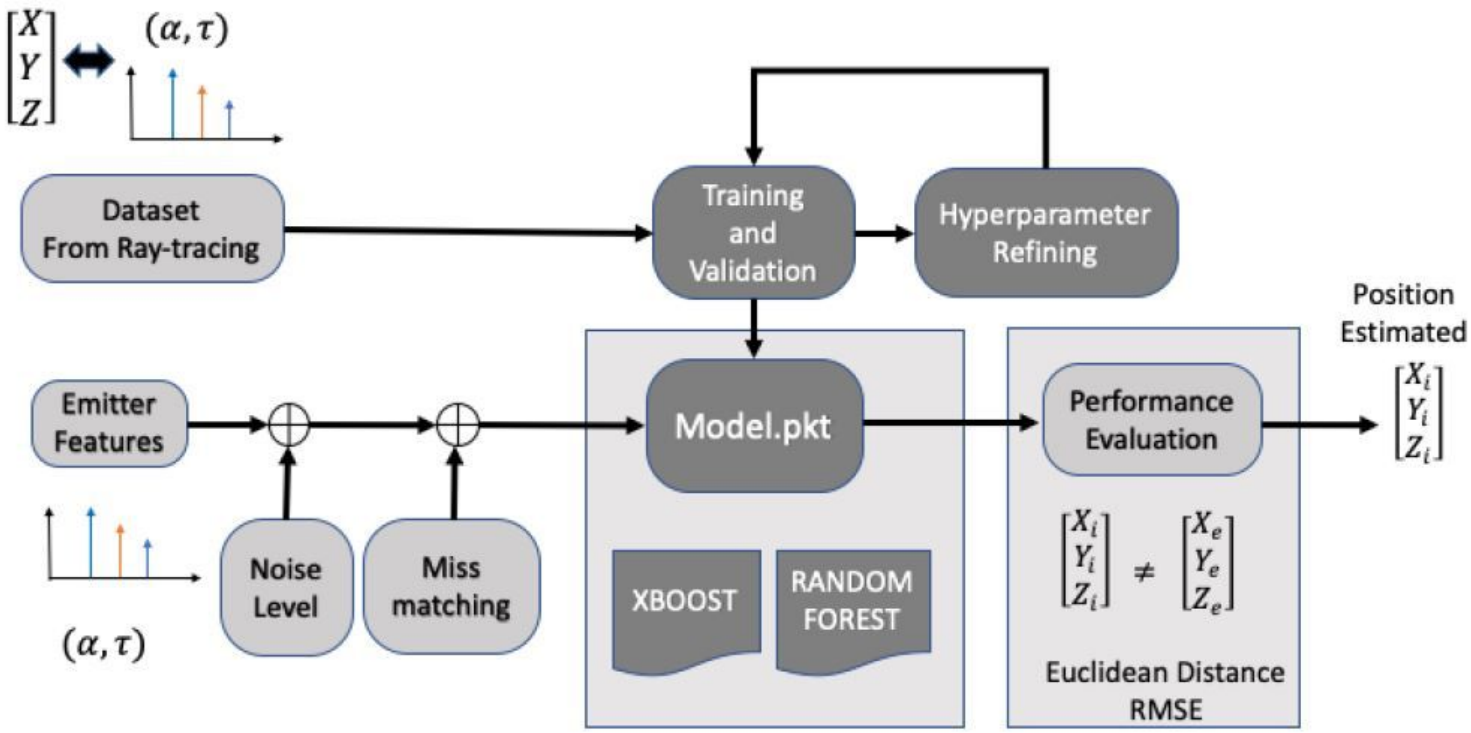

Figure 1

Overview of Proposed Approach

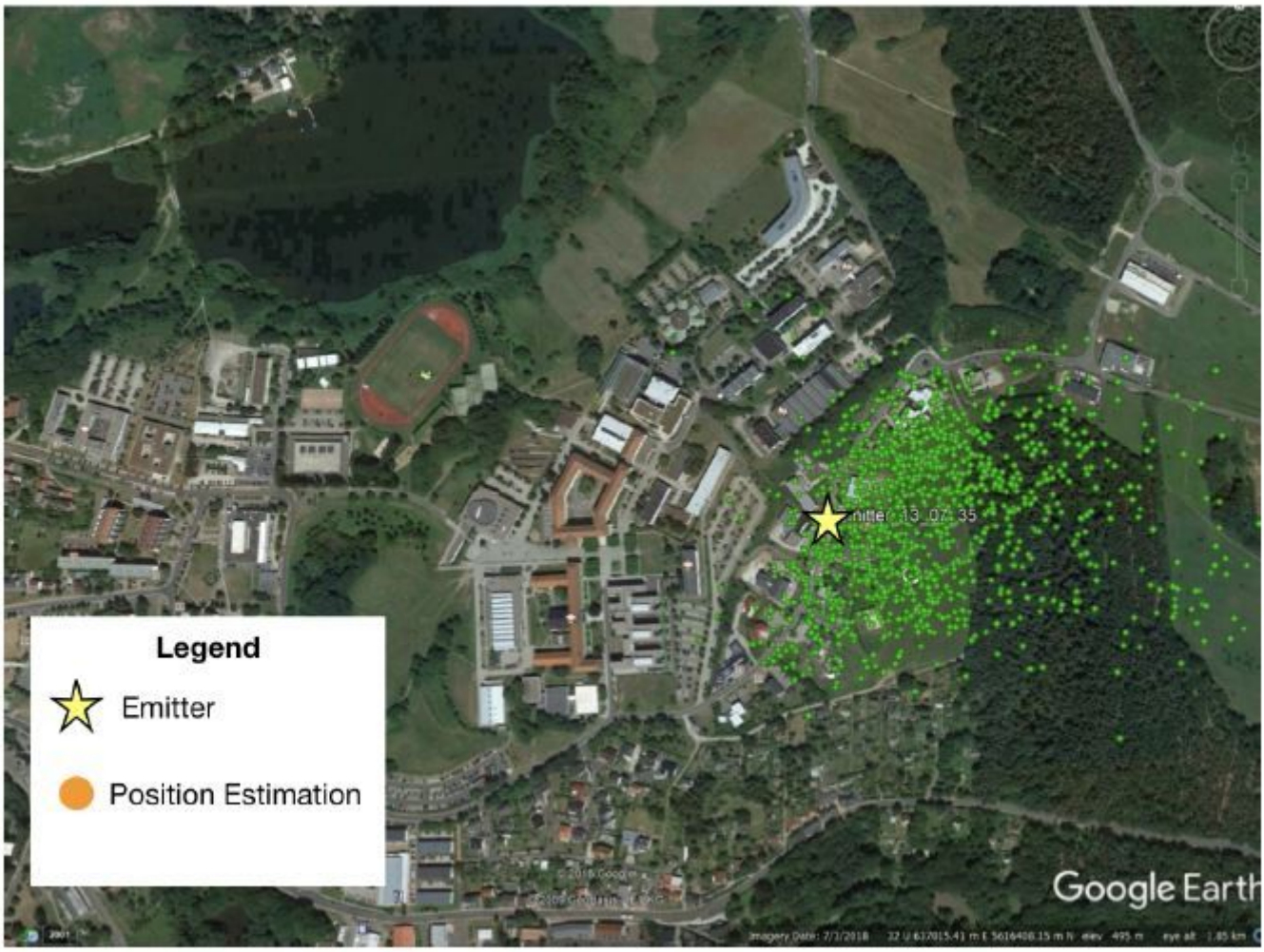

Figure 2

Performance of Localization System in Outdoor Scenario. Note: The designations employed and the presentation of the material on this map do not imply the expression of any opinion whatsoever on the part of Research Square concerning the legal status of any country, territory, city or area or of its authorities, or concerning the delimitation of its frontiers or boundaries. This map has been provided by the authors.

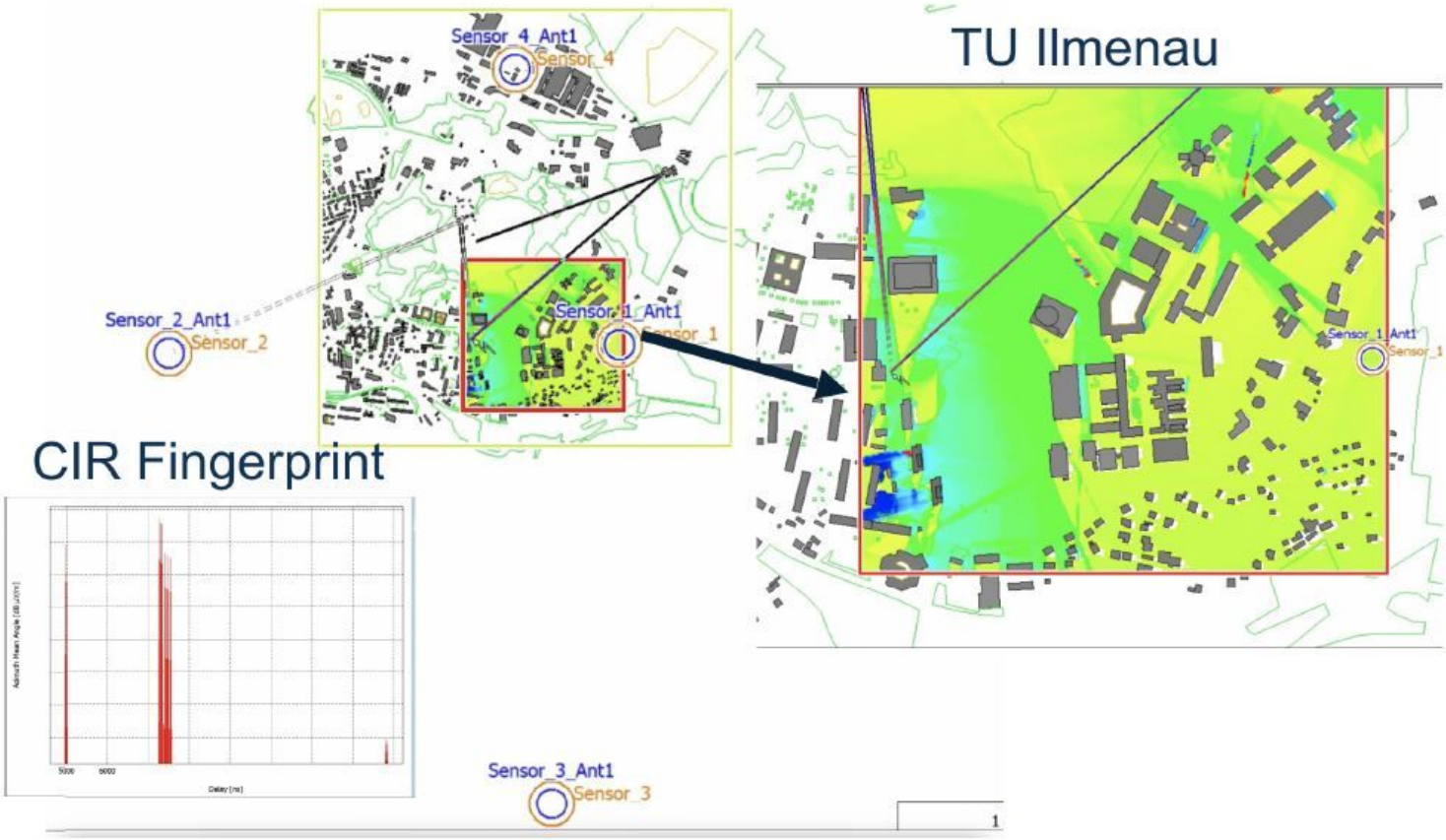

Figure 3

Buildings in Ray Tracing simulation. Note: The designations employed and the presentation of the material on this map do not imply the expression of any opinion whatsoever on the part of Research Square concerning the legal status of any country, territory, city or area or of its authorities, or concerning the delimitation of its frontiers or boundaries. This map has been provided by the authors.

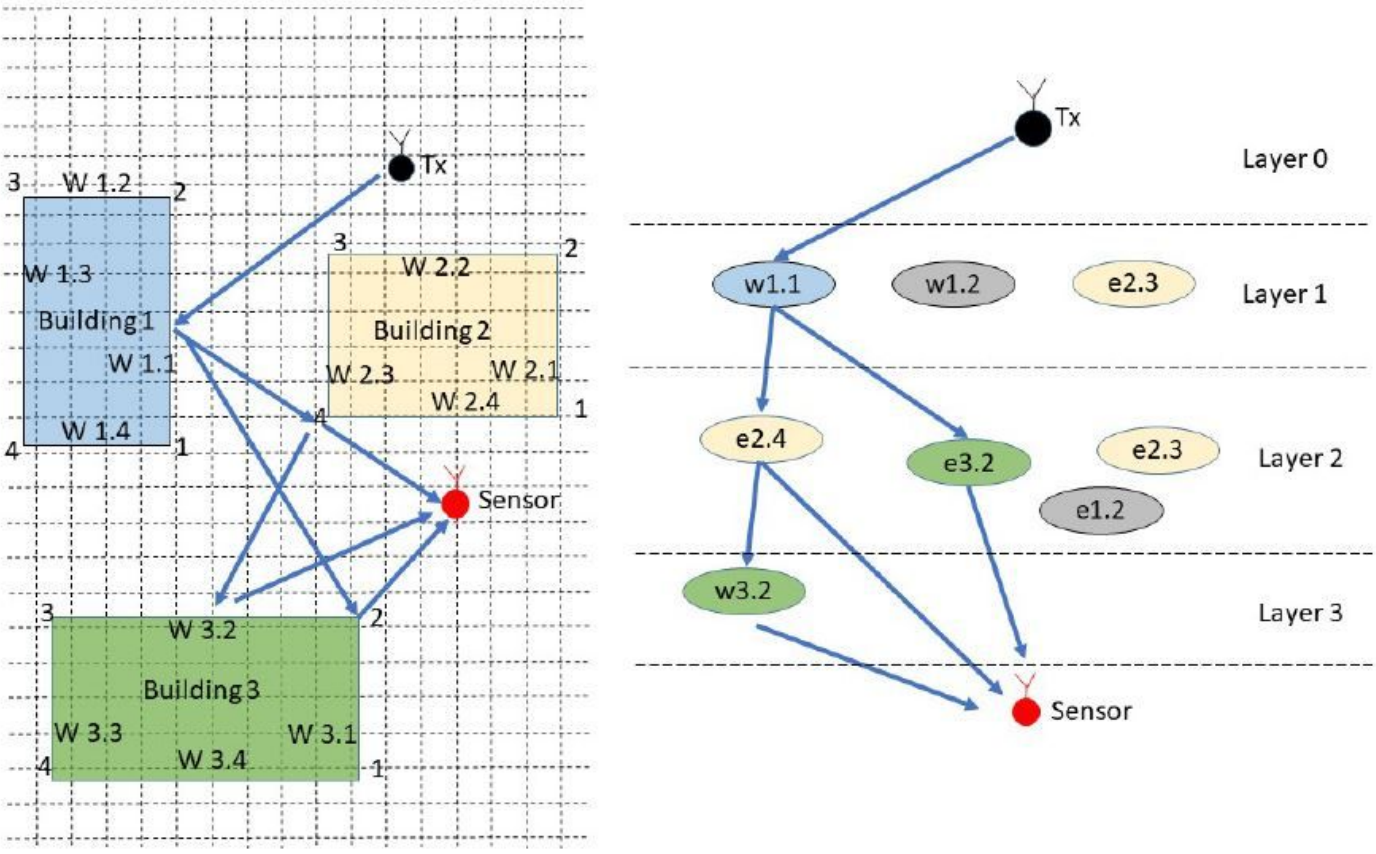

Figure 4

Wall and Edges form the Buildings in Ray Tracing simulation [?]

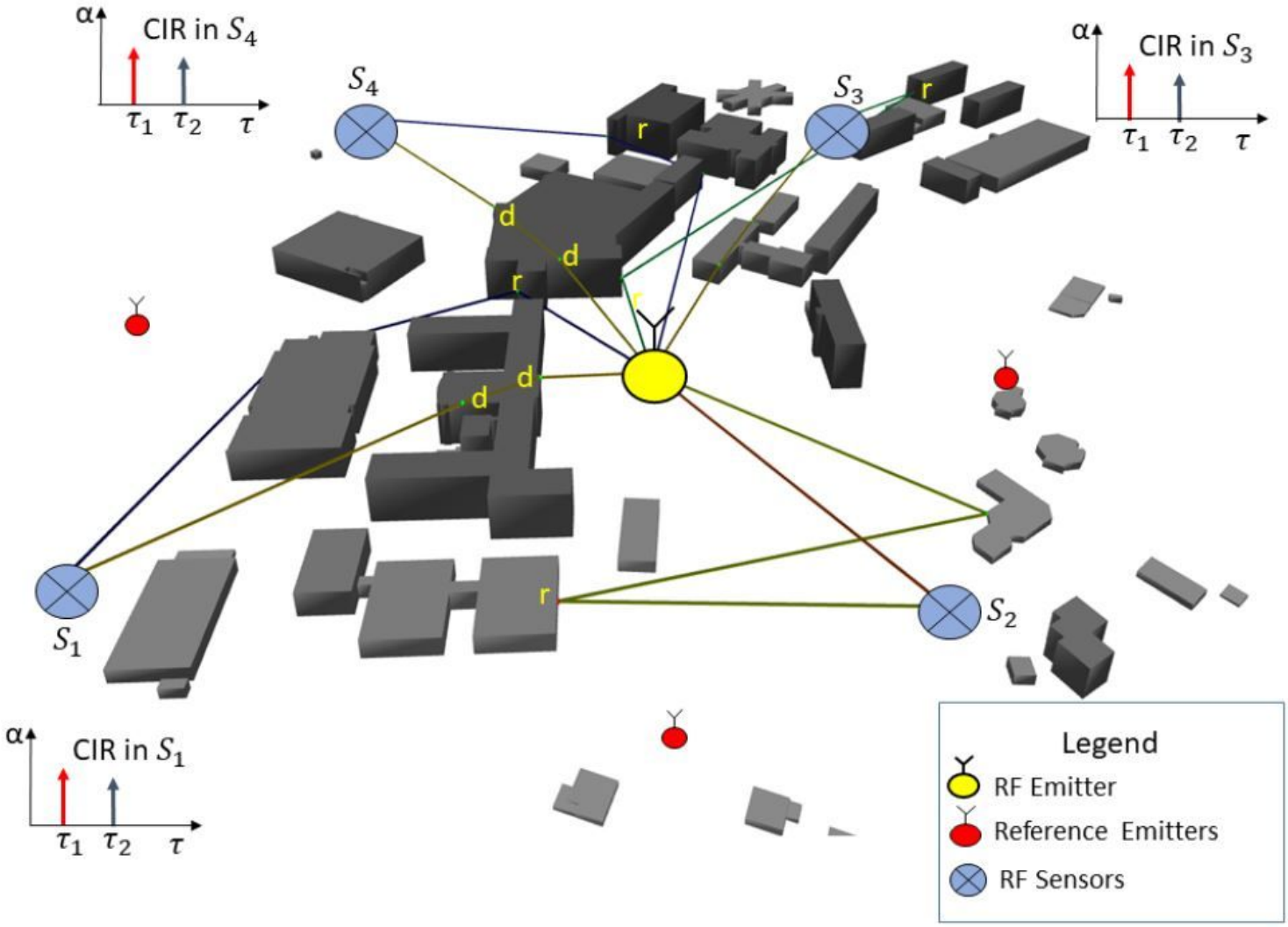

Figure 5

Extraction of CIR fingerprints using Ray Tracing

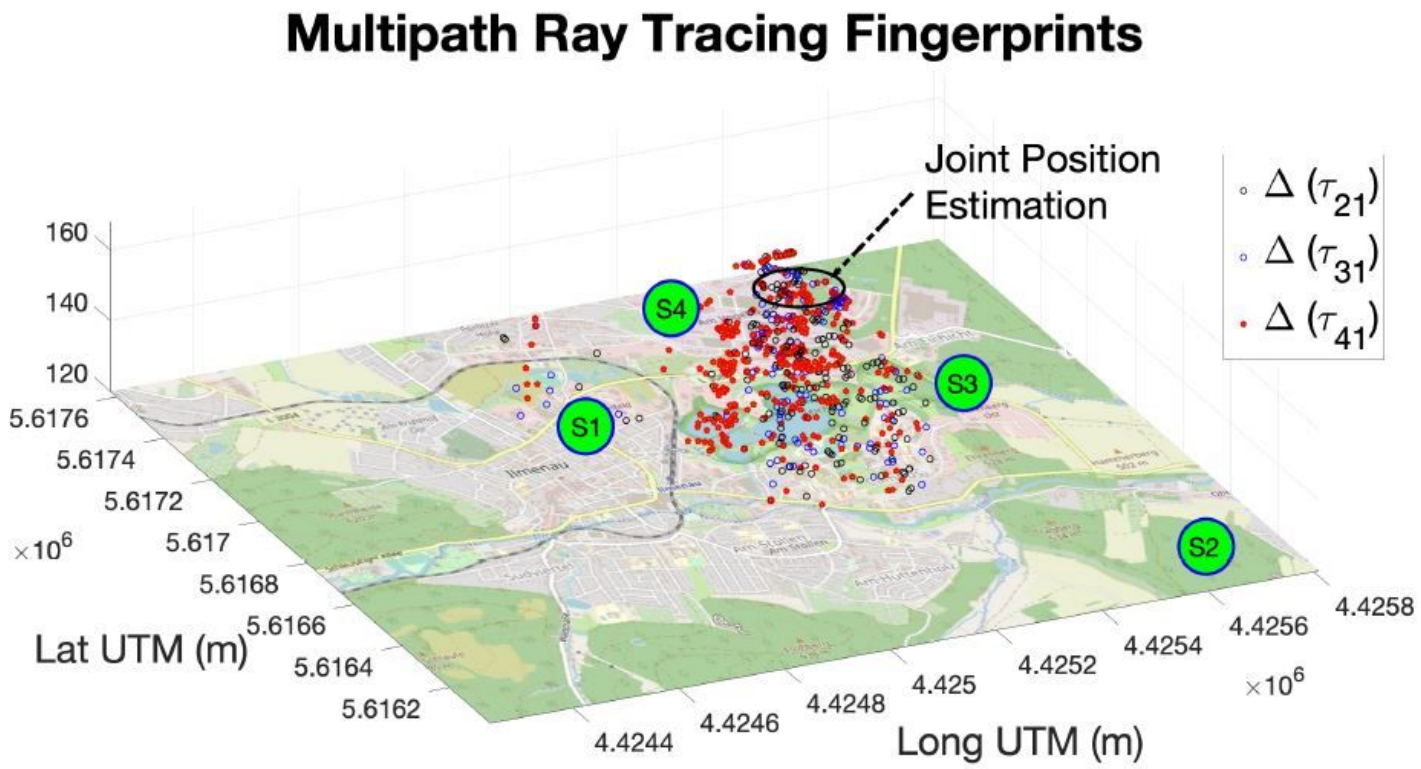

Figure 6

Ray Tracing Multipath Dataset, from [?] Note: The designations employed and the presentation of the material on this map do not imply the expression of any opinion whatsoever on the part of Research Square concerning the legal status of any country, territory, city or area or of its authorities, or concerning the delimitation of its frontiers or boundaries. This map has been provided by the authors.

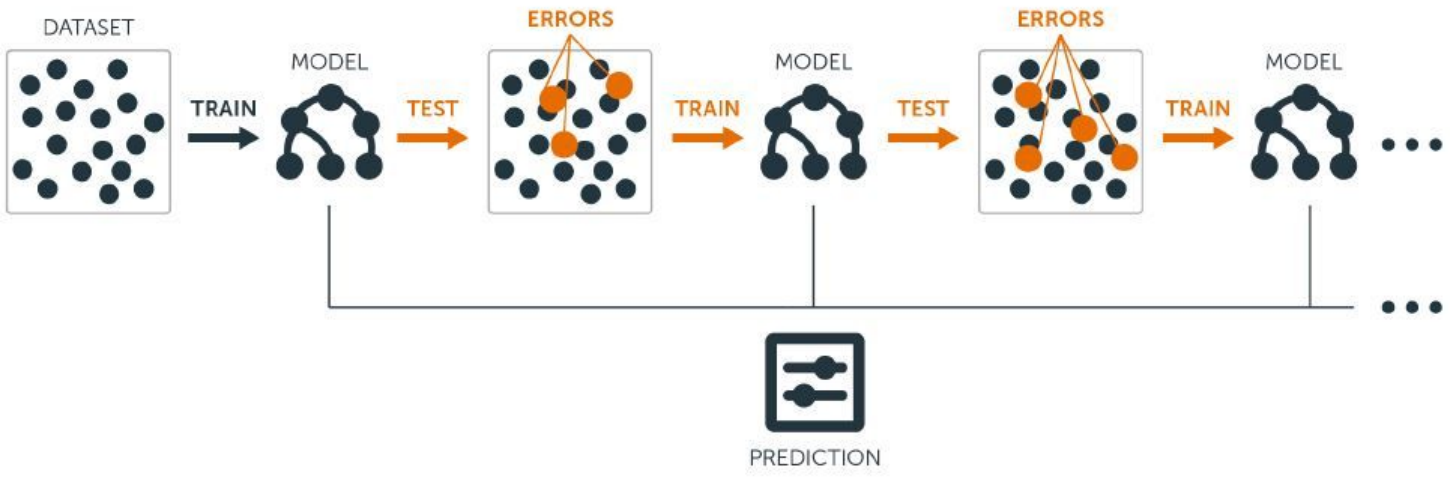

Figure 7

Ensemble sequentially in boosting. Source [?]

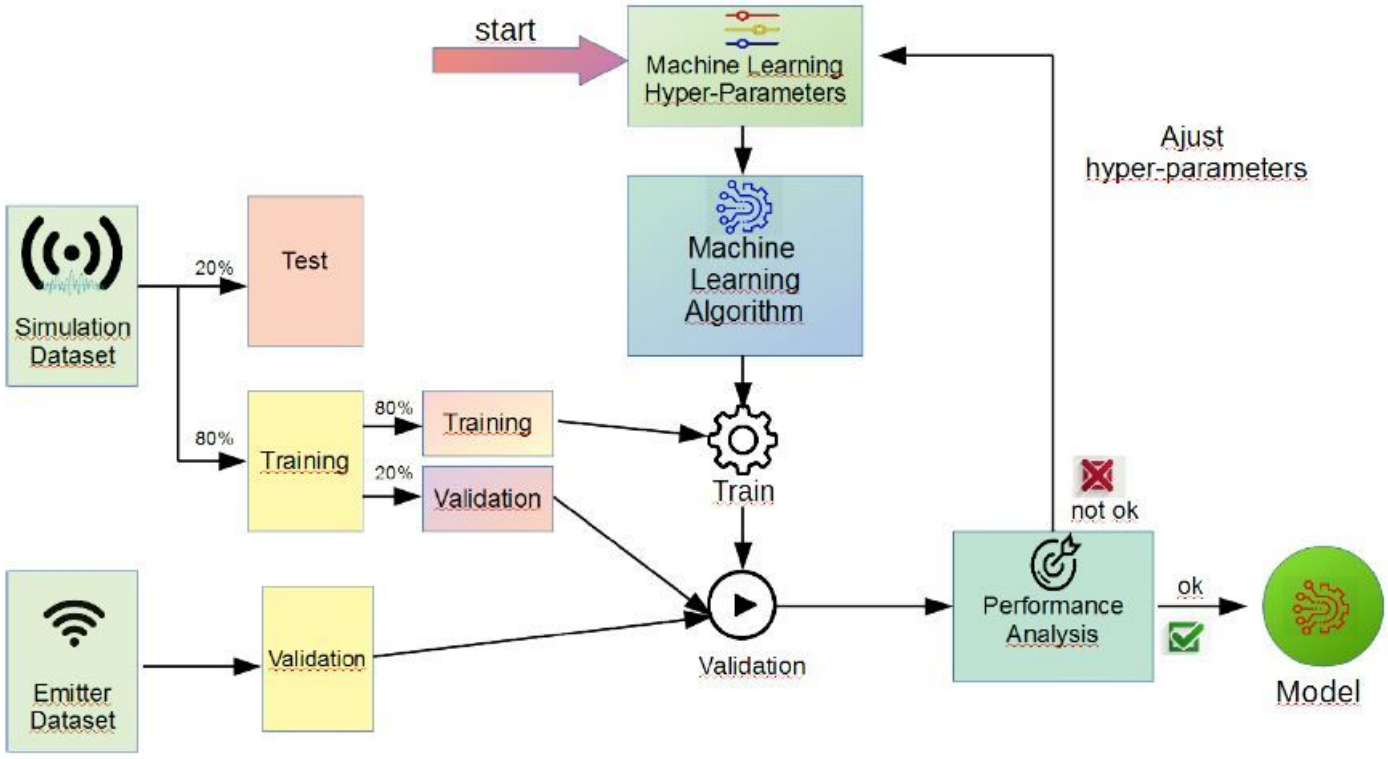

Figure 8

Training process using Simulation Dataset and Emitter Dataset.

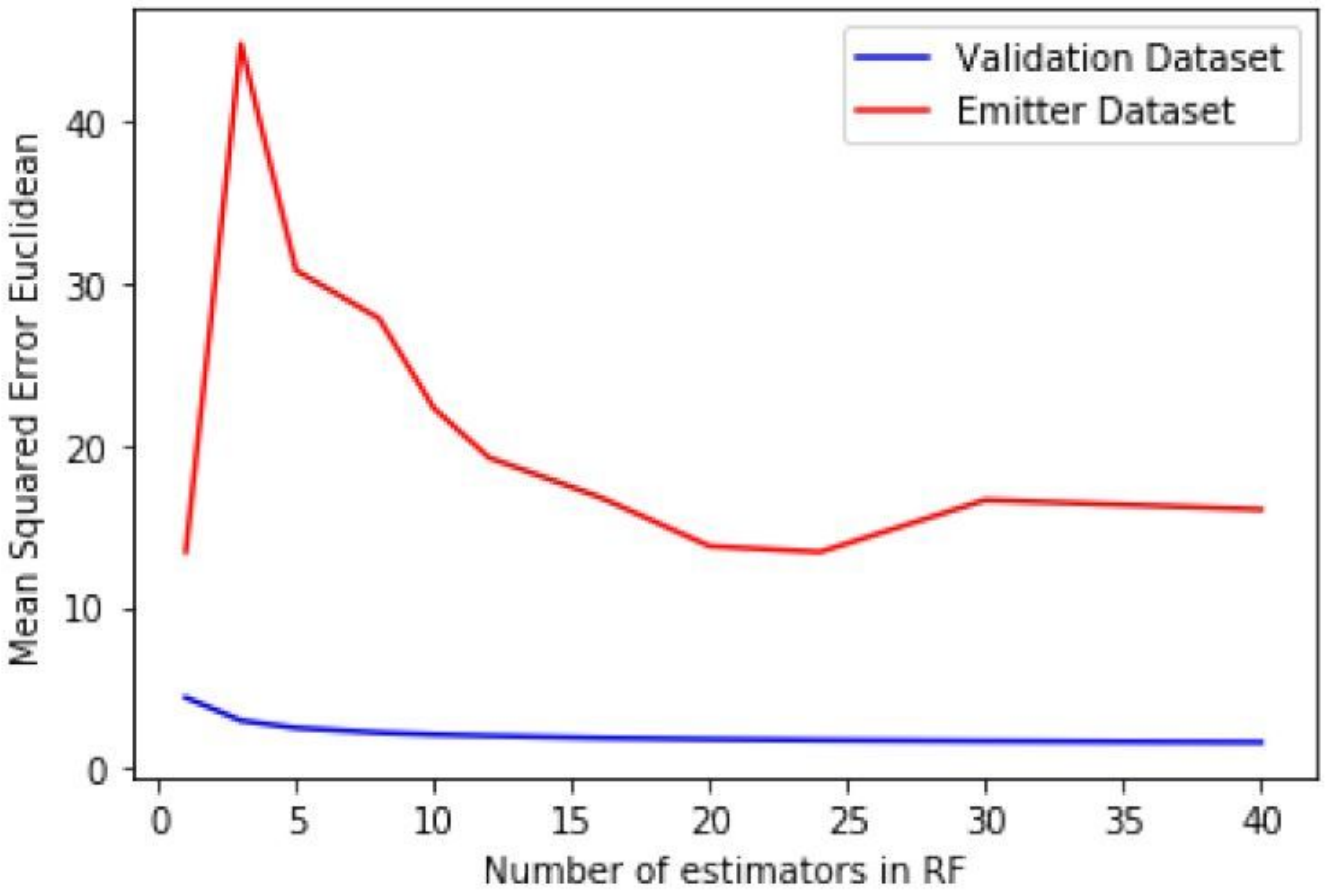

Figure 9

Median Euclidean Distance Error Variation for Validation dataset and Emitter Dataset according to number of estimators in random forest model

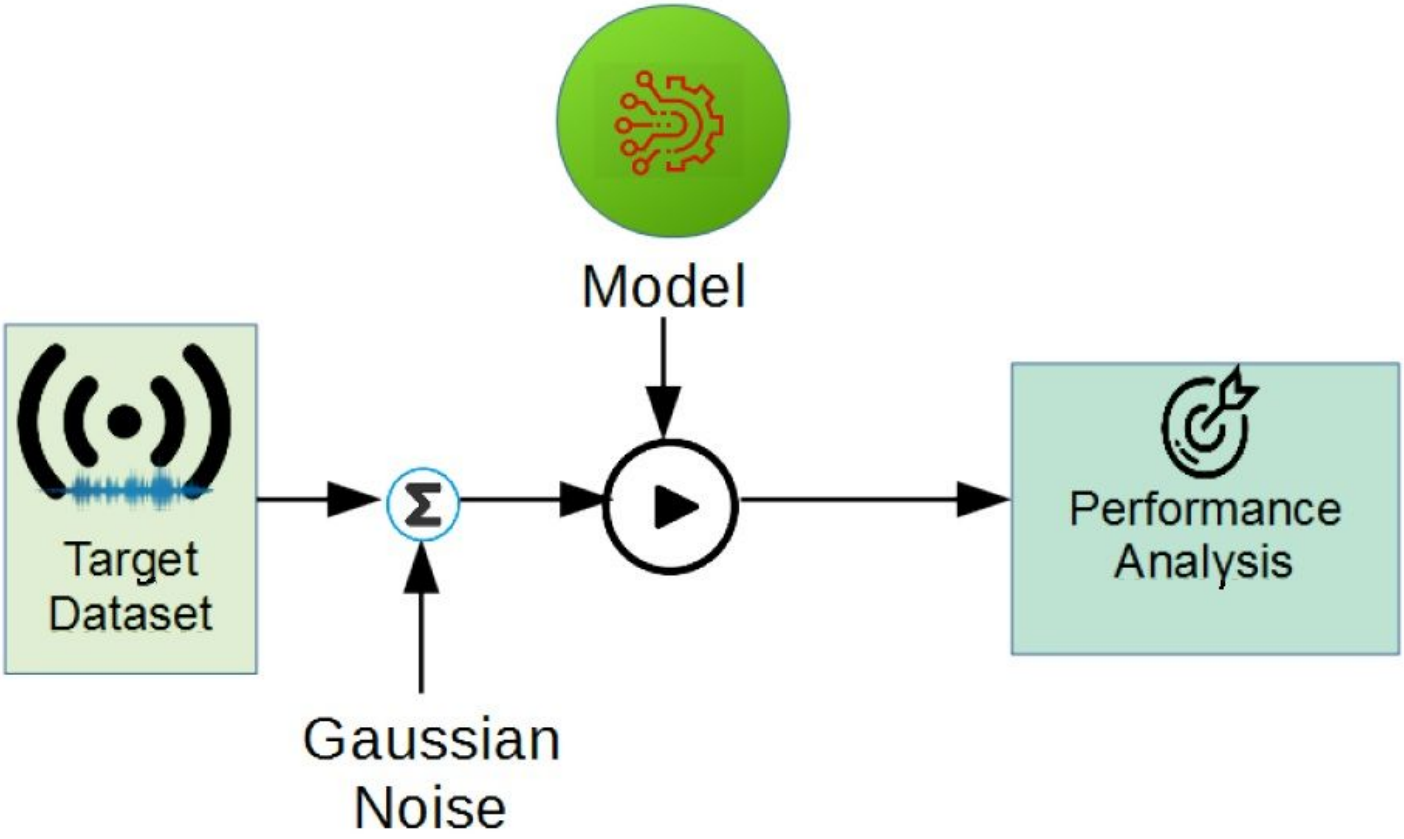

Figure 10

Noise Experiment Process.

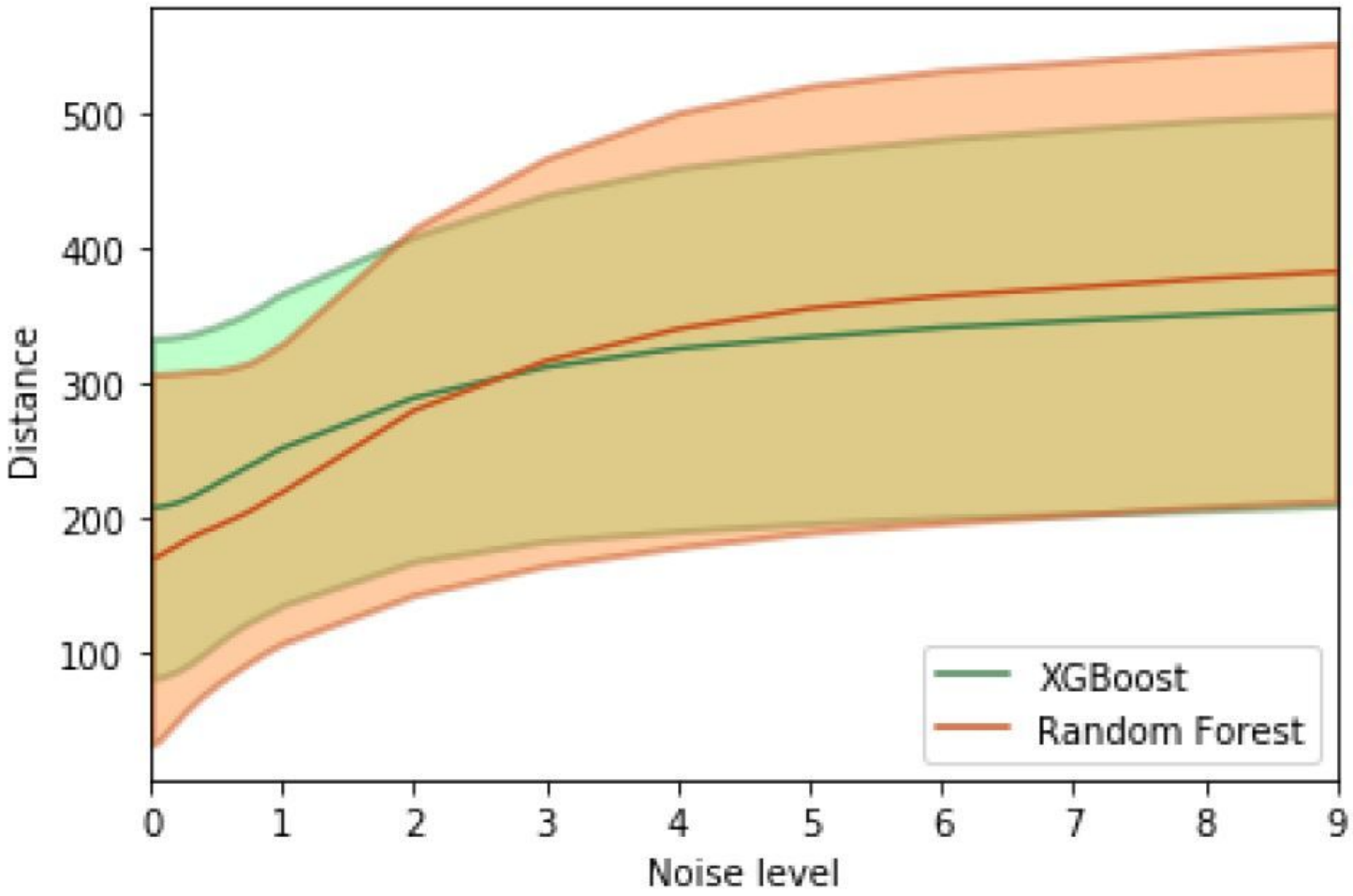

Figure 11

Noise Effects over Euclidean Distance Error for Random Forest model and XGBoost model.

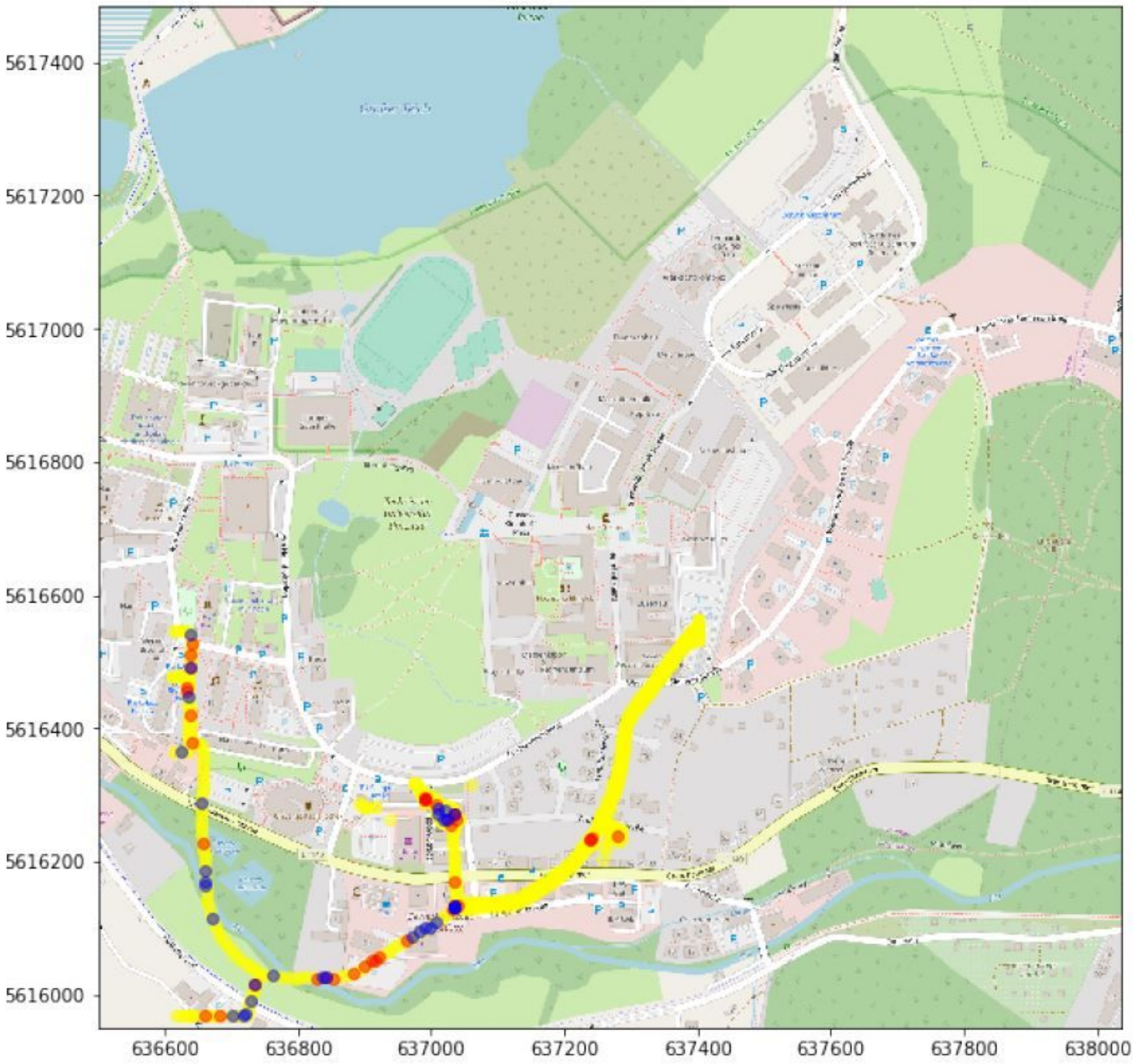

Figure 12

Position estimation where error was less than 50 meters for Random Forest (blue) and XGBoost (red) for noise experiment. Note: The designations employed and the presentation of the material on this map do not imply the expression of any opinion whatsoever on the part of Research Square concerning the legal status of any country, territory, city or area or of its authorities, or concerning the delimitation of its frontiers or boundaries. This map has been provided by the authors.

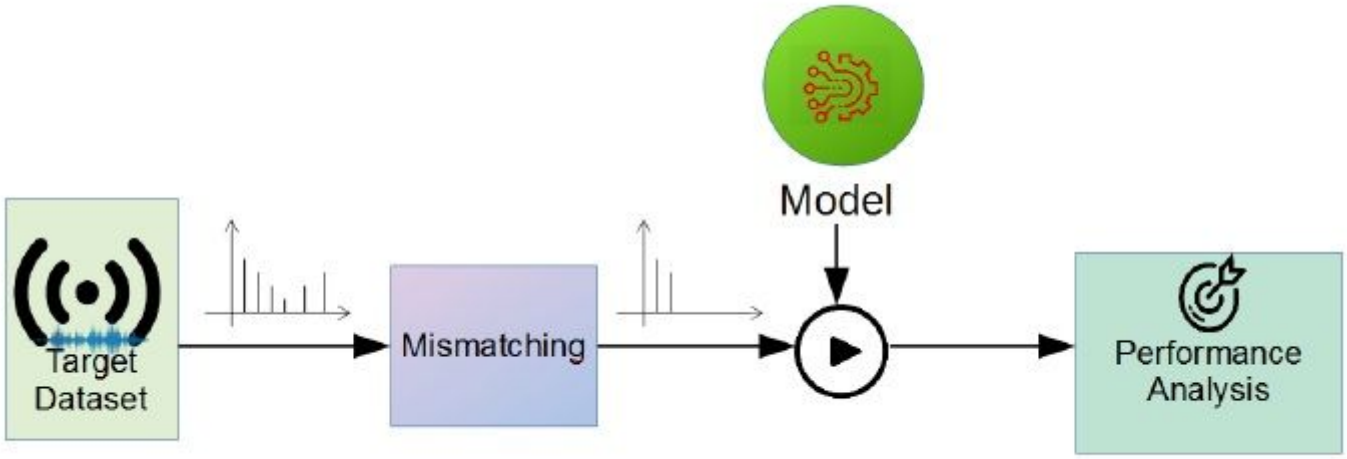

Figure 13

Mismatching Experiment Process.

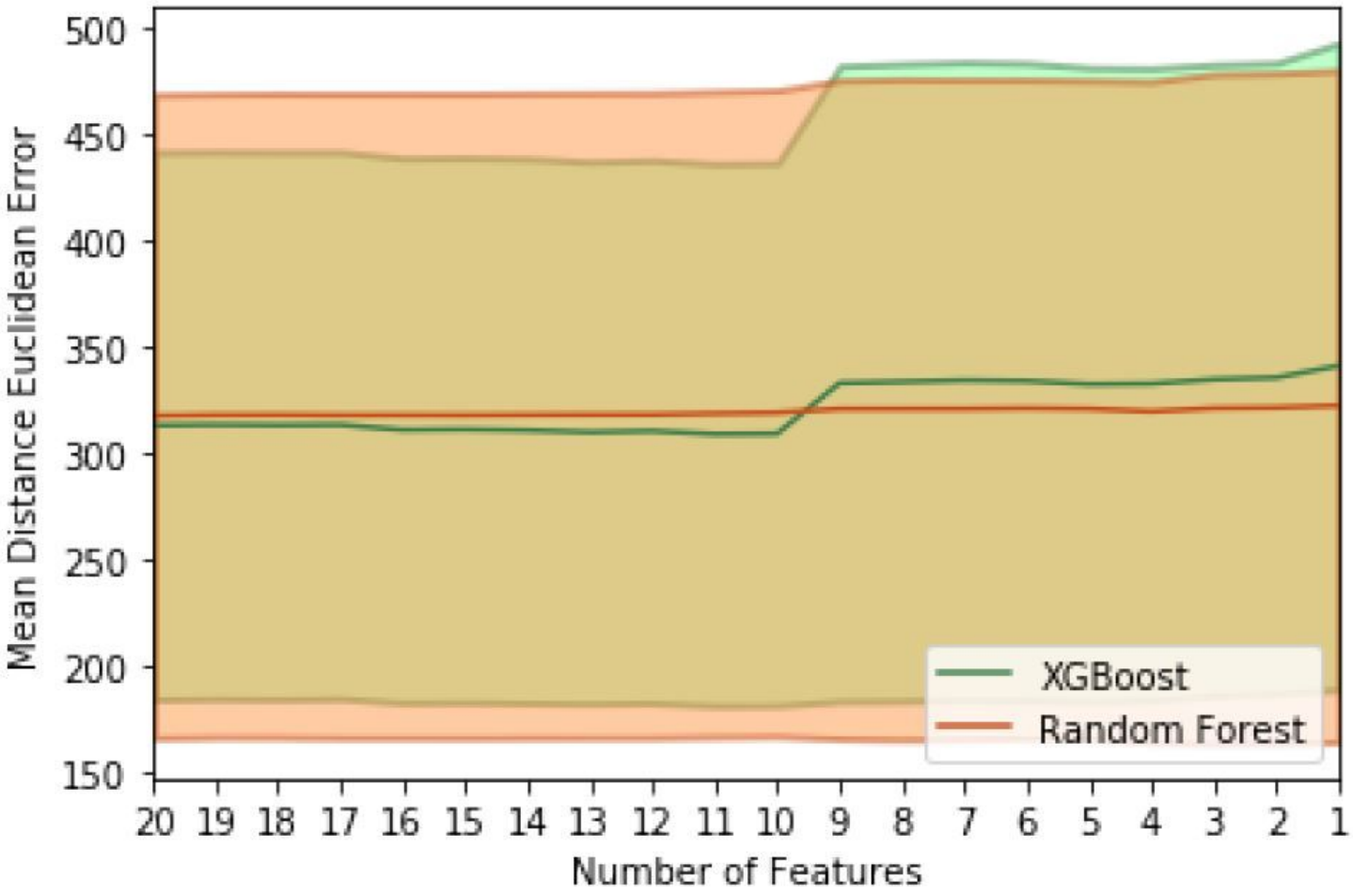

Figure 14

Mismatching effects over Euclidean Distance Error for Random Forest model and XGBoost model. Note: The designations employed and the presentation of the material on this map do not imply the expression of any opinion whatsoever on the part of Research Square concerning the legal status of any country, territory, city or area or of its authorities, or concerning the delimitation of its frontiers or boundaries. This map has been provided by the authors.

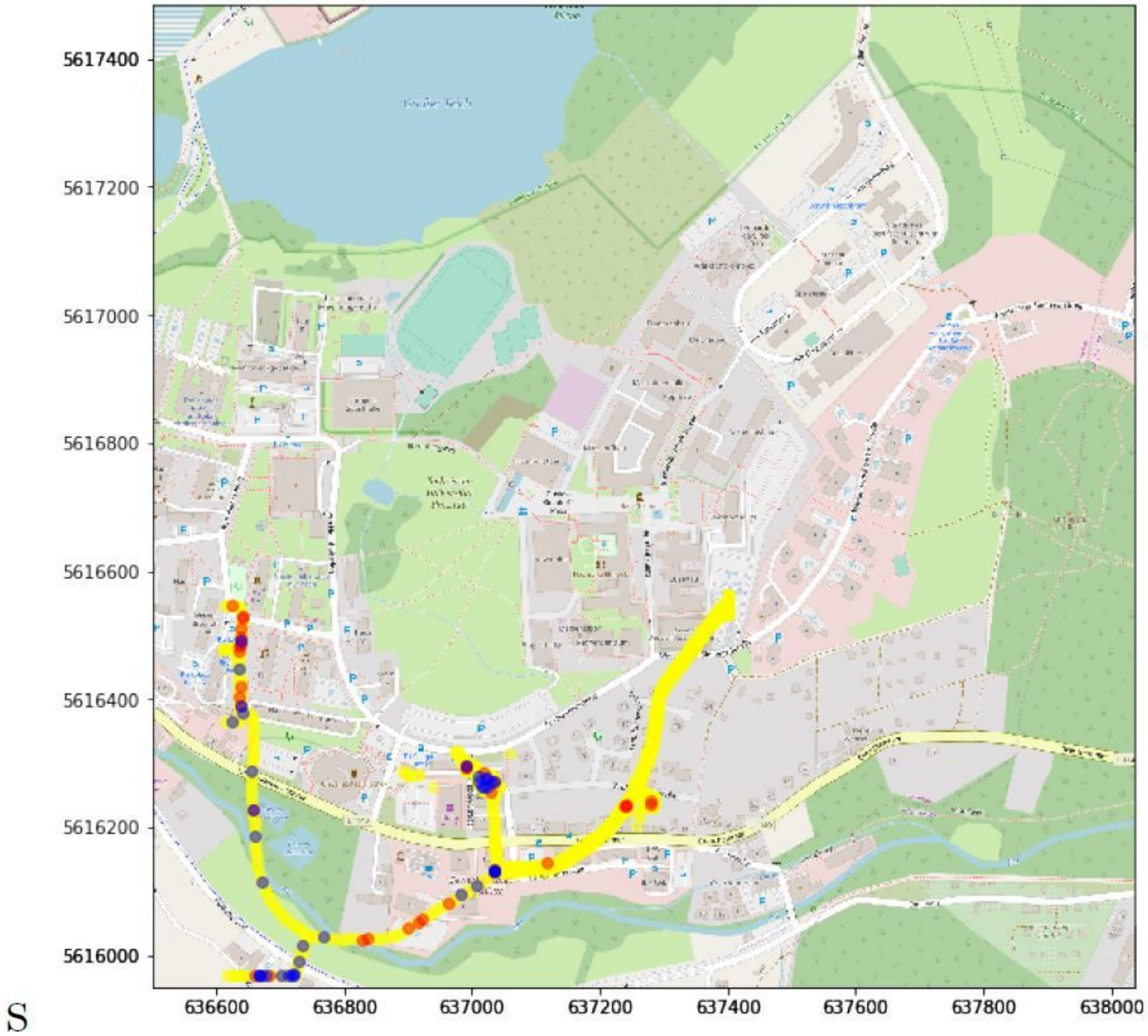

Figure 15

Position estimation where error was less than 50 meters for Random Forest (blue) and XGBoost (red) for mismatching experiment. Note: The designations employed and the presentation of the material on this map do not imply the expression of any opinion whatsoever on the part of Research Square concerning the legal status of any country, territory, city or area or of its authorities, or concerning the delimitation of its frontiers or boundaries. This map has been provided by the authors.