

# AI-based Localization: Learning from Synthetic Data from a Genetically Guided Digital Twin

J. Beysens, S. Narduzzi, L. Bergamini

*Synthetic datasets for training a machine learning model is a promising approach to reduce the need for real data. However, the simulated data should be representative of the real data. As a consequence, the tuning of the simulation parameters is critical to reproduce the real environment in the most accurate way. At CSEM we design and evaluate a genetic algorithm (GA) that proposes the optimal parameters for the network simulator.*

Indoor localization is a complex problem, for which many approaches and solutions are proposed. Although new technologies like Angle of Arrival or Time of Flight promise improved precision in location estimation, the received signal strength indicator (RSSI) remains a competitive alternative that is easier to deploy at lower cost. In a previous work [1], we studied how using machine learning (ML) can improve the estimation accuracy of an emitter using RSSI. However, ML requires a large amount of labeled data to properly train it, making it unfeasible for large-scale realistic scenarios. We demonstrated that it is possible to generate usable synthetic datasets by representing the deployment area in a simulator, which are sufficient to train the ML model and in this work we show the benefits from applying GA for the optimization of the simulation parameters.

Training an ML model to localize objects in the deployment environment necessitates an accurate virtual representation of it, as any mismatch between the real and virtual environments introduces a bias and has a strong impact on the performance of the ML model. The network simulator we used (OmNET++ with the INET framework) offers a wide range of parameters that can be tuned to accurately reproduce the reality. Further, it offers the possibility of introducing obstacles like walls, closets and so on. For each wall, it is possible to specify its material, and its dimensions (length, height, and thickness), and when the simulated signal passes through a wall, its RSSI is attenuated depending on the type of walls it crossed.

Our initial analysis shows that the parameters having the most critical impact on the RSSI value of each transmitted packet are 1) the wall thickness, 2) the path loss exponent and 3) the radio transmission power. We design a genetic algorithm (GA) that proposes a new set of those parameters at each iteration. A graphical representation of the GA loop is depicted in Figure 1. In each iteration of the genetic loop (generation), the GA proposes a set of populations (set of parameter values) to the simulator, which then generates a new simulated RSSI dataset for each set. A Multi-Layer Perceptron (MLP) with 9 layers is then trained using this simulated dataset. Each trained model is then tested on a small portion of data recorded from the real environment (real test set) to provide a fitness score. The fitness scores associated with each set of parameters guide the GA for the next generation, improving the virtual representation of the environment and reducing the mismatch over time. To reduce the execution time, we distribute the simulator and fitness score calculation on 8 different docker machines, each running on 24 cores in parallel. Ideally, the loop is repeated until convergence is reached, but given the high number of parameters we decided to stop the loop manually (a future improvement on which we are currently working concerns the modification of the GA to reduce the number of parameters to

optimize). At the end of the loop, the GA obtains a set of parameters that represents the configuration of the simulator that better represents the real environment (having the best fitness score).

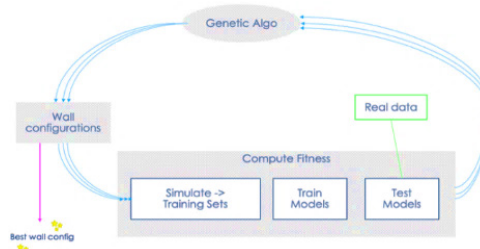


Figure 1: the genetic algorithm loop.

At this point, we train the MLP with a simulated dataset generated using this best configuration, and we use real data to evaluate the accuracy of the localization estimated by the MLP. We aim at realizing a system capable of localizing objects at a room level, which is more than enough for many applications that do not require the exact location. An estimation is considered correct if the estimated room corresponds to the room where the real emitter is placed. Table 1 compares the balanced accuracy on the real test set 1) obtained by the MLP when trained using a synthetic dataset generated using standard parameters ("Standard" column) to 2) those obtained by the same MLP but trained using a dataset obtained with the best configuration ("Best config") as suggested by the GA. We evaluate the performance using 5 different initialization seeds ("Seed"). Although we trained the model using synthetic data for the whole area to monitor, we only evaluate its accuracy for the rooms containing an emitter used in the fitness score calculation. Generalizing this to the whole area is part of future activities.

Table 1: Comparison of balanced accuracy on test set.

Seed	Standard	Best config
0	76.53 %	84.53 %
42	72.43 %	87.05 %
276	72.45 %	95.19 %
1024	72.10 %	83.78 %
12345	79.44 %	80.26 %
Avg (std)	74.59 % (+- 2.9)	86.23% (+-4.9)

We observe that the localization accuracy is increased by more than 10% on average, confirming the positive impact of the GA in the improvement of the representation of the reality in the simulated dataset. These results are extremely encouraging, and once we will reduce the GA convergence time and we generalize the MLP to work in rooms not part of the initial training data set, we will be able to obtain a reliable room-level localization solver that can be quickly deployed in any indoor environment and trained using only a small real dataset as reference.

[1] L. Bergamini, *et al.*, "Mantis: an Indoor Localization and Navigation Framework with Machine Learning Support", CSEM Scientific and Technical Report (2021) 24.