Artificial Neural Network-based Estimation of Individual Localization Errors in Fingerprinting

Filip Lemic and Jeroen Famaey
Internet Technology and Data Science Lab (IDLab), Universiteit Antwerpen - imec, Belgium
Email: {filip.lemic, jeroen.famaey}@uantwerpen.be

***Abstract*—Location information is among the main enablers of context-aware applications and wireless networks. Practical localization services are able to generate location estimates that are generally erroneous. To maximize its usability and benefits, each location estimate should be leveraged jointly with the corresponding estimate of its localization error. Hence, we propose an Artificial Neural Network (ANN)-based method for the estimation of individual localization errors. We do that for fingerprinting, one of the most prominent localization solutions for GPS-constrained environments. First, we provide insights on how to optimally hyperparameterize the proposed method. We do that by exploring its hyperparameter' space in order to find its close-to-optimal hyperparameterization for different environments and fingerprinting technologies. We believe the provided insights can serve to reduce the overhead of deploying the method in new environments. Second, we demonstrate that the method, when hyperparameterized according to the provided insights, substantially outperforms the current state-of-the-art. The improvement is more than 25% in the best case scenario.**

I. Introduction

Location information is a valuable source of contextual information that can be utilized by both end-users' applications (e.g., tracking of people through their smartphones or wearables for assisted living, wellness, social networking, and other applications) [1] and for optimizing the performance of wireless networks (e.g., location-based relaying, routing, or link establishment) [2]. Localization solutions are not able to provide perfectly accurate location estimates, but these estimates are generally burdened with certain levels of localization errors [3]. To maximize the usability, each location estimate should be utilized together with the corresponding estimates of its quality. This quality is primarily characterized by the amplitude of the localization error of the location estimate.

Static performance benchmarks have traditionally been utilized for estimating the amplitudes of localization errors. These benchmarks provide some statistical characterizations of localization errors across the whole environment of interest and, therefore, are not very accurate in estimating the localization error of an individual location estimate. This is because of the spatial variability and dynamic nature (e.g., due to interference and mobility) of localization errors in an environment [4]. To address this issue, we have proposed a set of regression-based methods for estimating individual localization errors have been proposed for fingerprinting localization [5]. Moreover, we have shown in that these methods significantly outperform the reference estimation grounded on the static benchmarks.

However, the regression-based methods are not perfectly accurate and their performance cannot be further improved [5]. This is due to their simplicity (i.e., small number of tunable hyperparameters), which hampers their potential modifications aiming at maximizing the estimation accuracy. The estimation of individual localization errors should be as accurate as possible, especially for the use-cases involving location-based optimization of wireless networks. For example, the authors in [6] and [7] have shown that the over- or underestimation of individual localization errors significantly reduces the benefits of location-based relay selection and device-to-device link establishment, respectively. Artificial Neural Networks (ANNs) are arguably the state-of-the-art machine learning tools for estimating various types of continuous output variables [8]. In contrast to regression, ANNs have a substantially larger number of tunable hyperparameters, which enables their optimization and tuning the ANN for a specific problem.

Along the discussion above, we propose an ANN-based method for estimating individual localization errors in fingerprinting. Moreover, we explore a relatively large hyperparameter' space and tuning ranges of the proposed method. Based on that, we derive indications on how to optimally hyperparameterize the regular ANN architecture for the given problem. We follow by experimentally demonstrating the feasibility of such ANN for estimating individual localization errors. Specifically, we show that the close-to-optimally hyperparameterized ANN significantly outperforms the state-of-the-art regression-based methods in various environments and for multiple fingerprinting technologies. We use fingerprinting due to its popularity for indoor [9], [10] and Global Positioning System (GPS)-confined outdoor environments [11].

II. ANN-based Estimation of Individual Localization Errors in Fingerprinting

Let us assume that a fingerprinting-based localization solution is deployed in a given environment. Moreover, we assume the availability of a static performance benchmark of the solution in the environment. First, the benchmark provides a set of M Received Signal Strength (RSS) observations from N Base Stations (BSs) used for fingerprinting, where each RSS observation is measured at a certain ground truth location in the environment. Hence, the RSS observations can be expressed by $\mathbf{RSS_i} = [RSS_{1,i}, ..., RSS_{N,i}]$, $i = 1, ..., M$. Second, the benchmark provides a mapping between the RSS observation $\mathrm{RSS_i}$ and the corresponding localization error $\mathrm{Error_i}$. The localization error $\mathrm{Error_i}$ for the RSS observation $\mathrm{RSS_i}$ is calculated as the Euclidean between the ground truth location i and the corresponding estimated location.

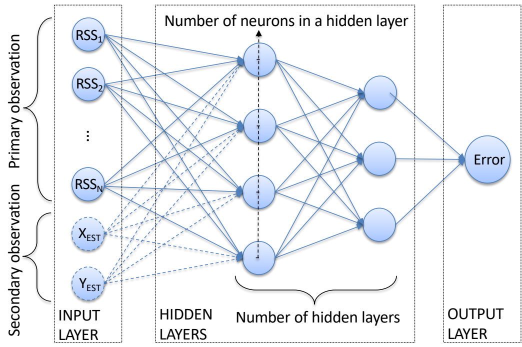


Figure 1: Schematics of the artificial neural network

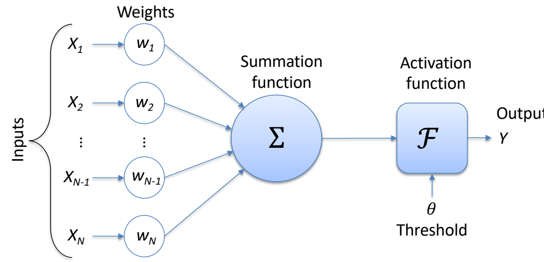


Figure 2: Schematics of a neuron

During the operation of the fingerprinting solution, i.e. when requested to provide an estimate, the ground truth location of that estimate is not known. This means that the localization error for that estimate cannot be calculated and, hence, it has to be estimated. We leverage the standard ANN architecture for this estimation. We use the ANN architecture because of its reported successful utilization for problems with low heterogeneity of input variables (e.g., [8]). The functional diagram of the proposed method is given in Figure 1. In general, the ANN consists of an input layer, a number of hidden layers, and an output layer. The input layer accepts the input variables that are then propagated through the trained ANN for estimating the output variable in the output layer. For the problem at hand, we first consider RSS observations as input variables, hence we call the RSS observations *primary observations*. We do that because there exist use-cases (e.g. [2]) in which there would be no need for estimating location if its localization errors would be higher than a certain set threshold, thus only an RSS observation for that location would initially be available. Second, we consider the estimated location as an additional input variable, thus we name it *secondary observation*. We consider estimated location in a 2-Dimensional (2D) space. Its expansion to a 3-Dimensional (3D) or reduction to 1-Dimensional (1D) space is straightforward.

The ANN generally consists of a number of neurons, with the schematics of a neuron as depicted in Figure 2. The neuron provides a mapping between weighted input variables and an output variable, as shown in the figure. Specifically, the neuron sums the products of inputs X_i and their corresponding weights w_i, and applies an activation function F to the product to get the output. An activation function enables potentially (and usually) nonlinear functional mapping between the input and output variables. The activation function is activated based on a threshold θ, otherwise the output variable equals to zero.

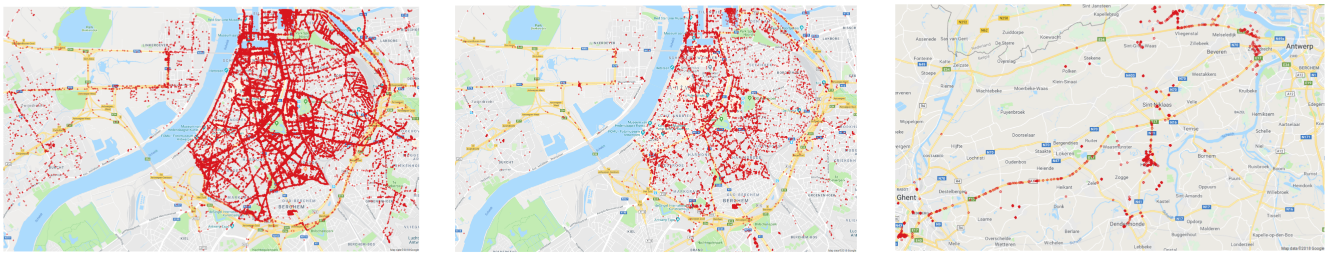
In order for the ANN to become operational, it has to be trained, i.e. the input weights for each neuron have to be learned. We leverage the previously described static performance benchmark for this training. Given that the output variables in the training data (i.e. the localization errors) are known and continuous values, the problem at hand is an instance of a supervised learning-based regression problem. In such a problem, it is possible to compare the output values calculated by the ANN with the correct output values and, based on them, calculate the continuous output error. This procedure is known as forward-propagation. The output error can then be used for adjusting the weights of the neurons, so that in the following iterations the network' calculated output values will be closer to the correct output values. The procedure for adjusting the weights is called back-propagation.

Intuitively, the accuracy of the ANN depends on the amount and quality of data in the training dataset. In addition, the accuracy of the ANN-based estimation can be maximized by carefully selecting hyperparameters of the ANN. The hyperparameters are variables that specify the structure and learning behavior of the ANN. Specifically, the hyperparameters include the number of hidden layers, as well as the number of neurons in each hidden layer, as shown in Figure 1. Moreover, the hyperparameters include the activation function F with its threshold θ, as well as the initialization of weights (i.e. initialization modes or initializers). The hyperparameters also include the constraints on weights' magnitudes and the dropout rate (percentage of randomly selected neurons that are ignored during training), both used to prevent overfitting of the ANN. The weights are modified using a function called optimization function or optimizer. Moreover, the batch size is the total number of training samples present in a single batch, where a batch is one chunk of a training dataset. Finally, one epoch corresponds to an entire dataset being passed forward and backward through the ANN once.

III. Evaluation

A. *Evaluation Methodology*

In the evaluation, we leverage three datasets from two types of outdoor environments (i.e., urban and rural). Their detailed descriptions are provided in [12], while here we briefly summarize their relevant features. First, we use LoRa and SigFox datasets collected in the city of Antwerp, Belgium. Second, we leverage a SigFox dataset collected in a rural environment between the cities of Ghent and Antwerp, Belgium. For each measurement location in the datasets, the Mobile Terminal (MT) broadcasts a packet, followed by its reception by the neighboring BSs. If the packet is received, the BSs log its RSS, otherwise the default RSS noise value of -200 dBm is logged. Hence, each data point contains a vector of RSS values observed by different BSs at a given location of the MT. Moreover, each data point contains the GPS coordinates of the MT, which allows correlating location information with an RSS observation. In the collections of measurements, proprietary nation-wide SigFox and LoRa networks were leveraged.



(a) LoRa dataset (b) SigFox dataset (c) SigFox-rural dataset

Figure 3: Overview of the used datasets

TABLE I: Initial hyperparameters used in ANN training

Hyperparameter	Value
Initialization mode	*uniform*
Activation function and optimized	*relu* / *rmsprop*
Number of hidden layers	2
Number of neurons in hidden layers	30 / 15
Dropout rate	30%
Number of batches/epochs	10/20
Weight constraint	*max(3)*

The LoRa dataset contains 123,529 data-points, as depicted in Figure 3(a). In total, 68 BSs are detected in the dataset. In the SigFox dataset, 14,378 RSS observations were collected (Figure 3(b)). Altogether, 84 BSs have received some of the packets transmitted by the MT. The SigFox dataset collected in the rural environment contains 25,638 data-points observed from altogether 137 BSs, as depicted in Figure 3(c).

The RSS vector from different BSs used for fingerprinting is selected as a fingerprint of a location, which is a well-known fingerprint creation procedure [13]. For calculating the similarity between a training and runtime fingerprint we use the Euclidean distance between RSS vectors, which is again a well-established and extensively used procedure [13]. In the post-processing procedure of fingerprinting, we use k-Nearest Neighbors (kNN) with parameter k set to 4, which has been shown to be best selection for a variety of environments [14].

We randomly divide the datasets into a fingerprinting training, ANN training, and evaluation sets in the ratio 10:70:20. The fingerprinting training set is used for "deploying" the fingerprinting solution. The ANN training set is mimicking a static performance benchmark used for the training of the ANN (and existing regression-based methods). The evaluation set is leveraged for establishing the accuracy of the proposed and existing methods for the estimation of individual localization errors. As the evaluation metric we use "prediction error", which is, for each evaluation point, defined as the absolute value of the difference between the calculated and estimated localization error. The distributions of prediction errors for the entire evaluation set are depicted as regular box-plots.

B. Optimal ANN Hyperparameterization

In the first step of the evaluation, our goal is to determine which hyperparameters have a considerable effect on the performance of the ANN. We do that so that in the future deployments of the method, the hyperparameters' optimization space could be reduced, i.e. the hyperparameter tuning would be needed only for influential hyperparameters. We define influential hyperparameters as the ones whose change affects the average prediction error by more than 10%. For the influential hyperparameters, our consequent goal is to determine if there are hyperparameterizations that consistently yield optimal performance across different environments and fingerprinting technologies. Given that such hyperparameterizations exist, the hyperparameter tuning in the future deployments could be constrained to the values indicated by our evaluation.

The initial hyperparameters used in the ANN training are summarized in Table I. These are selected by following a set of rule-of-thumb guidelines for parameterizing ANNs based on the input data features [8]. Note that the number of neurons in a given hidden layer equals half of the number of neurons in the previous hidden layer, rounded to the lower whole number. This is again based on the rule-of-thumb guidelines from the literature [8]. We tune hyperparameters individually, while keeping the other ones at their initial values (Table I). Hence, we are able to derive a set of close-to-optimal (i.e. locally optimal) ANN hyperparameters for different deployment environments and fingerprinting technologies. We follow this procedure, in contrast to performing an extensive multi-hyperparameter search (which would result in a globally optimal hyperparameters) because of an extremely high computational complexity and time overhead of that alternative. The results for different dropout rates and weight constraints, initialization modes, numbers of batches and epochs, activation functions, optimizer functions, and numbers of neurons and layers are shown in Figures 4 5, 6, 7, 8, and 9, respectively. The sub-figures depict the results for the LoRa, SigFox, and SigFox-rural datasets, respectively.

The first observation from the results is that indeed the optimal selection of hyperparameters can substantially improve the performance of the ANN. For example, only by changing the activation function from *softmax* to *tanh*, the prediction error can be reduced by roughly 30% for all three scenarios, as shown in Figure 7. Similarly, by changing the optimizer function from *nadam* to *adamax*, the prediction error can be reduced by up to 35%, as depicted in Figure 8. However, not all hyperparameters play a significant role in the ANN optimization for the given problem. For example, changes in the initialization modes have only minor effects on the overall performance of the ANN (i.e., less than 10% in average prediction error), as shown in Figure 5. Similarly, a change in the number of batches and epochs does not have a significant effect on the ANN performance (Figure 6).

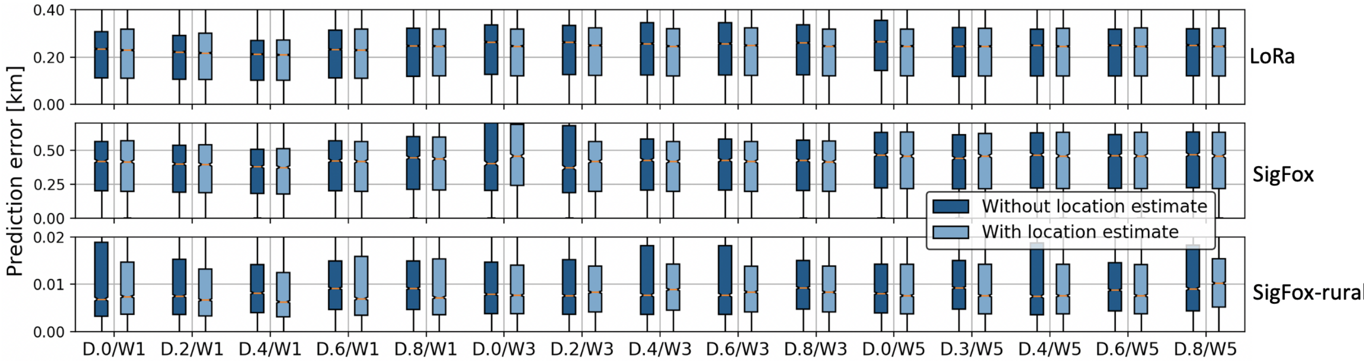


Figure 4: Summarized results for different dropout rates and weight constraints (e.g. D.3/W5: dropout rate = 30%, weight constraint = 5)

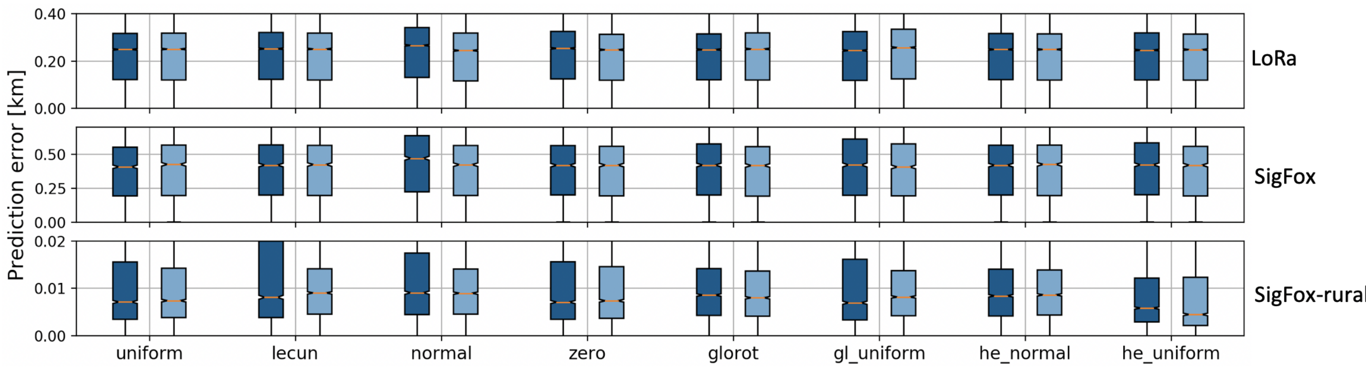


Figure 5: Summarized results for different initialization modes

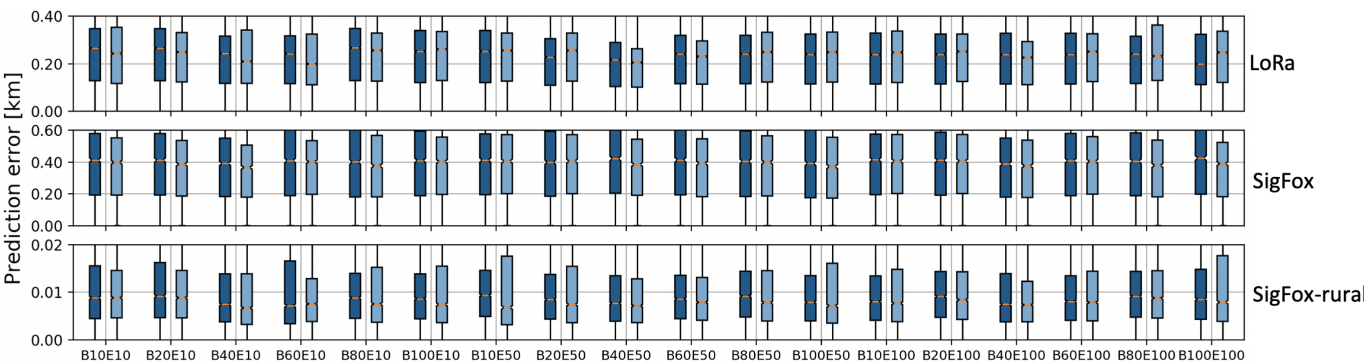


Figure 6: Summarized results for different batches and epochs

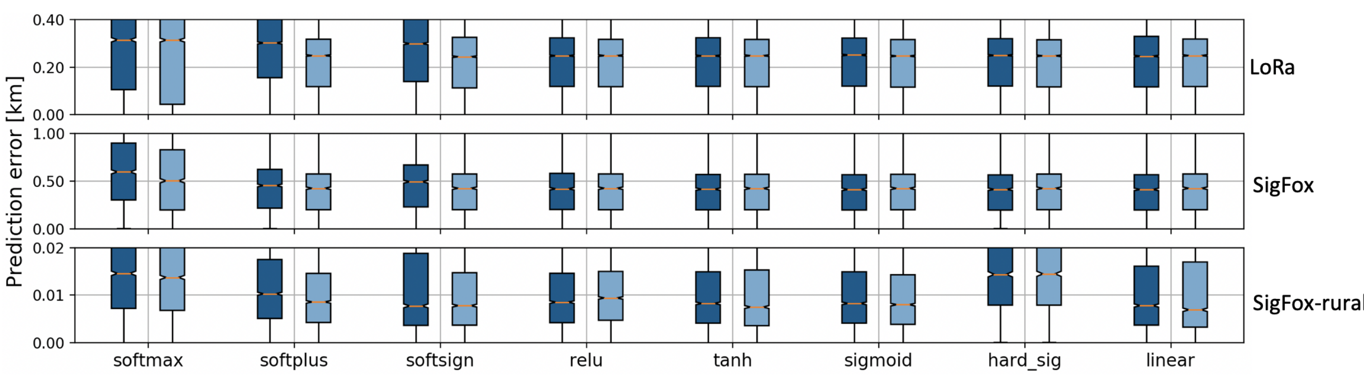


Figure 7: Summarized results for different activation functions

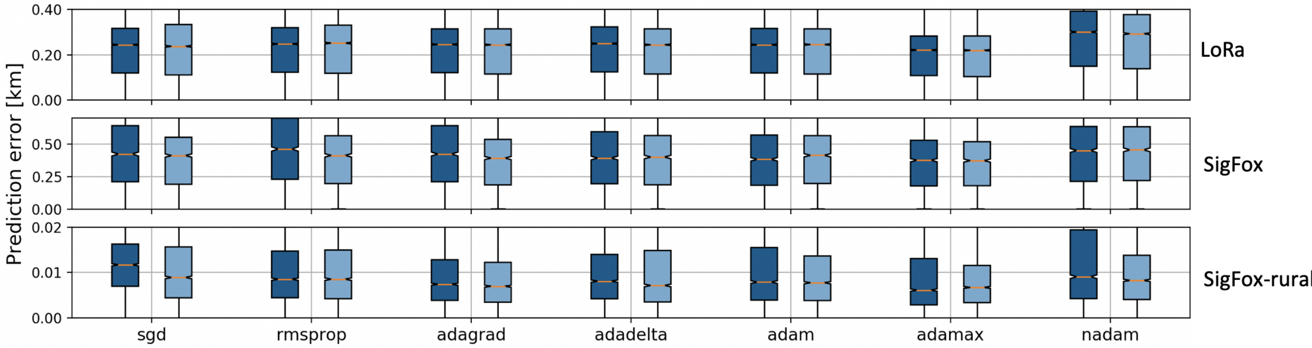


Figure 8: Summarized results for different optimizer functions

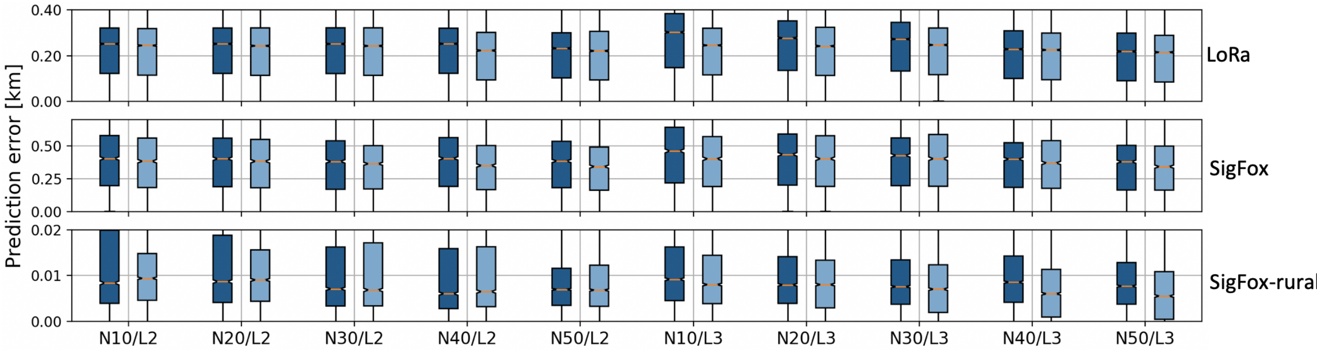


Figure 9: Summarized results for different numbers of neurons and layers (e.g., N30/L3: 30 neurons in the first hidden layer, 3 hidden layers)

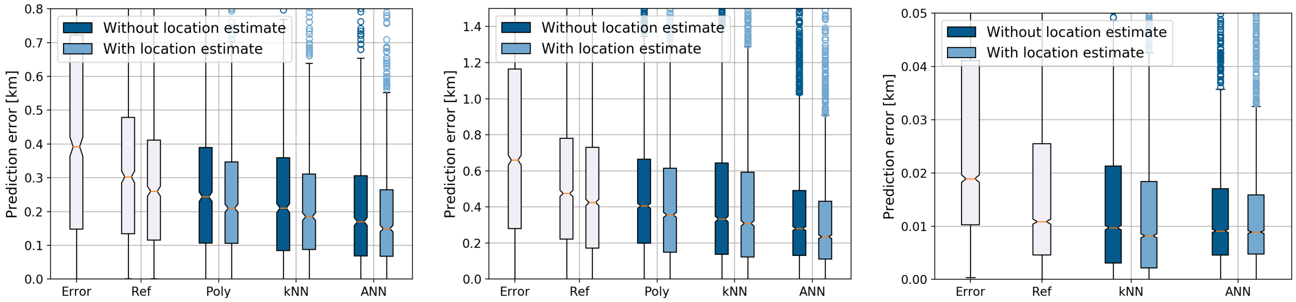
Second, our results indicate that close-to-optimal hyperparameters can be extrapolated from one environment to another, as well as from one technology to another. For example, the dropout rate of 0.4 and weight constraint of *max(1)* yield the optimal performance for LoRa and SigFox-based fingerprinting in the same urban outdoor environment, as well as for SigFox-based fingerprinting in a rural outdoor environment (Figure 4). Similar observations can be made for other hyperparameters, as visible in the figures.

The first observation motivates the need for careful selection of the values for the significant hyperparameters, which is, however, a burdensome and lengthy process. We believe that the derived indications can serve for substantially reducing the hyperparameters' optimization space for future deployments. This claim is further supported by the second observation that the extrapolation of hyperparameters across environments and technologies seems to be feasible. Specifically, the optimal hyperparameters include setting the activation function to *tanh*, optimization function to *adamax*, using the dropout rate of around 40% and weight constraint of max(1), and setting the number of neurons to 50, 25, and 12 in the first, second, and third hidden layer, respectively (in general terms, an increase in the number of neurons in the hidden layers benefits the estimation accuracy).

C. Comparison with Existing Methods

In the second step of the evaluation, we demonstrate the feasibility of the proposed method for the estimation of individual localization errors in fingerprinting. We do that by demonstrating that the performance of the close-to-optimally hyperparameterized ANN is substantially and consistently better than the existing state-of-the-art methods based on simple regression. The results are depicted in Figure 10. The first box-plot in the figures (labeled with "Error") depicts the distributions of localization errors. The second box-plot in Figures 10(a) and 10(b) (labeled with "Ref") shows the distributions of prediction errors in case the localization error is estimated using the reference estimation. In the reference estimation, as the estimated localization error the fingerprinting solution reports the average localization error across the entire environment, where the average localization error is derived from the static performance benchmark. This box-plot is omitted in Figure 10(c) for reasons discussed below. The third box-plot (the second one in Figure 10(c)) also depicts the distributions of prediction errors for the reference estimation. However, here the estimated localization error is derived by first estimating location, followed by mapping of that estimate to the nearest ground truth location from the benchmark. The localization error attributed to that location from the benchmark is then reported as the estimated localization error.

The following box-plots depict the distribution of prediction errors for the polynomial and kNN regression, and the ANN-based method. The polynomial and kNN regression are shown to be the best performing regression-based methods for the problem at hand [5]. These are depicted for both the case when only RSS observations have been used as an input variable and the case when estimated location have been used as an additional input variable for estimation. As visible in the figures, ANN-based estimation generally outperforms other methods, in terms of both average/median prediction errors and the outliers. The improvement is higher than 50% in the best case scenario, compared to the reference estimation based on static



(a) LoRa dataset (b) SigFox dataset (c) SigFox-rural dataset

Figure 10: Comparison between different methods for estimating localization errors in fingerprinting

benchmarks. Moreover, the ANN-based estimation yields at least 15% better performance in terms of average prediction error than the second best method (i.e. kNN regression) and this improvement is higher than 25% in the best case scenario.

Particularly interesting are the results observed for the SigFox-rural dataset. In the dataset, there are regions with a small number of measurement locations (even a single measurement point), as well as clusters with dense measurement locations (Figure 3(c)). This results in highly diverse localization errors, i.e. the localization errors in the clusters are much smaller than the ones at sparsely populated locations. This effect can also be discerned by the large difference between median and average localization errors, which equal roughly 18 and 370 meters, respectively. A similar observation has been made in [12]. The reference estimation of localization errors based on average localization error due to this heterogeneity yields very large prediction errors (i.e. on average more than 500 m) and is therefore omitted in Figure 10(c). For the same reason, the reference that grounds the estimation of localization error on mapping of the estimated location to the nearest location from the static performance benchmark yields very accurate results. Nevertheless, the ANN method yields 15% more accurate estimation than this reference in terms of average prediction errors, even in this more challenging scenario. Note that the polynomial regression also yields very high prediction errors, hence its results are also omitted.

IV. Conclusion

We proposed a method for ANN-based estimation of individual localization errors in fingerprinting. By evaluating the method using three large outdoor datasets, we have demonstrated its feasibility and encouraging performance for the given problem. Specifically, we have shown that it outperforms the reference based on average localization errors and the currently available regression-based methods by up to 55% and 30%, respectively. Moreover, we have reduced the ANN hyperparameters' space and tuning ranges, which will simplify the deployment of the method in new environments. We believe that the obvious drawback of the method, i.e. the fact that a significant amount of data has to be collected for its training, can be overcome by crowd-sourcing. Crowd-sourcing is an established procedure for collecting training measurements for fingerprinting (e.g. [15]), and there are no intrinsic differences between the two training procedures. Future efforts will aim at further optimizing the hyperparameters' space, as well as evaluating the method in new environments and for other technologies, primarily WiFi-based indoor fingerprinting.

Acknowledgments

This research received funding from the ICON project MuSCLe-IoT. MuSCLe-IoT is realized in collaboration with imec, with project support from VLAIO (Flanders Innovation and Entrepreneurship). Project partners are imec, Flash Private Mobile Networks, Engie M2M, Sensolus, and Aertssen. The authors would also like to thank Dr.-Ing. Vlado Handziski for the fruitful discussions that eventually led to this paper.

References

[1] L. Pei *et al.*, "Indoor/outdoor seamless positioning technologies integrated on smart phone," in *Advances in Satellite and Space Communications, 2009*, IEEE, 2009, pp. 141–145.

[2] R. Di Taranto *et al.*, "Location-aware communications for 5g networks," *IEEE Signal Processing*, vol. 31, no. 6, pp. 102–112, 2014.

[3] D. Lymberopoulos *et al.*, "A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned," in *Information processing in sensor networks*, ACM, 2015, pp. 178–189.

[4] F. Lemic, V. Handziski, *et al.*, "Experimental evaluation of rf-based indoor localization algorithms under rf interference," in *Localization and GNSS (ICL-GNSS)*, IEEE, 2015, pp. 1–8.

[5] F. Lemic *et al.*, "Regression-based estimation of individual errors in fingerprinting localization," *IEEE Access*, pp. 1–1, 2019.

[6] J. J. Nielsen *et al.*, "Location-quality-aware policy optimisation for relay selection in mobile networks," *Wireless Networks*, vol. 22, no. 2, pp. 599–618, 2016.

[7] F. Lemic *et al.*, "Location-based decision-making mechanism for device-to-device link establishment," in *Vehicular Technology Conference (VTC-Fall)*, IEEE, 2017, pp. 1–6.

[8] V. N. Vapnik, "An overview of statistical learning theory," *IEEE transactions on neural networks*, vol. 10, no. 5, pp. 988–999, 1999.

[9] V. Honkavirta *et al.*, "A Comparative Survey of WLAN Location Fingerprinting Methods," in *WPNC 2009*, IEEE, 2009, pp. 243–251.

[10] A. Behboodi *et al.*, "Hypothesis testing based model for fingerprinting localization algorithms," in *VTC Spring*, IEEE, 2017.

[11] T. Janssen *et al.*, "Outdoor Fingerprinting Localization Using Sigfox," in *Indoor Positioning and Indoor Navigation (IPIN)*, 2018, pp. 1–6.

[12] M. Aernouts *et al.*, "Sigfox and lorawan datasets for fingerprint localization in large urban and rural areas," *Data*, vol. 3, no. 2, 2018.

[13] F. Lemic, A. Behboodi, *et al.*, "Experimental Decomposition of the Performance of Fingerprinting-based Localization Algorithms," in *Indoor Positioning and Indoor Navigation (IPIN'14)*, 2014.

[14] F. Lemic *et al.*, "Demo: Testbed infrastructure for benchmarking rf-based indoor localization solutions under controlled interference," in *European Wireless Sensor Networks (EWSN'14)*, 2014, pp. 1–5.

[15] Q. Jiang *et al.*, "A probabilistic radio map construction scheme for crowdsourcing-based fingerprinting localization," *IEEE Sensors Journal*, vol. 16, no. 10, pp. 3764–3774, 2016.