Take down policy:
If you believe that this document breaches copyright, please contact the authors, and we will investigate your claim.

Towards Accelerated Localization Performance Across Indoor Positioning Datasets

Lucie Klus*,†, Darwin Quezada-Gaibor†,*, Joaquín Torres-Sospedra‡,
Elena Simona Lohan*, Carlos Granell† and Jari Nurmi*
*Electrical Engineering Unit, Tampere University, Tampere, Finland
†Institute of New Imaging Technologies, Universitat Jaume I, Castellón, Spain
‡Algoritmi Research Centre, Universidade do Minho, Guimarães, Portugal

Abstract—The localization speed and accuracy in the indoor scenario can greatly impact the Quality of Experience of the user. While many individual machine learning models can achieve comparable positioning performance, their prediction mechanisms offer different complexity to the system. In this work, we propose a fingerprinting positioning method for multi-building and multi-floor deployments, composed of a cascade of three models for building classification, floor classification, and 2D localization regression. We conduct an exhaustive search for the optimally performing one in each step of the cascade while validating on 14 different openly available datasets. As a result, we bring forward the best-performing combination of models in terms of overall positioning accuracy and processing speed and evaluate on independent sets of samples. We reduce the mean prediction time by 71% while achieving comparable positioning performance across all considered datasets. Moreover, in case of voluminous training dataset, the prediction time is reduced down to 1% of the benchmark's.

Index Terms—Cascade, Fingerprinting, Indoor positioning, Localization, Machine learning, Prediction speed

I. INTRODUCTION

The exponential growth of wearable and Internet of Things (IoT) devices requiring positioning services demands a quick response from the Indoor Positioning System (IPS) to provide Quality of Experience (QoE) to the end-users. The requirement is more impactful in real-time solutions where a fast response is a must [1]. That is why multiple algorithms and techniques were developed over the years in order to fulfil the requirements of these cutting-edge devices. For instance, data compression techniques have been applied onto indoor positioning data to reduce the processing and response times as well as to improve the positioning accuracy [2].

Due to the complex signal propagation characteristics in indoor scenarios, fingerprinting methods are commonly utilized. In order to operate, they require a pre-measured dataset of fingerprints, referred to as a radio map, to adapt the matching mechanism to the environment. In large scenarios such as universities, the radio maps contain hundreds of thousands of fingerprints [3]–[5], which may be time-consuming to

process during the online phase. To reduce the processing time during the online phase, the authors in [6] provided three new methods to enhance the coarse and fine-grained search in k-means clustering to be used in the offline and online phase of IEEE 802.11 Wireless LAN (Wi-Fi) fingerprinting. As a result, the authors reduced the computational cost of estimating the user position, while slightly improving the position estimation.

One of the student teams of the "Data Analytics and Machine Learning (ML)" program's challenge [7] considered the possibility of cascading the models and achieved the best results among other teams, outperforming several benchmarks. Their performance proves the cascade's efficiency in terms of positioning accuracy, yet does not consider computational effort, nor its generalizaiton capabilities.

The authors of [8] propose a clustering method based on the strongest Access Point (AP) match before performing localization by k-Nearest Neighbors (k-NN). Their solution outperforms benchmark solutions in terms of both Prediction Time (PT) and positioning accuracy. Similarly, [9] used classification algorithms to reduce the complexity in the online phase of Wi-Fi fingerprinting. They included environmental sensors (humidity and temperature sensors) to provide extra information to the fingerprinting techniques, allowing to reduce the location complexity. In the framework of [9], two classification techniques were used to first determine the floor where the target node is located, and the second to filter the location of the fingerprint in the radio map. As a result, the authors improved positioning accuracy.

In this work, we develop a novel approach for a multi-building and multi-floor indoor positioning, which takes advantage of the heterogeneity of the labels within the datasets. The proposed cascade of optimally chosen models for building classification, floor classification and 2D localization regression is able to positively impact the floor-hit, as well as highly improve the prediction time of the considered datasets.

The main contributions of this paper are as follows:
- We propose an indoor positioning strategy based on the cascade of three ML models, designed to iteratively reduce the search-space of the consecutive models and consequently strongly improving the PT without harming the positioning capabilities.
- We introduce and evaluate 9 distinct classification models and 7 different regression model architectures, both un-

derutilized and ever-present ones across the literature, to find a good general well-performing sequence of models on 14 open-source indoor positioning datasets.

- We perform the testing of the proposed cascade model on an independent subset of data, showing a significant increase in efficiency, in terms of PT.

The rest of the paper is structured as follows. Section II introduces the utilized methods and the proposed system model, Section III presents the datasets, and the numerical results of both the evaluation and testing phases, while Section IV concludes this work.

II. MATERIALS AND METHODS

In this section we present the technical parameters of this work, including the utilized ML models, and the principles of the fingerprinting method. Also, we describe the proposed system model and its work-flow.

A. Utilized Methods

In recent decades, the availability of numerous ML models became almost abundant, and their utilization omnipresent across scientific branches, as nowadays, almost each scientific paper mentions the advancements in Deep Learning (DL) and Artificial Intelligence (AI), considering only the most recent models. Nevertheless, the family of ML models offers countless different solutions as well, which are hardly ever utilized.

In this work, we utilize 9 ML methods, involving commonly used classifiers and regressors, as well as the under-utilized approaches, such as Linear Support Vector Machine (LSVM) or Decision Tree (DT).

We consider the following classifiers in the evaluation:

k-Nearest Neighbors (k-NN) with $k = 1$, 3 and 11 (further referred to as e.g. 3NN for k-NN with $k = 3$) as the representative of the most commonly used matching algorithm in non-parametric IPS. We utilize Manhattan distance as the cost function. The representatives of number of k were chosen based on the comprehensive sweep of values and parameters performed in [10], also utilizing some of the datasets used in this work. Moreover, the lower values of k perform well in sparsely covered areas, while higher k (e.g. 11) improves generalization capabilities of the model.

Weighted k-Nearest Neighbors (Wk-NN) with $k = 3$ and 11 (further referred to as e.g. W3NN for Weighted k-NN with $k = 3$) is the alteration of k-NN, which after finding the k neighbors considers their importance based on the inverse of their distance to the reference sample. The values of k were kept the same as in k-NN, despite Wk-NN's inherent capability of filtering less relevant samples.

Linear Support Vector Machine (LSVM) is a ML approach capable of efficiently finding boundaries between classes in high-dimensional space. We utilize the model with the linear kernel, as it achieved significantly better performance in terms of both accuracy and the training time than the commonly used Radial Basis Function. The model parameters are 'l2' penalty and the regularization parameter $C = 1$.

Decision Tree (DT) creates a tree-like structure of decision boundaries using the most relevant features to perform classification. We apply Gini impurity, 50 maximum depth of the tree and greedy (best) splitting strategy, since the randomization of trees is performed by the next model.

Random Forest (RF), as the name suggests, builds multiple randomized decision trees and performs classification as the ensemble of the individual results. We built the classifier using 10 trees with 50 maximum depth, Gini impurity, and the number of features considered in each split limited to a square root of total features. Further increasing the parameters did not provide improved results.

Neural Network (NN); The NN considered in this work involves of a single hidden densely connected layer with 100 neurons, followed by a classification layer with the number of neurons equal to the number of classes (e.g. 3 neurons for 3 possible floors). The training is realized with Adam optimizer for up to 100 epochs with 'l2' regularization of the hidden layer. We denote, that further tuning of the hyperparameters, which is beyond this paper's scope, might provide improved results, as previously discussed in the literature [11], [12].

AdaBoost (AdB) is an ensemble method, which subsequently grows multiple shallow decision trees, where each conclusive tree focuses on the incorrectly classified samples [13]. We set the number of estimators to 50, and depth of each tree to 1. The idea behind this model is considering the majority vote of many weak classifiers.

Naive Bayes (NB) is a simple classification scheme, which assumes pair-wise independence of features and their likelihood as Gaussian [14].

Quadratic Discriminant Analysis (QDA) fits a Gaussian density to each class, instead of finding only a linear boundary, as in e.g. LSVM. The classifier provides improved results in case of intersecting classes.

For the regression, we consider most of the ML model architectures as used for the classification schemes. We adjust the models as regressors, allowing to predict continuous values (classification models only predict discrete classes). The regressors include: **k-Nearest Neighbors (k-NN)** with $k = 1$ & $k = 3$ & $k = 11$, **Weighted k-Nearest Neighbors (Wk-NN)** with $k = 3$ & $k = 11$, **Linear Support Vector Machine (LSVM)**, **Decision Tree (DT)**, **Random Forest (RF)**, **Neural Network (NN)**, and **AdaBoost (AdB)**, with the same hyperparameter settings as their classifier alternatives.

B. Positioning via Fingerprinting

Nowadays, fingerprinting is one of the most commonly utilized approaches for localization in IPS. The simple fact, that the method does not require any specific knowledge about the deployment is one its greatest strengths. Fingerprinting positioning requires only the pre-recorded database of fingerprints from the deployment, and a matching algorithm in order to operate. The approach is most commonly used in deployments with strong Non-Line-of-Sight (NLoS) and signal scattering characteristics, such as indoors [15] or dense urban areas [16]–[18].

The traditional, and most commonly used matching algorithm is k-NN and its variants, whose tuneable parameters, such as number of k or distance metric, offer great fine-tuning options, resulting in unmatched positioning performance [10]. Recently, many works, such as [11], [12] substitute k-NN with different types of Neural Networks (NNs), which ensure faster matching capabilities, especially on large datasets, but usually do not achieve the same positioning accuracy.

In this work, we aim to evaluate multiple ML models, not restricted to k-NN and NN only, to build a cascade of models capable of significantly improving the PT, as well as keeping or improving the positioning accuracy, compared to using only a single model.

C. Proposed System Model

In this work, we develop a cascade positioning algorithm, which exploits the abundance of labels within the indoor positioning datasets. The considered cascade of models first finds the building estimate, then estimates the floor, and the last model finds the exact user coordinates, all by considering only the relevant samples. While certain works employ the strategy of creating a model for predicting building, floor, and 2-Dimensional (2D) position as the separate tasks [12], [19], we fully utilize the knowledge obtained in each step.

The visualization of the proposed cascade algorithm is shown in Fig. 1. Each dataset first trains a Building Hit (BH) classifier using all available samples within the deployment, and the building label as the target. For each building in the deployment, a separate Floor Hit (FH) classifier is created and trained only using the samples from the adequate building, while predicting the floor label. Finally, for each floor within each building, a separate 2D Localization (2D-L) regression model is trained using only the training samples from the relevant floor in the relevant building to accurately localize the user. After the models are trained in off-line fashion, the incoming fingerprint is evaluated on-line as follows.

In the first stage, the BH model estimates the fingerprint's building label, based on which the corresponding trained FH classifier is selected. In the second stage, the selected FH model predicts the fingerprint's floor label and enables the selection of the 2D-L model for the adequate floor. The third stage of the algorithm predicts the fingerprint's coordinates using the selected model and the ensemble of the building label, floor label and the user coordinates is complete.

The proposed cascade of models offers a stable separation of samples based on building and floor labels into pre-defined number of classes. While AP-wise clustering methods [8], [9], [19] separate the space into clusters based on samples' features, the proposed cascade splits the data based on their labels and thus ensures more balanced distribution of samples per classifier. As a result, each considered individual prediction model (e.g. per floor) has a sufficient number of samples. The balanced separation is crucial when considering trainable models (e.g. NN or RF), while reducing the number of training samples accelerates k-NN performance.
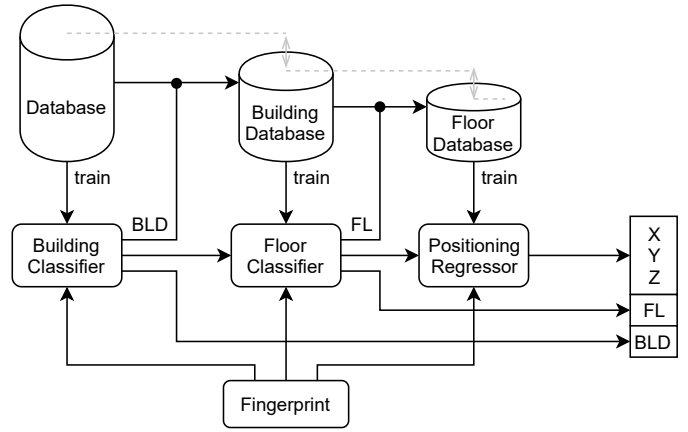

Fig. 1: General system model work-flow. Each subsequent model considers only the training samples relevant to the deployed area.

In order to demonstrate the best combination of the classifiers and the regressor in the cascaded architecture, we later compare the performance of 9 commonly known models. We propose the combination of methods to gain comparable accuracy of positioning in comparison to using the traditional model only, while vastly improving the time required for localization (prediction). Additionally, we propose a cascade combination of methods, which further decreases the PT at the expense of lowering the positioning accuracy.

III. EXPERIMENT AND RESULTS

In this section, we first introduce the 14 available datasets. These datasets were split into training and testing subsets and used for tuning, and later evaluating of the proposed method, as described in this section.

For the implementation of this work, we used Python 3.8 with Numpy, Scipy, and Scikit-Learn libraries. For the result processing and visualization, we used MATLAB R2020b. The hardware included Intel(R) Core(TM) i7-8750H processor with 32 GB RAM.

For the purposes of reproducibility and replicability of the experiments we share the codes on Zenodo [20].

A. Available Datasets

In order to evaluate the proposed method and its applicability over different scenarios, we used 14 publicly available fingerprinting datasets. All of them consist of multiple-floor environments, ranging from 2 to 16 floors and in total involving 82 individual floors, while UJI 1&2 are also an example of multi-building scenarios (each having 3 buildings). All datasets were obtained using Wi-Fi as a primary technology for Received Signal Strength (RSS) measurements. More details regarding their properties may be found in Table I.

The datasets were measured and provided by the three universities. Universitat Jaume I, Spain provided LIB 1&2 [3] and UJI 1&2 [4], Tampere University, Finland provided SAH 1 and TIE 1 [5], TUT 1&2 [21], [22], TUT 3&4 [23], TUT 5

TABLE I: Overview of Datasets

Dataset	Samples	Training samples	Testing samples	APs	Buildings	Floors
LIB 1	3696	576	3120	174	1	2
LIB 2	3696	576	3120	197	1	2
SAH 1	9447	9291	156	775	1	3
TIE 1	10683	10633	50	613	1	6
TUT 1	1966	1476	490	309	1	4
TUT 2	760	584	176	354	1	3
TUT 3	4648	697	3951	992	1	5
TUT 4	4648	3951	697	992	1	5
TUT 5	1428	446	982	489	1	3
TUT 6	10385	3116	7269	652	1	4
TUT 7	9291	2787	6504	801	1	3
UJI 1	20972	19861	1111	520	3	13
UJI 2	26151	20972	5179	520	3	13
UTS 1	9496	9108	388	589	1	16

TABLE II: Building Hit Classification Results for Validation

	1NN	3NN	W3NN	11NN	W11NN	LSVM	DT	RF	NN	AdB	NB	QDA
	Validation Misclassification [samples]											
UJI 1	**0**	**0**	**0**	1	1	**0**	2	1	**0**	190	3	73
UJI 2	**0**	2	2	3	3	**0**	5	5	**0**	45	7	24
Avg	**0**	1	1	2	2	**0**	3	3	**0**	117	5	48
	Validation Prediction Time [s]											
UJI 1	27.31	27.39	25.81	26.00	26.23	0.103	**0.007**	0.012	0.014	0.238	0.051	0.104
UJI 2	27.78	28.60	30.87	30.04	30.72	0.110	**0.008**	0.013	0.017	0.270	0.059	0.110
Avg	27.54	27.99	28.34	28.02	28.48	0.107	**0.008**	0.013	**0.016**	0.254	0.055	0.107

[24] and TUT 6&7 [25]. University of Sydney, Australia provided UTS 1 [12].

These datasets were already used in a number of other previously published works, e.g., in [2], [11], [26] and [27].

B. Choosing the Methods - Validation

In this subsection, we explain the method for choosing the best-performing combination of a cascade of two classifiers and a regressor for BH, FH, and 2D-L estimates, respectively. As a result, we aim to obtain the optimally performing model for an arbitrary dataset.

For this purpose, we split the original training database of each dataset into training and validation parts in 80:20 ratio. The part dedicated for testing is omitted in the whole validation phase, as it is used for unbiased evaluation of the final model only.

Building Hit (BH)

In order to find the most suitable method for building classification, we performed a sweep over all classifiers as described in Section II, in terms of validation accuracy, training time and time required for prediction. The results of the sweep can be found in Table II, where the validation accuracy is represented by the number of misclassified samples. We provide the total number of misclassifications, rather than a fraction, since only a limited number of samples are in the areas where more than 1 building can be considered. The PT reflects the effort required for the task.

We note that the performance of the BH can only be validated on 2 datasets, as, among considered, only the datasets UJI 1 and UJI 2 are multi-building environments.

The results in Table II show the perfect classification accuracy of 1NN, LSVM and the NN models. As the building

TABLE III: Floor Hit Classification Results for Validation

	1NN	3NN	W3NN	11NN	W11NN	LSVM	DT	RF	NN	AdB	NB	QDA
	Validation Accuracy [%]											
LIB 1	**100**	**100**	**100**	**100**	**100**	**100**	98.3	**100**	**100**	**100**	96.6	91.4
LIB 2	**100**	**100**	**100**	**100**	**100**	**100**	99.1	**100**	**100**	**100**	97.4	97.4
SAH 1	**100**	**100**	**100**	99.8	**100**	**100**	99.7	99.6	**100**	88.6	96.0	96.3
TIE 1	**100**	99.9	99.9	99.6	99.7	**100**	99.5	99.4	99.9	67.7	87.4	87.4
TUT 1	98.0	98.3	98.3	97.0	97.3	99.0	93.6	96.6	**99.3**	66.2	56.1	54.1
TUT 2	**100**	99.1	99.1	94.0	97.4	98.3	96.6	97.4	99.1	89.7	74.4	71.8
TUT 3	89.3	82.1	82.9	72.1	75.7	93.6	73.6	91.4	**95.0**	55.7	60.7	42.9
TUT 4	95.7	96.1	95.8	94.2	94.4	96.3	87.5	94.3	**96.6**	66.9	61.3	69.9
TUT 5	**100**	**100**	**100**	90.0	95.6	**100**	94.4	92.2	**100**	62.2	91.1	75.6
TUT 6	99.8	**100**	**100**	99.8	99.8	**100**	97.3	99.2	**100**	71.2	84.6	84.1
TUT 7	**98.9**	97.8	98.6	97.7	97.8	98.4	96.1	96.8	98.6	91.2	92.7	85.5
UJI 1$_1$	**99.8**	99.7	99.7	99.4	99.5	99.5	97.2	99.1	**99.8**	63.4	49.8	68.1
UJI 1$_2$	**99.9**	99.8	99.8	99.2	99.7	99.6	98.4	99.6	99.7	68.1	63.7	64.6
UJI 1$_3$	99.8	99.6	**99.8**	99.6	99.7	99.4	97.9	99.3	99.6	56.4	47.0	42.0
UJI 2$_1$	98.7	98.8	99.0	98.7	98.8	**99.1**	96.0	98.3	**99.1**	58.7	42.8	48.6
UJI 2$_2$	99.1	98.7	98.8	98.2	98.6	**99.2**	96.6	98.4	**99.2**	68.7	65.7	64.1
UJI 2$_3$	99.6	99.3	99.4	99.5	99.6	99.5	97.4	98.8	**99.7**	60.8	51.5	48.4
UTS 1	**99.9**	99.8	**99.9**	99.3	**99.9**	**99.9**	99.5	99.8	**99.9**	23.1	92.3	98.8
Avg	98.8	98.3	98.4	96.6	97.4	99.0	95.5	97.8	**99.2**	69.9	72.8	71.7
	Validation Prediction Time [s]											
Avg	2.687	2.675	2.657	2.724	2.622	0.052	**0.002**	0.005	**0.003**	0.084	0.026	0.056

misclassification leads to severe positioning errors, therefore the building-hit performance is considered as the most impactful metric. Since the following goal is to reduce the positioning effort, 1NN classifier was eliminated from consideration due to the lengthy PT. Of the remaining two models, NN predicted the correct building in 16 ms, while LSVM in 107 ms, marking it as the best-performing BH model. As a downside, according to our evaluation, NN required 6.28 s to train the model, while LSVM was trained in 1.34 s and k-NN does not require any model training at all.

Floor Hit (FH)

The next validation step finds the most suitable FH classifier by performing a sweep over all datasets. The multi-building datasets, UJI 1 and UJI 2, were split depending on the samples' building labels, forming 6 smaller databases in total (denoted as e.g. UJI 1$_1$ for first building of the UJI 1 dataset). This step effectively follows the 100% BH split from the previous step, achieved by any of the three best-performing methods (k-NN, NN and LSVM).

The results of the performed experiment are presented in Table III, showing the full results of the validation accuracy sweep and the averaged results for the PTs. The highest FH is achieved by NN model, followed by LSVM and 1NN. The PT of the NN is second only to the DT model, whose accuracy is lower by 4.3%. Consequently, NN is chosen as the FH classification model.

2D Localization (2D-L)

In order to find the precise 2D location in the given building and on the given floor, the cascade algorithm runs a 2D regression. To determine the over-all best-performing regression method, we evaluated the results of the sweep across all regression models introduced in Section II. The averaged results of the sweep may be found in Table IV. We provide only the mean model performance across all floors, as the full table of results (82 floor datasets in total) is too extensive and it does not provide additional value to the paper.

TABLE IV: 2D Regression Results for Validation

	1NN	3NN	W3NN	11NN	W11NN	LSVM	DT	RF	NN	AdB
Validation Accuracy [m]										
Avg	**2.69**	3.27	**2.93**	5.64	4.18	8.90	4.03	4.90	5.04	8.06
Validation Prediction Time [ms]										
Avg	**0.08**	**0.08**	**0.07**	**0.07**	**0.08**	77.4	5.31	7.08	180.2	111.2
Number of floor datasets with the lowest 2D pos. error [-]										
	52	1	**20**	0	2	0	5	0	1	1

In all 14 available and analyzed datasets, there is a total of 82 floors distributed over 18 buildings. The overall best positioning accuracy was achieved by 1NN regression, followed by W3NN, with the lowest average positioning error of 2.69 and 2.93 meters over all datasets, respectively. Additionally, 1NN achieved the lowest positioning error for 52 out of 82 considered floor datasets and W3NN in 20 floors. The suitability of k-NN for utilization in fingerprinting in terms of good positioning accuracy has been already documented in literature [2], [10], [22], but the time required for prediction is usually very long for voluminous datasets, as it is highly dependent on the datasets size. It is usually considered to be the biggest downside of this method.

Nevertheless, considered k-NN regressors also performed the best in terms of PT required to finish the positioning. This is in contrast to k-NN's behaviour in previous stages (BH and FH validation), where its PT was by far the slowest (as expected) of all considered methods. This change in the behaviour was caused by the notable reduction of the dataset size due to the previous steps in the cascade sequence of the proposed method. Additionally, the method does not require any previous training, saving additional resources for the service provider.

C. Numerical Results - Testing

In this section, we provide the results of the chosen cascade model on the independent test datasets to provide an unbiased evaluation. NN is utilized as the best-performing BH and FH models, while 1NN and W3NN are utilized as two alternatives of the 2D-L model. The considered benchmark models are the stand-alone 1NN and W3NN models trained on the full training dataset. The results of the benchmark directly compare the performance of the cascade model to the stand-alone model with the same parameters.

Table V presents the results of the proposed solutions, as well as that of the corresponding benchmark models, displaying the raw values for the benchmarks and normalized ones for the proposed solution, directly showing the corresponding improvement or deterioration. We present the raw results for the benchmarks in terms of BH [%], FH [%], 2D positioning error ϵ_{2D} [m], 3-Dimensional (3D) positioning error ϵ_{3D} [m] and the PT [s], while we use the normalized results (denoted as $\tilde{BH}$, $\tilde{FH}$, $\tilde{\epsilon_{2D}}$, $\tilde{\epsilon_{3D}}$, and $\tilde{PT}$ [-]) to evaluate the proposed model's performance, similarly to [26]. We denote, that the normalized values are representing the ratios between the

proposed method's and the benchmark's results. Therefore, the normalized values lower than 1, represent the improvement for the entities we aim to decrease (ϵ_{2D}, ϵ_{3D} and PT), while the entities we aim to increase (BH and FH) are improved in cases, where the normalized value is higher than 1.

The best performing cascade in terms of validation accuracy was the combination of $\mathbf{NN} \to \mathbf{NN} \to \mathbf{1NN}$. The overall results show a slight improvement (by 4%) in FH and a slight increase of the positioning error (6%) by the proposed solution while using the same underlying model (1NN) as the benchmark. The most relevant improvement is observed in the comparison of the PT, which is improved by more than 70% compared to the benchmark, on average. Table V also shows, that when the more voluminous datasets are considered, the improvement in PT is even more substantial (e.g. the PT of the test samples in UJI 1 is reduced from 214 s to 2.87 s by applying the cascade).

We also provide the averaged normalized results of the second best cascade combination in terms of validation accuracy, namely NN $\to$ NN $\to$ W3NN. The results are normalized towards W3NN benchmark in order to provide unbiased comparison to the plain use of the single model. Although, in comparison with the previous combination, the overall FH has been further improved (by an additional 1% due to more favorable random initialization) and the PT has decreased 6% more, the average positioning accuracy has dropped by 2%, adding up to the 8% drop in 3D positioning error in comparison to the use of 1NN regression. Although the performance of the two is comparable, the trade-off is unfavorable in general.

Additionally, we offer a cascade model combination of DT $\to$ DT $\to$ W3NN, which represents the combination of the fastest methods in the validation. This cascade provides a drastic decrease in PT, as it requires only 16% of time in comparison to the W3NN benchmark, on average. Specifically, the PT of the aforementioned test set in UJI 1 is only 2.38 s. Nevertheless, this cascade increases the positioning error by 25% compared to the benchmark, which may not be an acceptable trade-off for many applications.

The dependency of the PT on the number of samples in the training dataset is visualized in Fig. 2, along with their linear expectation. The prediction time of the proposed cascade model (blue in Fig. 2) splits each database into smaller parts, thus effectively reduces the k-NN's search space and consequently the PT of the solution. While the results of the benchmark show the steady increase in the prediction time with the increasing volume of the training dataset, the number of samples itself is not the only parameter affecting the k-NN's complexity, as e.g. the number of APs and the chosen distance metric impact the complexity of the calculation as well.

D. Discussion

Section III first presents a sweep over numerous ML methods, then based on the performance, we choose the best performing ones. The procedure is logical, and the final models are the representatives of the most popular models from the literature, namely NN or k-NN. The logical conclusion is

TABLE V: Numerical results of the benchmark and the proposed cascade of models

	BH [%]	FH [%]	ϵ_{2D} [m]	ϵ_{3D} [m]	PT [s]	$\tilde{B}H$ [-]	$\tilde{F}H$ [-]	$\tilde{\epsilon}_{2D}$ [-]	$\tilde{\epsilon}_{3D}$ [-]	$\tilde{P}T$ [-]
			Baseline				NN $\to$ NN $\to$ 1NN			
LIB 1		99.84	3.03	3.04	2.91		1	1.11	1.10	0.82
LIB 2		97.72	4.13	4.20	3.25		1.02	0.93	0.91	0.83
SAH 1		46.79	8.26	9.05	13.6		1.26	0.97	0.95	0.12
TIE 1		60.00	6.20	7.16	3.94		1	0.92	0.92	0.10
TUT 1		90.00	9.50	9.60	1.24		1	1.03	1.03	0.35
TUT 2		72.73	12.71	12.89	0.22		1.10	1.23	1.23	0.65
TUT 3		91.62	9.49	9.59	42.8		1.03	1.05	1.04	0.12
TUT 4		95.27	6.34	6.41	42.7		1.01	1.07	1.07	0.05
TUT 5		88.39	6.84	6.92	1.41		1.08	1.18	1.17	0.60
TUT 6		99.99	1.96	1.96	96.7		1	1.06	1.06	0.20
TUT 7		99.19	2.34	2.35	178		1	1.02	1.02	0.13
UJI 1	100	87.76	10.73	10.83	214	1	1.03	0.90	1.04	0.01
UJI 2	100	85.34	7.87	8.05	572	1	1.02	1.20	1.19	0.03
UTS 1		92.78	8.38	8.76	13.1		1.04	1.09	1.05	0.04
Avg						1	1.04	1.06	1.06	0.29
							NN $\to$ NN $\to$ W3NN			
Avg						1	1.05	1.09	1.08	0.23
							DT $\to$ DT $\to$ W3NN			
Avg						1	0.96	1.25	1.25	0.16

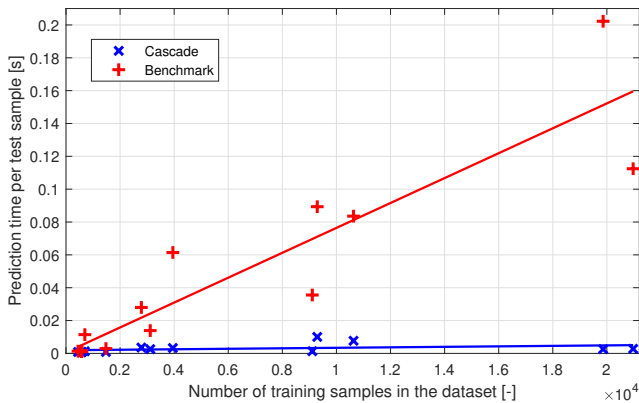
Fig. 2: Comparison of a test sample prediction speed between the NN $\to$ NN $\to$ 1NN model and the 1NN benchmark.

that their ever-present utilization is well-justified. The NNs offer the unmatched generalization properties, while k-NN and its alternatives offer straightforward matching approach. Nevertheless, the remaining models' performance was closely behind the chosen ones (NN and k-NN), outperforming them in several evaluation metrics, such as DT having the fastest PT. The results of this work prove the effectiveness of the mainstream ML methods, while highlighting the vast options a data scientist has in terms of model variety. Additionally, we show that by utilizing the models in cascade to iteratively filter the samples, the 2D-L 1NN model's PT can be drastically reduced.

IV. CONCLUSION

This work proposes an efficient strategy for accelerating the RSS positioning in indoor scenarios. The proposed solution utilizes a cascade of ML models to perform the positioning task, while drastically saving computational resources and achieving comparable positioning accuracy to the benchmark. The proposed system model utilizes the building and floor labels to iteratively reduce the search space of the consecutive models.

In the evaluation, we performed the comparison of a wide variety of ML models for each task, including 9 distinct classifiers and 7 regressors, while comparing their performance on 14 datasets. As the result of the exhaustive evaluation, the work distinguished the cascade of neural network (i.e., the first stage in the cascaded architecture), neural network (second stage) and 1NN (third stage) for building classification, floor classification, and 2D localization, respectively, as the combination with the best positioning performance. The second highlighted model was a cascade of two decision trees for the first two stages and weighted 3NN for the third cascaded stage, as the fastest-performing combination.

The results on the independent test sets showed that the strategy of sequentially splitting the voluminous datasets enables absolute freedom in selecting the optimal model for each task, while the 2D localization model still efficiently utilizes the highly accurate k-NN, without the drawback of a lengthy prediction. The proposed best-performing model reduced the Prediction Time by 71% and improved the Floor Hit by 4%, on average, while only slightly deteriorating the 2D positioning error, when compared to the stand-alone 1NN benchmark. The results for the voluminous dataset UJI 1 demonstrated 100-fold decrease in prediction time. Additionally, the fastest-performing model was able to reduce the Prediction Time by 84% on average, at the cost of 4% lower Floor Hit and 25% lower positioning accuracy.

REFERENCES

[1] A. Ometov, V. Shubina, L. Klus, J. Skibińska, S. Saafi, P. Pascacio, L. Flueratoru, D. Q. Gaibor, N. Chukhno, O. Chukhno *et al.*, "A survey on wearable technology: History, state-of-the-art and current challenges," *Computer Networks*, vol. 193, p. 108074, 2021.

[2] L. Klus, D. Quezada-Gaibor, J. Torres-Sospedra, E. S. Lohan, C. Granell, and J. Nurmi, "Rss fingerprinting dataset size reduction using feature-wise adaptive k-means clustering," in *2020 12th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*. IEEE, 2020, pp. 195–200.

[3] G. M. Mendoza-Silva, P. Richter, J. Torres-Sospedra, E. S. Lohan, and J. Huerta, "Long-term wifi fingerprinting dataset for research on robust indoor positioning," *Data*, vol. 3, no. 1, 2018.

[4] J. Torres-Sospedra, R. Montoliu, A. Martínez-Usó, J. P. Avariento, T. J. Arnau, M. Benedito-Bordonau, and J. Huerta, "UJIIndoorLoc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems," in *Int. Conf. on Indoor Positioning and Indoor Navigation*, 2014, pp. 261–270.

[5] E. S. Lohan, J. Torres-Sospedra, and A. Gonzalez, "WiFi RSS measurements in Tampere University multi- building campus, 2017," Aug. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.5174851

[6] J. Torres-Sospedra, D. Quezada-Gaibor, G. M. Mendoza-Silva, J. Nurmi, Y. Koucheryavy, and J. Huerta, "New cluster selection and fine-grained search for k-means clustering and wi-fi fingerprinting," in *2020 International Conference on Localization and GNSS (ICL-GNSS)*, 2020, pp. 1–6.

[7] J. Rojo, G. M. Mendoza-Silva, G. R. Cidral, J. Laiapea, G. Parrello, A. Simó, L. Stupin, D. Minican, M. Farrés, C. Corvalán *et al.*, "Machine learning applied to wi-fi fingerprinting: The experiences of the ubiqum

challenge," in *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2019, pp. 1–8.

[8] B. Ezhumalai, M. Song, and K. Park, "An efficient indoor positioning method based on wi-fi rss fingerprint and classification algorithm," *Sensors*, vol. 21, no. 10, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/10/3418

[9] K. Maneerat and K. Kaemarungsi, "Roc: Robust and low-complexity wireless indoor positioning systems for multifloor buildings using location fingerprinting techniques," *Mobile Information Systems*, vol. 2019, pp. 1–22, 02 2019.

[10] J. Torres-Sospedra, P. Richter, A. Moreira, G. Mendoza-Silva, E.-S. Lohan, S. Trilles, M. Matey-Sanz, and J. Huerta, "A comprehensive and reproducible comparison of clustering and optimization rules in wi-fi fingerprinting," *IEEE Trans. on Mobile Computing*, 2020.

[11] R. Klus, L. Klus, J. Talvitie, J. Pihlajasalo, J. Torres-Sospedra, and M. Valkama, "Transfer learning for convolutional indoor positioning systems," in *2021 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2022, pp. 1–8.

[12] X. Song, X. Fan, C. Xiang, Q. Ye, L. Liu, Z. Wang, X. He, N. Yang, and G. Fang, "A novel convolutional neural network based indoor localization framework with wifi fingerprinting," *IEEE Access*, vol. 7, pp. 110 698–110 709, 2019.

[13] T. Hastie, S. Rosset, J. Zhu, and H. Zou, "Multi-class adaboost," *Statistics and its Interface*, vol. 2, no. 3, pp. 349–360, 2009.

[14] H. Zhang, "The optimality of naive bayes," *Aa*, vol. 1, no. 2, p. 3, 2004.

[15] P. Bahl and V. Padmanabhan, "Radar: an in-building rf-based user location and tracking system," in *Proceedings IEEE INFOCOM 2000*, vol. 2, 2000, pp. 775–784 vol.2.

[16] F. Lemic, V. Handziski, M. Aernouts, T. Janssen, R. Berkvens, A. Wolisz, and J. Famaey, "Regression-based estimation of individual errors in fingerprinting localization," *IEEE Access*, vol. 7, pp. 33 652–33 664, 2019.

[17] G. G. Anagnostopoulos and A. Kalousis, "A reproducible comparison of rssi fingerprinting localization methods using lorawan," in *16th Workshop on Positioning, Navigation and Communications*, 2019.

[18] J. Purohit, X. Wang, S. Mao, X. Sun, and C. Yang, "Fingerprinting-based Indoor and Outdoor Localization with LoRa and Deep Learning," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*. IEEE, 12 2020, pp. 1–6.

[19] N. Marques, F. Meneses, and A. Moreira, "Combining similarity functions and majority rules for multi-building, multi-floor, wifi positioning," in *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2012, pp. 1–9.

[20] L. Klus and et al. (2022, June) Supplementary materials for "Towards Accelerated Localization Performance Across Indoor Positioning Datasets". [Online]. Available: https://doi.org/10.5281/zenodo.6421461

[21] A. Razavi, M. Valkama, and E.-S. Lohan, "K-means fingerprint clustering for low-complexity floor estimation in indoor mobile localization," in *2015 IEEE Globecom Workshops*, 2015.

[22] A. Cramariuc, H. Huttunen, and E. S. Lohan, "Clustering benefits in mobile-centric wifi positioning in multi-floor buildings," in *2016 International Conference on Localization and GNSS*, 2016.

[23] E.-S. Lohan, J. Torres-Sospedra, H. Leppäkoski, P. Richter, Z. Peng, and J. Huerta, "Wi-fi crowdsourced fingerprinting dataset for indoor positioning," *MDPI Data*, vol. 2, no. 4, 2017.

[24] P. Richter, E. S. Lohan, and J. Talvitie. WLAN (WiFi) rss database for fingerprinting positioning. [Online]. Available: https://zenodo.org/record/1161525

[25] Lohan. (2020, May) Additional TAU datasets for Wi-Fi fingerprinting- based positioning. [Online]. Available: https://doi.org/10.5281/zenodo.3819917

[26] J. Torres-Sospedra, I. Silva, L. Klus, D. Quezada-Gaibor, A. Crivello, P. Barsocchi, C. Pendão, E. S. Lohan, J. Nurmi, and A. Moreira, "Towards ubiquitous indoor positioning: Comparing systems across heterogeneous datasets," in *2021 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2021, pp. 1–8.

[27] D. Quezada-Gaibor, L. Klus, J. Torres-Sospedra, E. S. Lohan, J. Nurmi, and J. Huerta, "Improving dbscan for indoor positioning using wi-fi radio maps in wearable and iot devices," in *2020 12th International Congress on Ultra Modern Telecommunications and Control Systems*, 2020, pp. 208–213.