DeepLocBIM: Learning Indoor Area Localization Guided by Digital Building Models

Marius Laska and Jörg Blankenbach, *Member, IEEE*

***Abstract*—Fingerprinting-based indoor localization is a cost-effective approach to provide coarse-grained indoor location information for pedestrian mass-market applications without the requirement of installing additional positioning infrastructure. While most solutions aim at pinpointing the exact location of a user, estimating a zone/area is a promising approach to achieve a more reliable prediction. Area localization predominantly utilizes a predetermined building model segmentation to obtain zone/area labels for collected fingerprints. We propose a novel approach to multifloor indoor area localization by directly predicting polygon zones that contain the position of the user. Our model learns to construct the zones from the wall segments and thus predicted areas have a high conformity to the underlying building model (semantic expressiveness). On a self-collected as well as on a public fingerprinting data set, we compare our model with two reference approaches. We demonstrate that the utilized surface areas of the polygons are on average up to 50% smaller than those of the reference models and provide a high semantic expressiveness without requiring manual floor plan segmentation.**

***Index Terms*—Area localization, building information modeling (BIM), deep learning, fingerprinting, indoor localization, multifloor.**

I. Introduction

OBTAINING the location of a user or entity is the key requirement to realize location-based services (LBSs). Within indoor environments, common localization approaches that utilize global navigation satellite systems (GNSSs) do not function as required, since the signals are attenuated by building components [1]. Due to the estimated potential of indoor location-based services [2], the development of alternative indoor localization approaches has emerged as a research field of particular interest. Several technologies [e.g., WLAN, Bluetooth, and ultrawideband (UWB)] have been used with a variety of techniques [received signal strength (RSS), Time of Arrival (ToA), fingerprinting, etc.] [1]. Their main discriminative features are the provided localization accuracy and reliability as well as the accompanied costs due to dedicated infrastructure, development or maintenance of the system. Indoor localization systems that utilize UWB as base technology provide the highest accuracy (submeter). However, they require the installation of several dedicated UWB transmitters and localization devices that support UWB. Therefore, they are mainly suitable for deployments where accuracy is critical (e.g., industrial context) [3].

On the other hand, WLAN fingerprinting provides a much lower localization accuracy but usually does not require additional hardware to be installed [4]. Instead, it uses the existing RSS of installed WLAN access points (APs). Those differ based on the location within a building, which can be used as a location fingerprint to estimate the position. In an offline phase, the map is scanned at certain reference locations and the collected fingerprints are stored within a database. Those are used to build a model that learns the relation between fingerprints and location and is capable of estimating the location for unseen fingerprints during the online phase [4].

Most fingerprinting-based indoor localization systems aim to pinpoint the exact location of a user, which is challenging since, indoors, RSS suffers from several radio signal propagation defects such as multipath propagation [1]. However, especially for many LBS aimed at pedestrians (e.g., within a shopping mall), coarse-grained localization is sufficient under the premise that the system is easily deployable. Therefore, fingerprinting-based solutions have been proposed that explicitly determine the area/zone the user is located at [5].

Usually, the floor plan is segmented into areas (e.g., rooms) during preprocessing. The fingerprints are then partitioned into classes that serve as labels for training a classifier. The segmentation of the environment clearly affects the resulting classification accuracy as well as the knowledge gain of the user (expressiveness). On a really coarse-grained segmentation, a classifier might offer high classification accuracy but few advantages for the user. However, a fine-grained segmentation yields a high knowledge gain for the user but might cause a lower classification accuracy. Furthermore, the conformity to the underlying building structure affects the expressiveness of the model. Area predictions that are in compliance with the rooms, walls, etc., yield additional knowledge gain for the user.

Determining an *a-priori* segmentation before training an area classifier has several limitations: 1) it fixes the introduced tradeoff between accuracy and expressiveness of the model; 2) semantic segmentation (e.g., based on rooms) might be too fine-grained for large-scale settings and result in a low classification accuracy; 3) adaptation to new data might require resegmentation followed by time-consuming model retraining; and 4) determining a segmentation that ensures good separation in signal space while adhering to the floor plan structure is challenging [6]. Previously, we introduced a model DeepLocBox (DLB) [7] that directly estimates an area via bounding box regression, however, the boxes do not match

Manuscript received 23 July 2021; revised 18 November 2021; accepted 27 January 2022. Date of publication 7 February 2022; date of current version 8 August 2022. *(Corresponding author: Marius Laska.)*

The authors are with the Geodetic Institute and Chair for Computing in Civil Engineering and Geo Information Systems, RWTH Aachen University, 52074 Aachen, Germany (e-mail: marius.laska@gia.rwth-aachen.de).

Digital Object Identifier 10.1109/JIOT.2022.3149549

with the existing building structure, which limits its *semantic* expressiveness.

In this work, we introduce DeepLocBIM that leverages the digital building model (building information modeling (BIM) model) to achieve a higher semantic expressiveness while preserving the benefit of no *a-priori* floor plan segmentation. The model learns to estimate the area of the user which it models as a polygon constructed from the walls that are automatically obtained from the underlying BIM model. The model is suitable for multibuilding/multifloor localization and provides a much higher expressiveness than the standard approach of a classifier trained on a predetermined floor plan segmentation.

In summary, we claim the following contributions.

1) We propose a novel model (DeepLocBIM) for fingerprinting-based indoor area localization.
2) Instead of classifying a presegmented area, DeepLocBIM directly constructs polygon areas, which gives the model a higher expressive power.
3) In contrast to its predecessor DLB [7], our new model leverages the digital building model and thus its predictions are more valuable, since they match with the structure of the building (e.g., rooms).
4) DeepLocBIM can be used within multifloor settings and its output can be interpreted as an ensemble that delivers an uncertainty measure for the floor classification task.
5) At the same level of accuracy, our model achieves higher geometric and semantic expressiveness compared to related works.

The remainder of this article is structured as follows. In Section II, related work is introduced with a focus on deep learning for fingerprinting, area localization, and the potential of digital building models for indoor localization. Section III defines the quality constraints of area localization models. In Section IV, our model (DeepLocBIM, DLBIM) is formally introduced followed by a brief overview of two reference models in Section V. Subsequently, DLBIM is evaluated against the reference models in Section VI and the results are concluded in Section VII.

II. Related Work

A. Deep Learning for Fingerprinting

The success of deep learning in a variety of fields has encouraged the application of related methods in the domain of fingerprinting-based indoor localization [8]. The fingerprinting method is applied in device-free [9], [10] as well as in device-based [11], [12] settings, while we focus on the latter one in this article.

The majority of fingerprinting localization approaches rely on RSS for fingerprint construction. Those have the benefit of being accessible via commodity smartphones. It was demonstrated that deep neural networks can effectively cope with the imperfections of RSS and outperform traditional methods such as k-nearest neighbor [7], [13]. A variety of deep models have been investigated. The unprocessed RSS vector can be used as input for feedforward neural networks [13], [14] or be preprocessed to obtain a low-dimensional embedding via stacked autoencoders (SAEs) [12], [15]. Convolutional neural networks can be applied using 1-D convolution on time-series RSS [16] or directly over the RSS vector [12]. By constructing 2-D input images from the RSS vectors, 2-D CNNs adopted from the field of image classification can be used [11]. Instead of predicting static independent locations, the continuous positions of a user can be modeled as trajectory. Those trajectories have been used for training long short-term memory (LSTM) deep networks [17], [18].

B. Area Localization/Zone Detection

Pinpointing the exact position with RSS-based fingerprinting is hardly possible due to the characteristics of RSS. However, many location-based services do not require exact position estimation [5], which led to several publications that try to predict the area/zone of location rather than the exact position of the user. Most approaches construct predefined areas, which are then used to partition the fingerprints into classes that serve as labels for training a classifier. The construction of predefined areas can be done by either: 1) ignoring the floor plan and the collected fingerprints (e.g., grid based); 2) utilizing the floor plan structure (e.g., room-based partitioning); or 3) using the fingerprints to partition the space into classes that can be separated well in the signal space (e.g., via clustering). Certain works follow a hybrid approach of combining 1) and 3). Note that the segmentation is a preprocessing step, which means that although a segmentation might be done without considering the collected fingerprints, they are still incorporated during classification as in the classical fingerprinting approach. Finally, there exist works, where the areas are not determined beforehand to apply a classification model, but the model directly outputs the area/zone during prediction (e.g., bounding box regression). In the following, we will discuss relevant approaches following the introduced classification framework.

1) RSS and Floor Plan Independent Presegmentation: The partitioning into areas can be done without using the collected fingerprints (RSS) nor knowledge about the floor plan. Yean *et al.* [19] followed a grid-based zone localization approach. They propose a random forest as an end-to-end pipeline with auto-tuned hyperparameters. The system is evaluated on a self-collected data set and on the public data set. Chiriki *et al.* [20] realized multiclass zoning localization by applying support vector machines (SVMs). The model is evaluated on two closed data sets where details on the construction of the zones are missing.

2) Floor Plan Aware Presegmentation: Several approaches utilize the floor plan structure to obtain a partitioning during preprocessing [5], [21], [22]. The resulting areas are mostly composed of rooms or halls. This has the benefit that the areas classified by the system provide a high semantic expressiveness. Liu *et al.* [21] proposed an algorithm for probability estimation over possible areas by incorporating the user's trajectory and existing map information to filter unreasonable results. Anzum *et al.* [5] similarly constructed zones in open areas/spaces and train a counter propagation neural network to classify the correct zone. AlShamaa *et al.* [22] applied belief

functions theory to predict the correct zone and evaluate the system in a small-scale setting.

Another approach is to use the rooms to partition an indoor environment, which is especially suitable for large-scale deployments within shopping malls. Lopez-pastor *et al.* [23] obtained labeled fingerprints by randomly walking within a shop. Similarly, Wei *et al.* [24] collected the fingerprint while a customer is paying via a smartphone application. They utilize the fact that the location is known at that point. Rezgui *et al.* [25] proposed a normalized rank-based SVM to cope with diverse collection devices to predict the shop within a mall. Zhang *et al.* [26] applied a stacked denoising autoencoder variation to reduce the fingerprint dimension and use gradient boosting decision trees for shop classification.

3) RSS Aware Presegmentation: In fingerprinting solutions, the RSS signals are used to determine the location. Therefore, their discriminative characteristics can be used to obtain the underlying floor plan segmentation. Salazar *et al.* [27] assigned fingerprints to zones via fuzzy *c*-means clustering. A fuzzy inference system is subsequently used to determine the zone-level localization. Zhang *et al.* [28] divided the map into zones by *k*-means clustering on the RSS signals. Subsequently, they train an extreme learning machine to classify the correct zone. The method is compared to manual segmentation and reference point classification. Laska *et al.* [6] proposed an iterative adaptive segmentation of the floor plan using crowdsourced data that is perpetually collected. The segmentation is obtained by using the floor plan structure and the RSS signals, whereas its granularity can be adjusted toward the user's demands. A hierarchical clustering is utilized by Hernández *et al.* [29] together with a classifier ensemble to detect the correct zone on each level of the hierarchical clustering result.

4) Without Predetermined Segmentation: An *a-priori* floor plan segmentation limits the expressiveness of a model, since it only allows it to predict one of the presegmented areas. Therefore, Laska and Blankenbach [7] proposed DLB, an area/zone estimation that does not require a predetermined floor plan segmentation. Instead, it directly estimates a bounding box that contains the ground-truth location of the user. The box size depends on the estimated prediction error. DLB does not consider the building structure, which is tackled by the model proposed in this article.

C. Building Information Modeling for Indoor Localization

Digitalization and automation are among the most important trends in the architecture, engineering, and construction (AEC) industry. A key element in this context is BIM [30], [31]. BIM refers to a novel working method that relies on digital models, which are used for the consistent collection, management, and exchange of all building-related data over the entire life cycle including also the creation of BIM models for existing buildings [32], [33].

Most indoor localization systems are targeted at providing localization services within a distinct building. Especially, for infrastructure-based localization, already installed sensing networks such as WLAN APs or nodes installed exclusively for localization such as BLE beacons or UWB transmitters heavily affect the resulting localization accuracy as well as the costs of the systems. Having information on the building of interest can thus be effectively used to support planning and installation of the system, as well as to impose geometric constraints on developed localization algorithms [34]. In the field of automation in construction, BIM has gained momentum over the past years [30]–[33] and its potential for enhancing indoor localization is increasingly acknowledged in the field [35]. Liu *et al.* [35] provided a recent review of potential application of BIM in the context of indoor localization and navigation. They identify the generation of navigation models as the main application of BIM followed by the direct application of BIM for enhancing indoor localization models. Li *et al.* [34] proposed a system for localizing trapped occupants during fire emergencies. BIM is used to provide building-aware beacon placement to minimize the amount of nodes of the ad-hoc network while keeping room-level localization accuracy. Herbers and König [36] utilized augmented reality devices for spatial mapping and point cloud matching of digital building models. Ha *et al.* [37] tackled the problem of missing labeled training data for image-based indoor localization. They train a CNN on rendered images of a BIM model to predict the position from real images during the localization phase. Chang *et al.* [38] proposed a system to simulate the virtual sound field through BIM and train a CNN for sound-based indoor localization.

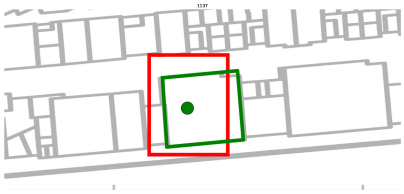


Fig. 1. Illustration of semantic expressiveness of two area/zone predictions. The green point represents the ground-truth location.

III. Quality of Area Localization Model

The nature of the predicted areas of an area localization model affects the overall quality of the model. In [6], we introduced the concepts of *performance* and *expressiveness* for quantifying the quality of indoor area localization models. The model performance characterizes the classification ability of the model, which can be measured by the classification accuracy or related metrics such as the F1-score. The expressiveness represents the knowledge gain of the user obtained by the predictions of the model. It was stated that coarse-grained area localization provides a lower knowledge gain than fine-grained area localization. As a consequence, the average surface area of predicted areas/zones has been proposed as a metric to quantify the expressiveness. Finally, the area classification score (ACS) was introduced to capture the interplay between performance and expressiveness, which can be parameterized to favor models at each end of the spectrum [6].

In this work, we want to stress another component that affects the expressiveness of area localization models. If predicted areas have a high conformity to the underlying floor plan, the expressiveness of the model is higher. Assume the two predictions visualized in Fig. 1. Although both predictions

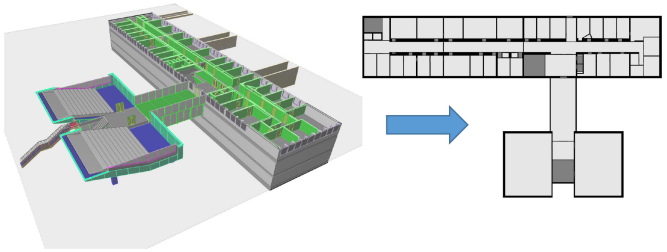


Fig. 2. Horizontal section (gray plane) in a BIM model for obtaining the vectorial footprint of the walls as basis for the algorithm.

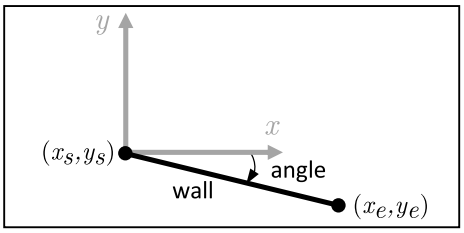


Fig. 3. Angle for classifying walls into horizontal/vertical.

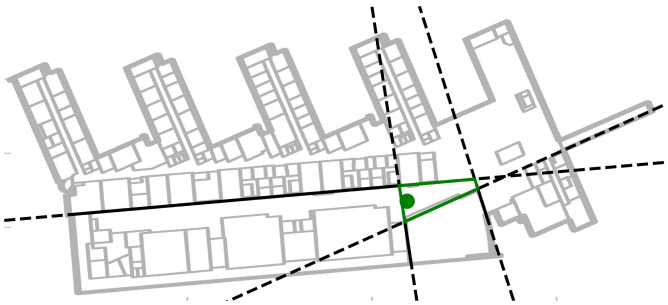


Fig. 4. Polygon construction.

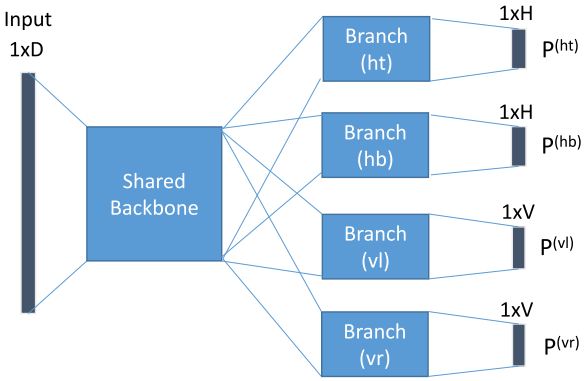


Fig. 5. Base architecture of the DeepLocBIM neural network.

have a similar surface area, the green area prediction clearly provides more benefit to the user, since it exactly matches the room of the floor plan. In contrast, the red prediction indicates that the real position might as well be in the hall above.

Therefore, we further refine expressiveness into *geometric* and *semantic* parts. Geometric expressiveness is quantified by the average predicted surface area of the model, whereas semantic expressiveness is given as the conformity to the underlying floor plan. The latter is hard to quantify and can only be qualitatively assessed. If we manually partition the floor plan, we automatically guarantee a level of semantic expressiveness. Otherwise, the underlying floor plan should be incorporated during the prediction or segmentation to achieve semantic expressiveness.

IV. DeepLocBIM (DLBIM)

A. Intuition

The goal of our model is to provide an area estimation with high geometric and semantic expressiveness without the prerequisite of a presegmented floor plan. Previous area estimation models, which do not require prior floor plan segmentation, such as DLB [7] suffer from low semantic expressiveness, since the predictions have no relationship to the underlying building structure. In this work, we propose a novel approach to indoor area localization. The key idea is that the model learns to predict zones as 2-D polygons containing the ground-truth location of the user. Those are defined by four distinct boundary wall segments, which are a chosen subset among the building walls. The building walls are extracted from a vectorized floor plan, which can be derived from a horizontal section of a BIM model as depicted in Fig. 2. The set of vectorized walls is then divided into horizontal and vertical ones and skeletonized subsequently, such that each wall can be represented by a 4-tuple (x_s, y_s, x_e, y_e) consisting of start and end point of the wall. Since building walls can be oblique, we define the separation between horizontal and vertical walls as follows. Let (x_s, y_s) be the origin of a Cartesian coordinate system. The wall is said to be horizontal if the absolute angle between $\vec{w} = (x_e, y_e)^T$ and the unit vector $(1, 0)^T$ is smaller than 45°; otherwise, it is classified as vertical. The defined angle is visualized in Fig. 3. The model selects a boundary wall segment for each direction (top, bottom, left, and right). By finding the intersection points of the vectors defined by the line segments, we can construct the resulting polygon, which is exemplarily shown in Fig. 4.

B. Base Architecture

We assume a fingerprinting data set $\mathcal{D} = \{fp_n = (\mathbf{a}_n, \mathbf{l}_n)\}$ for $n = 1, \ldots, N$, where $\mathbf{a}_n$ is a D-dimensional fingerprint $\mathbf{a}_n = (a_1, \ldots, a_D)^T$ that contains the RSS of the APs and the tagged position $\mathbf{l}_n = (l_x, l_y)^T$ of the fingerprint. To implement our model, we utilize a deep neural network, which produces four separate outputs: $\mathbf{p}^{(ht)} \in \mathbb{R}^H$ is the predicted probability distribution over H possible horizontal walls for the top boundary line segments. $\mathbf{p}^{(hb)} \in \mathbb{R}^H$ is analogously defined for the bottom boundary line segments. $\mathbf{p}^{(vl)} \in \mathbb{R}^V$ and $\mathbf{p}^{(vr)} \in \mathbb{R}^V$ are the predicted probability distribution over V possible vertical walls for the left and right boundary line segments. The network input is given by the D-dimensional fingerprint vector $\mathbf{a}_n$ that holds the RSS to each AP. The base architecture of the model is depicted in Fig. 5, where light blue components are modular and dark blue parts are fixed. The model utilizes a shared backbone neural network to extract knowledge that is relevant across all outputs. The shared network is followed by a separate neural network branch for producing the various outputs. Finally, each output branch produces

a probability distribution over the possible wall segments via softmax activation, where the size of the outputs depends on the number of possible walls (H for top and bottom segment and V for left and right segments). The exact neural network architectures of the modular components remain interchangeable and might depend on the given data set. In Section VI-B, we present the chosen architecture of DLBIM for the studied data sets.

C. Target Vector Construction

In order to guide the learning process of the network, we supply the model with separate target probability vectors ($\mathbf{y}^{(ht)}$, $\mathbf{y}^{(hb)}$ and $\mathbf{y}^{(vl)}$, $\mathbf{y}^{(vr)}$) over the possible choices of walls. Intuitively, the probability mass should be concentrated among the closest walls to the ground truth position and weighted by their distance d to it. Furthermore, we also want to consider walls for which the projected point does not lie on the wall but its distance r to the wall's closest endpoint is small enough. We will only cover the construction of the top and bottom target vectors ($\mathbf{y}^{(ht)}$, $\mathbf{y}^{(hb)}$) with horizontal walls ($\mathbf{W}^{(h)}$) being possible candidates. The remaining two target vectors ($\mathbf{y}^{(vl)}$, $\mathbf{y}^{(vr)}$) can be derived analogously.

To determine the parameters d and r, we do the following. We project the fingerprint location $\mathbf{l} = (l_x, l_y)^T$ onto the walls to obtain $\mathbf{pr} = (\mathrm{pr}_x, \mathrm{pr}_y)^T$. Let $w_h = (x_s, y_s, x_e, y_e)$ be a tuple defining the end and start points of any horizontal wall onto which we project $\mathbf{l}$. We want to perform an orthogonal projection, which requires that w_h is defined as vector in a 2-D vector space. By translating its start point to the origin, we obtain the vectors

$$\mathbf{v_h} = (x_e - x_s, y_e - y_s)^T \tag{1}$$

$$\mathbf{p}' = \left(l_x - x_s, l_y - y_s\right)^T. \tag{2}$$

Now, we can project $\mathbf{p}'$ onto $\mathbf{v_h}$

$$\mathcal{P}' = \frac{\mathbf{v_h} \cdot \mathbf{p}'}{\mathbf{v_h} \cdot \mathbf{v_h}} \mathbf{v_h} \tag{3}$$

and finally, obtain the projection in the original coordinate system as

$$\mathbf{pr} = \mathcal{P}' + (x_s, y_s)^T. \tag{4}$$

Subsequently, we can calculate $d = \mathrm{pr}_y - l_y$ and $r = \min\{|\mathrm{pr}_x - x_e|, |\mathrm{pr}_x - x_s|\}$. The computation of d and r is visualized in Fig. 6, where (a) shows a point-based representation of the parameters and (b) illustrates the computation of the projection (red point) by moving the origin of the coordinate system to (x_s, y_s) and applying orthogonal projections. Using (4), we can obtain the projected point (b) within the original coordinate system (a) and compute d and r.

In case that d is smaller than max_dist and $\mathbf{pr}$ lies on the wall or r is smaller than max_range, we set the entry in one of the target vectors of that specific wall to d; otherwise, we set it to 0. The sign of d is used to differentiate between, whether we set the entry in $\mathbf{y}^{(ht)}$ or $\mathbf{y}^{(hb)}$. For the horizontal walls, a positive value indicates that the wall lies above the fingerprint position and thus represents a candidate for the top boundary segment, the opposite holds for a negative value. We repeat the process for all horizontal walls. Afterward, we

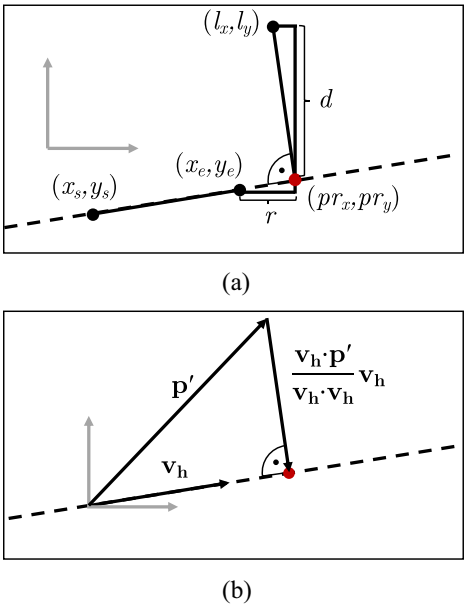


Fig. 6. Visualization of derivation of parameters d and r that are used for target vector construction. (a) Point-based representation within original coordinate system. (b) Vector-based method to obtain pr by using linear projections, where the origin (gray arrows) is moved to (x_s, y_s).

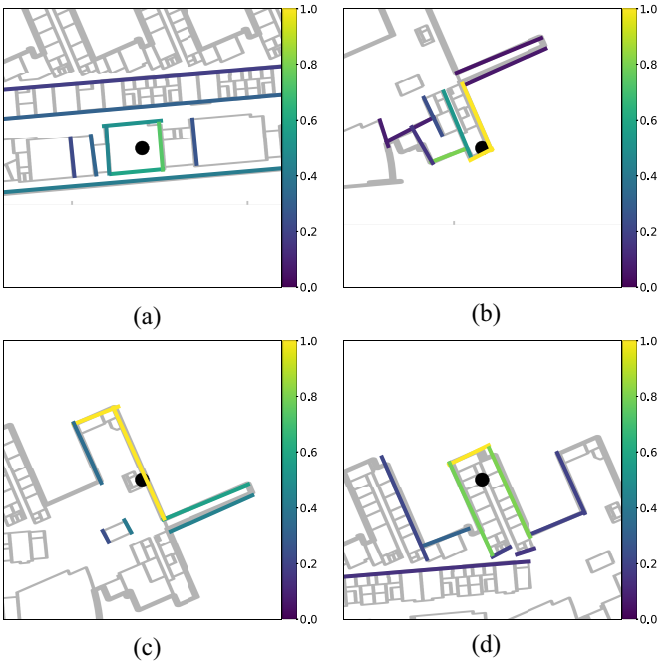


Fig. 7. Visualization of target vector construction. (a)–(d) Resulting target vector constructions for four example fingerprint positions. The color indicates the probability mass of the wall within the target vector.

invert nonzero entries of the target vector and scale each entry by the sum of the previously inverted entries. This ensures that close walls receive a higher probability mass and that the resulting entries sum to 1 and consequently form a valid probability distribution. Fig. 7(a)–(d) visualizes the target vectors for four sample fingerprints, where we set max_dist = 25 and max_range = 10. The color of each wall indicates its probability mass within the target vector. Note that, in each subfigure, four target vectors ($\mathbf{y}^{(ht)}$, $\mathbf{y}^{(hb)}$, $\mathbf{y}^{(vl)}$, $\mathbf{y}^{(vr)}$) are shown and only a subset of the building walls is considered

Algorithm 1 Get_Horizontal_Target_Vectors($\mathbf{l}$, $\mathbf{W}^{(h)}$)

Input: $\mathbf{l}$: fingerprint position, $\mathbf{W}^{(h)}$: horizontal walls
Output: $\mathbf{y}^{(ht)}$, $\mathbf{y}^{(hb)}$
Initialisation: $\mathbf{y}^{(ht)} = [|\mathbf{W}^{(h)}|]$, $\mathbf{y}^{(hb)} = [|\mathbf{W}^{(h)}|]$
for k=0 to $|\mathbf{W}^{(h)}|$ **do**
  $\mathbf{y}^{(ht)}[k]$, $\mathbf{y}^{(hb)}[k]$ = get_horizontal_score($\mathbf{l}$, $\mathbf{W}^{(h)}[k]$)
end for
for k=0 to $|\mathbf{W}^{(h)}|$ **do**
  if $\mathbf{y}^{(ht)}[k] > 0$ **then**
    $\mathbf{y}^{(ht)}[k] = (\mathbf{y}^{(ht)}[k])^{-1}$
  end if
end for
$\mathbf{y}^{(ht)}$ /= sum($\mathbf{y}^{(ht)}$)
for k=0 to $|\mathbf{W}^{(h)}|$ **do**
  if $\mathbf{y}^{(hb)}[k] > 0$ **then**
    $\mathbf{y}^{(hb)}[k] = (\mathbf{y}^{(hb)}[k])^{-1}$
  end if
end for
$\mathbf{y}^{(hb)}$ /= sum($\mathbf{y}^{(hb)}$)
return $\mathbf{y}^{(ht)}$, $\mathbf{y}^{(hb)}$

Algorithm 2 Get_Horizontal_Score($\mathbf{l}$, w)

Input: $\mathbf{l}$: fingerprint position, w: possible wall in target vector
Output: $d^{(ht)}$, $d^{(hb)}$
Initialisation: $d^{(ht)}$, $d^{(hb)} = 0$
$\mathbf{pr}$ = proj($\mathbf{l}$, w)
if lies_within_endpoints($\mathbf{pr}$, w) **or** dist_to_closest_endpoint($\mathbf{pr}$, w) < max_range **then**
  $d = \mathbf{pr}_y - \mathbf{l}_y$
  if |d_loss| < max_dist **then**
    if $d > 0$ **then**
      $d^{(ht)} = d$
    else
      $d^{(hb)} = d$
    end if
  end if
end if
return $d^{(ht)}$, $d^{(hb)}$

as possible candidates. The pseudocode for obtaining the top and bottom target vectors of the horizontal walls is given in Algorithms 1 and 2.

D. Loss

We want the loss of the model to incorporate the distance between the predicted probability distribution and the target probability distributions $\mathbf{y}^{(ht)}$, $\mathbf{y}^{(hb)}$, $\mathbf{y}^{(vl)}$, $\mathbf{y}^{(vr)}$ as described in the previous section. Therefore, we apply categorical cross-entropy loss

$$L = -\sum_{i=1}^{H} y_i^{(ht)} \cdot \log\left(p_i^{(ht)}\right) + y_i^{(hb)} \cdot \log\left(p_i^{(hb)}\right) - \sum_{j=1}^{V} y_j^{(vl)} \cdot \log\left(p_j^{(vl)}\right) + y_j^{(vr)} \cdot \log\left(p_j^{(vr)}\right). \quad (5)$$

Furthermore, we want to avoid *collapsing* polygon predictions. Those occur if either the predicted top and bottom wall segments or the predicted left and right wall segments are identical. Therefore, we add a regularization term to the loss, which we define as

$$R = \sum_{i=1}^{H} p_i^{(ht)} \cdot p_i^{(hb)} + \sum_{j=1}^{V} p_j^{(vl)} \cdot p_j^{(vr)}. \quad (6)$$

We obtain the final loss function as

$$\mathcal{L} = L + \lambda R \quad (7)$$

where λ controls the importance of the regularization term. The choice of λ is discussed in the evaluation section of the model.

E. Multifloor Setting

The model can be slightly modified to support multifloor environments, by letting the model choose among walls on any floor. Thus, the prediction of the floor is implicitly defined by the choice of the boundary wall segments. Given F floors, the size of the output vector for top and bottom boundary segments becomes $\sum_{f=1}^{F} H^{(f)}$, where $H^{(f)}$ is the amount of horizontal walls on the fth floor. Analogously, the size of the output vector for left and right boundary segments is given as $\sum_{f=1}^{F} V^{(f)}$. The implicit floor prediction can be treated as an ensemble classification. We take the majority of the floors associated with the predicted line segments. Furthermore, we can correct the walls that fall out of line according to the majority vote. For those, we can utilize the most likely prediction by limiting options to the walls located on the floor that is predicted by the majority of the ensemble. During the evaluation, we study the impact of noncoincidental ensemble predictions.

V. Reference Models

In order to compare the proposed approach to existing indoor area localization solutions, we briefly introduce two approaches that we will utilize as reference models. The first is based on a semantic floor plan segmentation, whereas the second one requires no predetermined floor plan segmentation.

A. Presegmented (Data and Floor Plan Aware) Area Classification

The floor plan is manually partitioned into nonoverlapping areas, where segments are constructed only within zones where data were collected. We do not consider the RSS value for partitioning as this might lead to overlapping areas that would limit the semantic expressiveness of the resulting classifier. Given the multifloor floor plan segmentation, we label the fingerprints with the segment that they are located in. We train a neural network-based classifier. Softmax is chosen as an output activation function and categorical cross-entropy loss for training the model. The floor is implicitly given by the location of the classified segment within the building. During subsequent evaluation, we refer to this model as AC.

B. DeepLocBox: Area Localization by Bounding Box Estimation

DLB [7] can be seen as the predecessor of DLBIM. It estimates the area of the user as a bounding box, where the width and height of the box should depend on the certainty of the model. It does not require a predetermined floor plan segmentation. In contrast to DLBIM, the floor plan structure is not utilized to determine the bounding boxes and thus the boxes have a low semantic expressiveness.

The box prediction output is defined by its center (c_x, c_y) and its width w and height h. The loss function of the model consists of two components: 1) the center loss captures the deviation of the predicted center of the box from the ground-truth point and 2) the size loss regulates the box dimensions. In order to support multibuilding/multifloor localization, the map is divided into grid cells of fixed size and fingerprint locations are encoded within the corresponding local coordinate system of the grid cell. The model simultaneously classifies the grid cell and estimates the bounding box within the cell. This is realized by modeling the output as vector of length $O = 5 * G$

$$\text{output} = \Big(c_x^{(1)}, c_y^{(1)}, w^{(1)}, h^{(1)}, g^{(1)}, \ldots \; c_x^{(G)}, c_y^{(G)}, w^{(G)}, h^{(G)}, g^{(G)}\Big) \tag{8}$$

where G is the number of grid cells. Let $\mathbf{t} = (t_x, t_y, t_g)$ be the target vector with (t_x, t_y) being the encoded coordinates within the t_gth grid cell. Each fifth entry $g^{(i)}$ of the output corresponds to the confidence of the model that the target is within the ith cell. The largest $g^{(i)}$ determines the chosen cell. Let $\mathbb{1}_i$ be 1 if $i = t_g$ and 0 otherwise, further let $j = \text{argmax}\{g^{(1)}, \ldots, g^{(G)}\}$ [7].

The final loss function of DLB is given as

$$\begin{aligned}\mathcal{L} = \alpha \cdot - \sum_{i=1}^{G} \mathbb{1}_i \cdot log(g_i) \;\; &\text{// classification loss} \\ + \;\; \text{sum}\Big\{\Big(\mathbf{c}^{(j)} - \mathbf{t}\Big)^2 \;\; &\text{// box center loss} \\ + \Big(\big(|\mathbf{c}^{(j)} - \mathbf{t}| - \mathbf{d}^{(j)}/\beta\big)\Big)^2\Big\} \;\; &\text{// box size loss}\end{aligned} \tag{9}$$

with $\mathbf{d}^{(j)} = (w^{(j)}, h^{(j)})^T$, $\mathbf{c}^{(j)} = (c_x^{(j)}, c_y^{(j)})^T$ and $\mathbf{t} = (t_x, t_y)^T$ [7]. The floor is implicitly given by the location of the classified grid cell within the building.

VI. Evaluation

We begin by analyzing the proposed DLBIM model performance with emphasis on the choice of parameters. Subsequently, it is compared to the previously introduced reference models.

A. Data Sets

We choose two data sets for evaluation. The first data set was self-collected at RWTH Aachen University. It is a multifloor crowdsourced data set that covers a total of 4 out of 7 floors with more than 3000 collected fingerprints. The data were collected over the period from December 2018 to August 2020 featuring multiple peak collection phases. Using a self-implemented smartphone application, collected fingerprints were manually tagged. The data distribution is visualized in Fig. 8(a)–(d), where the heatmaps visualize the amount of labeled data per 4 × 4 m grid cell. For the utilized building, a georeferenced digital building model was constructed via laser scanning and transformed into keyhole markup language (KML). The walls that are utilized by the DLBIM model are obtained from the registered walls of the building model. In total, the building model contains 97

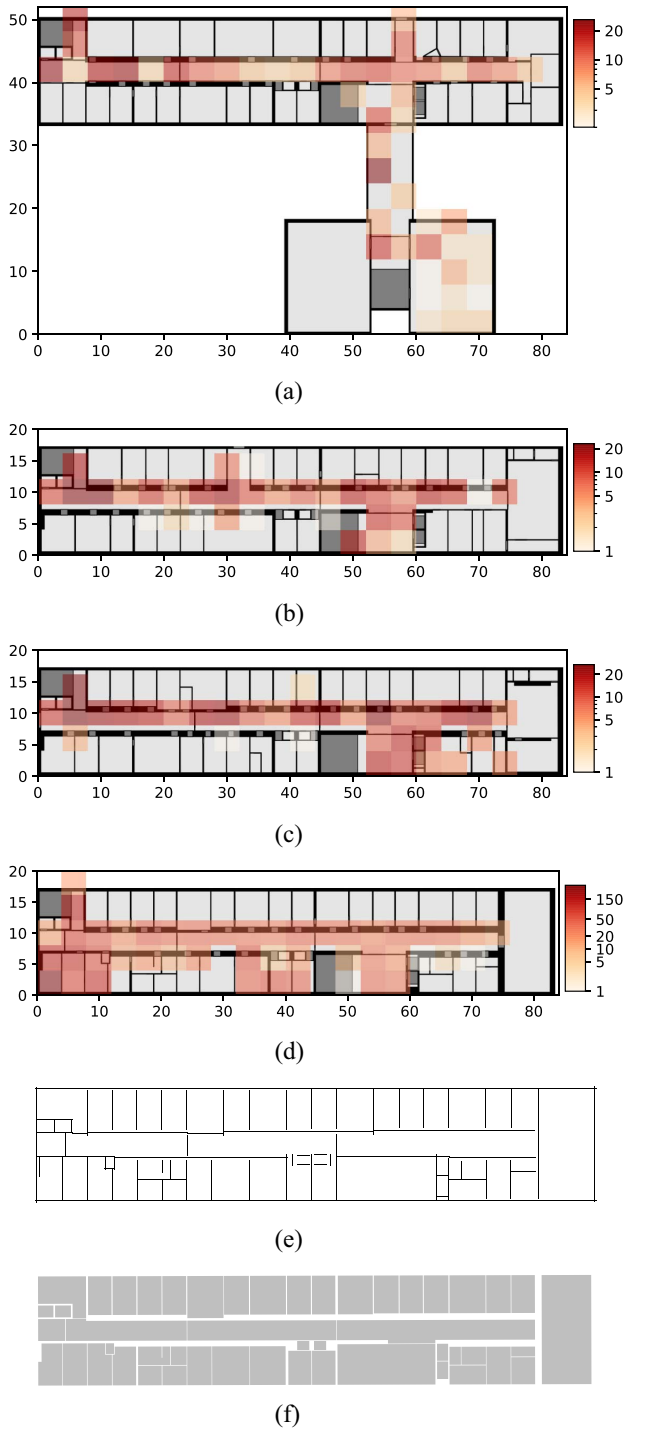


Fig. 8. Visualization of the RWTH data set. (a) RWTH floor 1. (b) RWTH floor 2. (c) RWTH floor 3. (d) RWTH floor 4. (e) RWTH floor 4 (walls). (f) RWTH floor 4 (rooms).

horizontal walls where (30, 21, 22, 24) represents the tuple of walls per floor in ascending floor order. The total amount of contained vertical walls is 190 with a distribution over floors as (48, 45, 48, 49). The predetermined areas/zones that are used to train the regular area classification models (see Section V-A) are also automatically extracted from the rooms registered of the building model. Finally, all geometries are transformed from world coordinates into a mutual local coordinate system. The walls and rooms are exemplarily visualized for the 4th floor in Fig. 8(e) and (f). For validation purposes, we split the data set into five random folds to apply cross validation.

The second data set is a publicly available WLAN fingerprinting data set, composed of 4648 fingerprints collected with 21 devices in a university building in Tampere, Finland [39]. It was published with a predetermined split into train (20%) and test (80%) data, which is adopted during our evaluation. The data distribution is visualized in Fig. 9(a)–(d), where the heatmaps illustrate the amount of fingerprints per 6 × 6 m grid cell. The data set is published with images of the floor plans. We utilize those to manually extract the most characteristic walls for the training of DLBIM. In total, we have 97 horizontal walls where (27, 21, 17, 18, 14) represents the tuple of walls per floor in ascending floor order. The total amount of vertical walls is 194 with a distribution over floors as (41, 46, 44, 43, 20). Furthermore, we manually select areas/zones based on data coverage and the floor plan structure to train the regular area classification models (see Section V-A). The walls and rooms are exemplarily visualized for the 1st floor in Fig. 9(e) and (f).

B. Model Configurations

The configurations of the best performing neural networks on the given data sets are introduced subsequently. We utilize fully connected dense layers as the shared backbone as well as in the output branches of the DLBIM model. More precisely, the model has one shared dense hidden layer with 1024 hidden units and each output branch has one hidden layer with 512 hidden units. The architecture is visualized in Fig. 10, where D represents the dimension of the fingerprint input vector. The DLB and the AC models also follow the fully connected feedforward architecture. However, the chosen configuration of hidden layers and units varies with the data set and the type of model. The configuration is presented in Table I. Every model utilizes a dropout layer after each hidden layer with dropout probability 0.5. ReLU is chosen as the activation function after hidden layers, whereas the output activation function depends on the model. All models are trained using the Adam optimization algorithm with a learning rate of 0.001 for at most 100 epochs while early stopping on a distinct validation data sets determines the exact number of epochs.

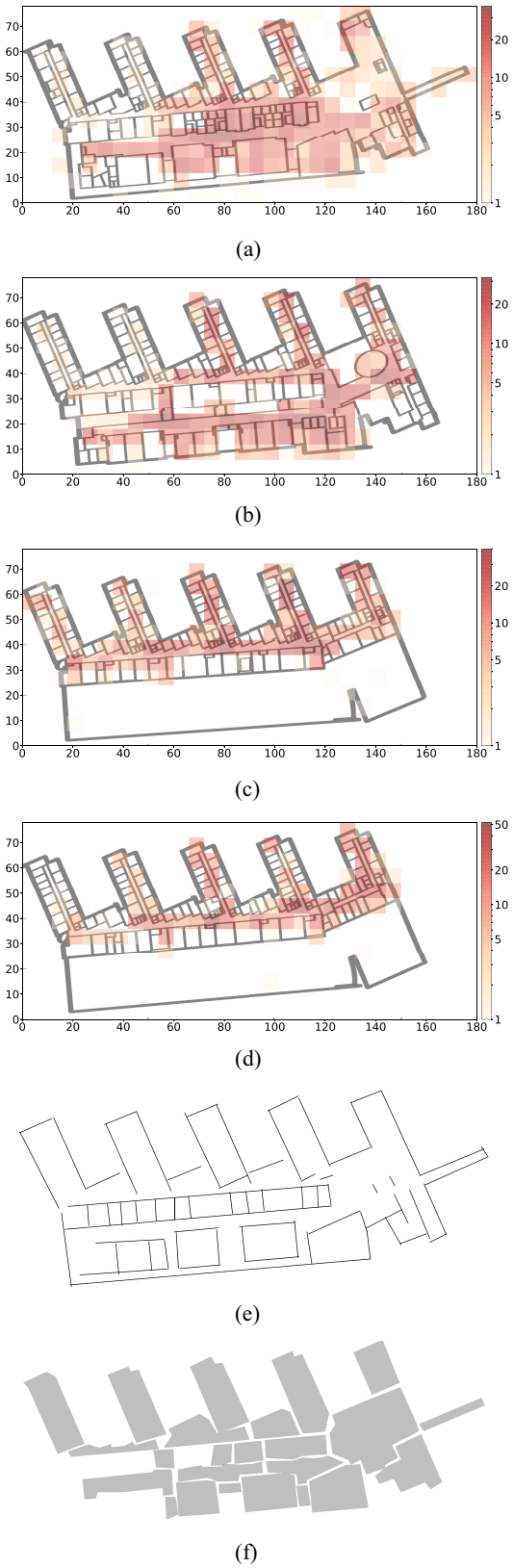


Fig. 9. Visualization of the Tampere data set. (a) Tampere floor 1. (b) Tampere floor 2. (c) Tampere floor 3. (d) Tampere floor 4. (e) Tampere floor 1 (walls). (f) Tampere floor 1 (rooms).

C. DLBIM Evaluation

1) Effect of Label Preprocessing: The ground-truth labels (target vectors) that we supply to the model mainly influence the resulting performance. It has to be carefully selected, which walls we want to supply to the model for a certain fingerprint. Having too many possible choices might decrease

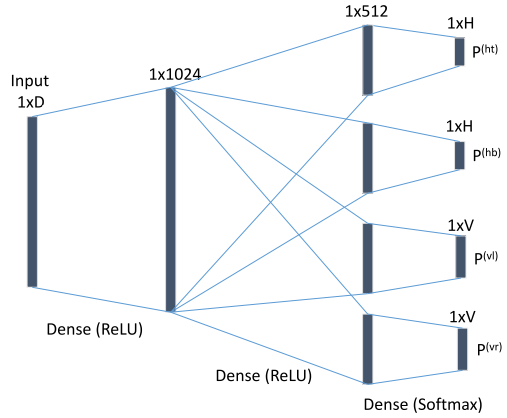


Fig. 10. Chosen neural network architecture of the DLBIM model.

TABLE I
CONFIGURATIONS OF TUNED MODELS ON DATA SETS

Dataset	Model	Hidden layers	Hidden units
RWTH	DLBIM	1 (shared) 1 (each output branch)	1024 (shared) 512 (output branches)
	DLB	2	512
	AC	2	512
Tampere	DLBIM	1 (shared) 1 (each output branch)	1024 (shared) 512 (output branches)
	DLB	2	256
	AC	1	512

the model performance, but too few walls will limit the model's options to make an optimal choice given the entire training data. In Section IV-C, we introduced two main preprocessing parameters to select the ground-truth boundary segments: max_dist (d_th) and max_range (r_th).

We want to analyze their effects on four characteristics: 1) the number of fingerprints that end up with all-zero target vectors; 2) the average number of nonzero entries within the target vectors; 3) the accuracy of the predicted polygons; and 4) the mean surface area of the predicted polygons. The results are presented in Fig. 11, where the top row [(a)–(d)] reports the effects on the RWTH data set and the bottom row [(e)–(h)] for the Tampere data set. It can be seen that choosing d_th too low (< 15) results in a large percentage of target vectors with all-zero entries. Those cannot be utilized during training of the model and thus the achieved accuracy (Area ACC) is strongly affected. While this effect can be seen for both data sets, it is particularly critical for the RWTH data set in combination with a low r_th value. A low range threshold r_th reduces the average amount of nonzero entries per target vector [Fig. 11(b) and (f)], which has a positive effect on the resulting accuracy but also limits the model expressiveness which results in a larger average surface area of the predicted polygons. On the RWTH data set, a combination of r_th $= 5$ and d_th $= 15$ results in the largest accuracy of 89% and an average surface area of 32 m^2. This parameter selection is utilized throughout the subsequent comparison with the introduced reference models. For the Tampere data set, we select a combination of r_th $= 10$ and d_th $= 25$.

TABLE II
EFFECT OF NONCOINCIDENTAL ENSEMBLE PREDICTION ON VARIOUS PERFORMANCE METRICS

Dataset	Metric	Coincidental	Non-coincidental
RWTH	Count	2910	12
	Floor ACC	0.991	0.500
	Area ACC	0.891	0.333
	Area $[m^2]$	33.95	186.59
Tampere	Count	3882	69
	Floor ACC	0.952	0.696
	Area ACC	0.707	0.362
	Area $[m^2]$	186.05	488.31

2) Effect of Regularization Constant: In Section IV-D, we introduced a regularization term to avoid collapsing polygon predictions. The importance of the regularization term is scaled by λ, which we investigate in the following. We assess its effect on the relative number of collapsing polygon predictions, while monitoring its effect on the average area hit rate. The results on the Tampere data set are depicted in Fig. 12. While the relative amount of collapsing polygon predictions is already small without any regularization, it can be significantly decreased by choosing $\lambda > 2$. Furthermore, this has no real impact on the resulting area hit rate. For our final DLBIM model, we choose $\lambda = 3$. Regarding the RWTH data set, the effect of collapsing polygon is a minor threat. Tested for multiple seeds, on average at most 1 out of 585 predicted polygons (on the test data) collapses, which prevents drawing any conclusion on the choice of the regularization constant.

3) Effect of Ensemble Learning: When using the DLBIM model in a multifloor localization setting, the predicted boundary wall segments can be implicitly used for classifying the floor. Each wall prediction of the model is made independently, which means that the implicitly predicted floors might differ. We can interpret the model as an ensemble classifier for the task of floor detection. The predicted floor is chosen based on a majority voting of the different model outputs. Intuitively, if the model's independent floor predictions do not match, we should be more curious whether the final floor prediction is correct.

We validate this claim by separately analyzing the model performance (floor hit rate, area hit rate, and average surface area) for coincidental ensemble predictions and those that have been corrected during post-processing (noncoincidental). The results are shown in Table II. Only on a small amount of predictions the ensemble does not reach a consensus. However, in that cases, this has a strong impact on the reported model metrics. The floor hit rate decreases to 50% and 70% on the RWTH and Tampere data sets, respectively. The area prediction is only correct in less than 37% on both data sets. Furthermore, the average surface area of the corrected polygon drastically increases. We can conclude that a noncoincidental ensemble prediction is a strong indicator for model uncertainty. While the total amount of cases without ensemble consensus seems marginal, its magnitude becomes

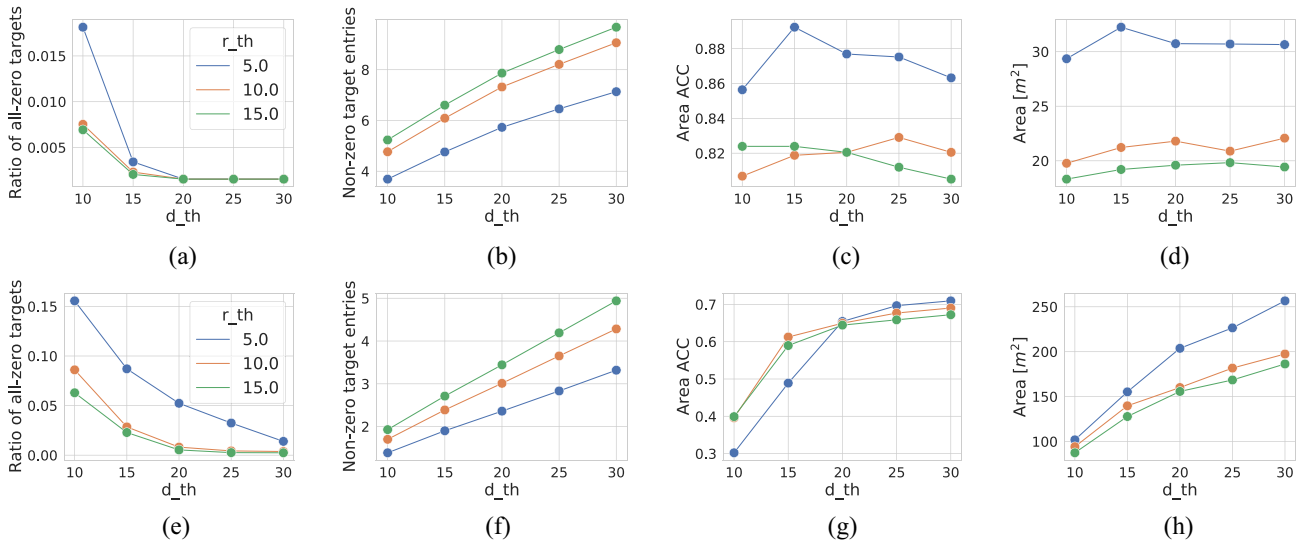


Fig. 11. Effect of target vector selection on the (a)–(d) RWTH data set and (e)–(h) Tampere data set.

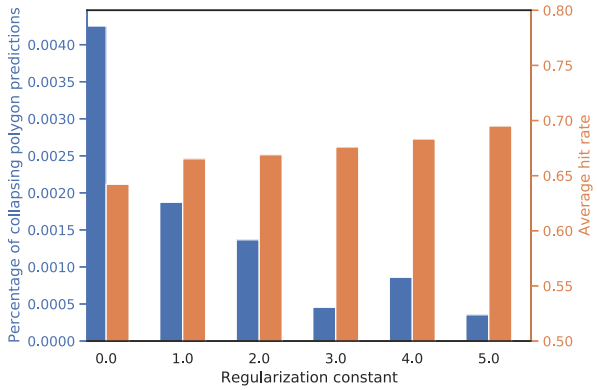


Fig. 12. Effect of the regularization constant on the percentage of collapsing polygon predictions and the average area accuracy on the Tampere data set.

TABLE III
COMPARISON OF AREA LOCALIZATION MODELS

Dataset	Model	Floor ACC	Area ACC	Deviation (of wrong areas)	Mean area [m^2]	Median area [m^2]
RWTH	DLBIM	0.989	0.889	3.929	34.573	21.837
	DLB	0.992	0.856	3.609	42.237	37.171
	AC	0.977	0.889	5.609	58.187	56.241
Tampere	DLBIM	0.947	0.701	6.793	191.329	139.034
	DLB	0.945	0.699	6.919	278.173	288.600
	AC	0.928	0.706	8.328	231.149	205.479

more relevant when comparing it to the amount of wrong floor predictions, which are only 31 on the RWTH data set and 207 on the Tampere data set.

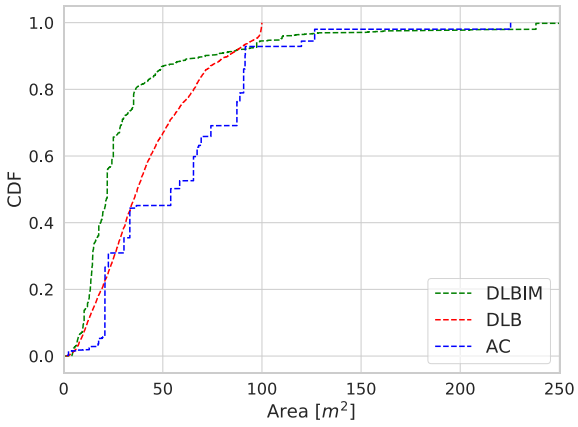
D. Comparison With Other Models

After having discussed the performance of the proposed DLBIM model with emphasis on the choice of preprocessing and parameter selection, we subsequently compare its performance to the introduced reference models of Section V. The results are presented in Table III. We asses the major accuracy indicators comprising the prediction accuracy (Floor ACC and Area ACC) and the deviation in case of false area predictions. The floor detection rate is similar for all models, while DLBIM has the highest on Tampere and DLB on the RWTH data set. However, as discussed in Section VI-C, non-coincidental ensemble predictions for the different boundary segments can be interpreted as a warning indicator for a higher chance of wrong floor predictions. This is a strong benefit of DLBIM against the other models.

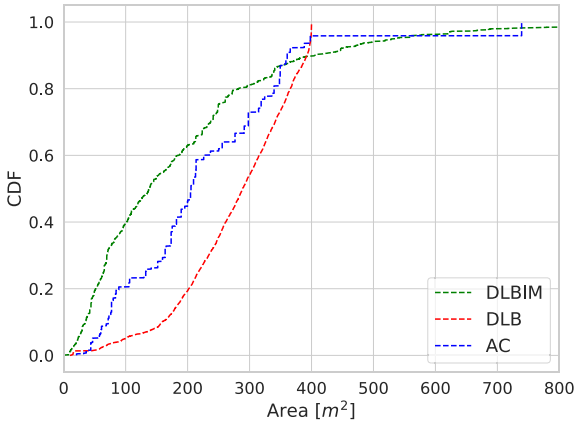
Both DLBIM and DLB deviate less in case of a wrongly estimated area as opposed to the AC models. The accuracy of area localization models can only be analyzed in context with the expressiveness of the models. When choosing only few areas, a model might easily achieve the area hit rate of close to 100%, however, the information gain of such a model would be limited.

All models achieve a similar area hit rate of above 85% on the RWTH data set and roughly 70% on the Tampere data set, which allows for comparison of the models with respect to their expressiveness. The crucial difference between the introduced DLBIM and the other models is the far narrower area estimation (geometric expressiveness) that yields the same or better accuracy than the other models. On the RWTH data set, the median surface area of DLBIM to achieve an 89% correct area estimation is 43% less than the surface area predicted by DLB, which only estimates the correct area in 86%. When comparing DLBIM to AC, the required surface area to achieve the same area estimation accuracy of DLBIM is 62% less compared to AC. On the Tampere data set, the median surface area of DLBIM to achieve a 70% correct area estimation is 51% and 32% lower than those of DLB and AC, respectively.

The distribution of surface areas predicted by the various models is visualized in Fig. 13. Note that the comparison is fully valid on the Tampere data set, since all model have the same area estimation accuracy. On the RWTH data set, the accuracy of the DLB model is slightly lower than those of



(a)



(b)

Fig. 13. Cumulative distribution function (CDF) of the surface area predicted by the various models. (a) CDF on the RWTH data set. (b) Results on the Tampere data set.

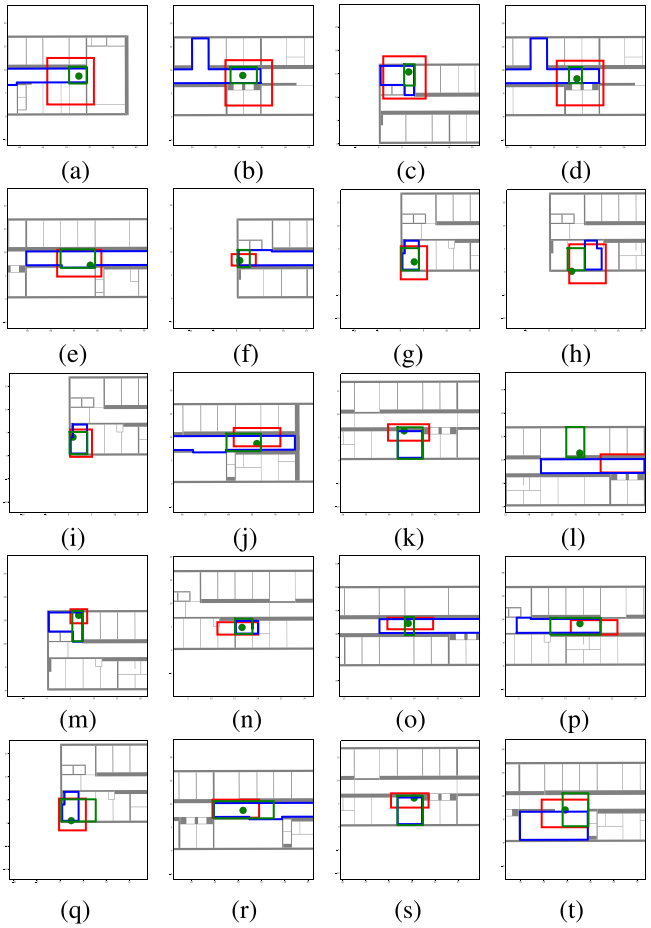


Fig. 14. Visual comparison of semantic expressiveness of the different area localization models on the RWTH data set. (a)–(t) depict selected examples, where green predictions show DLBIM, red predictions show DLB, and blue predictions show AC.

the other models, which implicitly allows the surface area to be smaller. It can be seen that the proposed DLBIM model clearly outperforms the other models with respect to geometric expressiveness. Finally, we discuss the semantic expressiveness of the three models. For the AC model, it is given due to the manually predetermined areas/zones, which are however fixed and result in a lower geometric expressiveness. The proposed DLBIM model utilizes the walls for area construction to achieve a high semantic expressiveness, which is a strong benefit against its predecessor (DLB) that has no knowledge on the floor plan structure. We illustrate this in Fig. 14 (RWTH data set) and Fig. 15 (Tampere data set), which exemplarily show the output of the three models for several test fingerprints.

It can be observed that the polygons predicted by DLBIM have a high conformity to the floor plan structure and thus a much higher semantic expressiveness as the DLB model. Furthermore, predictions are not tied to the predetermined floor plan split but constructed on an individual basis.

E. Discussion and Limitations

To the best of our knowledge, DeepLocBIM is the first approach that directly incorporates the building model within the learning phase of a neural network to provide indoor area localization. A strict split into predetermined zones fixes the expressiveness of the model as shown for the reference model during the evaluation. In contrast, DeepLocBIM achieves a high conformity to the underlying floor plan, while using its flexibility in modeling area/zone during the prediction phase to achieve a higher accuracy. This combination and its suitability for large-scale deployments emphasize the practical impact of DeepLocBIM for fingerprinting-based indoor area localization.

In the following, we discuss the potential limitations of our approach and point out possible solutions. The proposed position encoding scheme has limitations when encoding a position outside the building. Given a location $\mathbf{l}_n = (p_x, p_y)^T$. Assume that no enclosing horizontal bottom wall exists, $\{w_h = (x_s, y_s, x_e, y_e) | y_s \leq p_y\} = \varnothing$. The target vector $\mathbf{y}^{(hb)}$ would be all-zero, which results in the model learning unuseful representations. However, that is only critical for those points lying outside the convex hull that encloses all building walls. The Tampere data set contains a total of 7/697 of such critical fingerprints within the training partition and 45/3951 within the testing partition. We excluded the critical fingerprints from the training partition, however, we kept

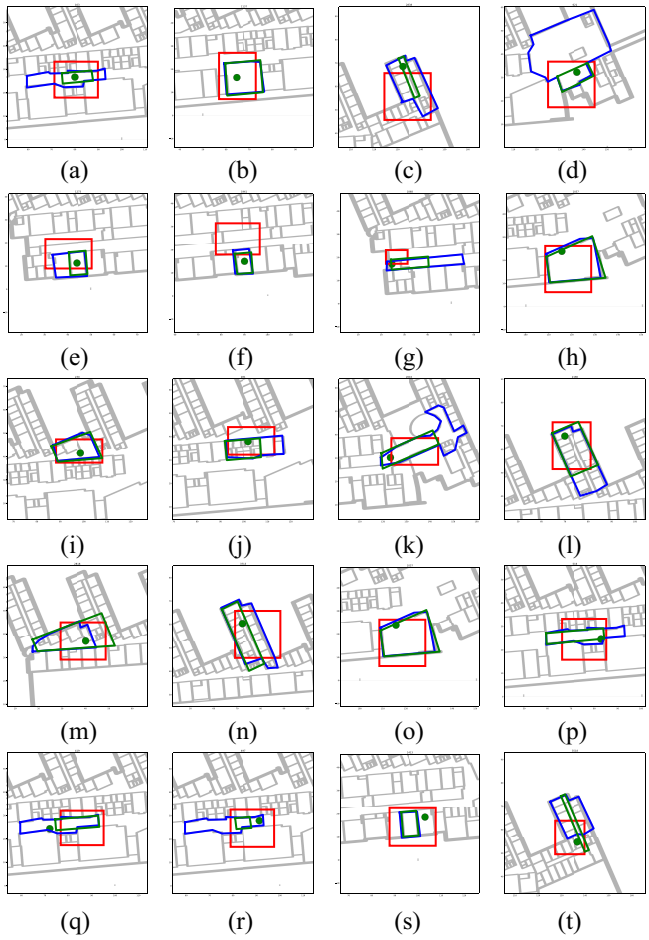


Fig. 15. Visual comparison of semantic expressiveness of the various area localization models on the Tampere data set. (a)–(t) depict selected examples.

those within the test set to not artificially manipulate the evaluation. We believe that this limitation is nevertheless minor, since we are particularly interested in localization *within* buildings. The issue of all-zero target vectors has also been studied in Section VI-C and was presented in Fig. 11(a) (RWTH) and Fig. 11(e) (Tampere). In addition to the inevitable all-zero target vectors, we showed that a wrong parameterization might increase this number and lead to excluding too many fingerprints from the training data set. Therefore, the choice of reasonable preprocessing parameters d and r is crucial.

The strict separation between horizontal and vertical walls and the applied differentiation criterion might be unfavorable for buildings where the orientation of several walls reside at the boundary of the decision criterion. Since we did not encounter such examples in our tested settings, the effect on the behavior of the model is not explicitly studied. However, the Tampere data set features several walls close to the decision criterion, which did not cause any problems.

We presented an approach for applying our model in multifloor environments using a single neural network. This has several benefits such as the obtained uncertainty when interpreting the different model outputs as an ensemble for the task of floor prediction (see Section VI-C3). However, it should be noted that the size of the predicted target vectors of the model grows with the number of walls and accordingly with the number of floors. This might impact the performance for really large settings by requiring output dimensions within the order of thousands [40]. For such scenarios, an approach using a dedicated network for each floor, which is then executed simultaneously, might be beneficial.

In Section VI-C1, we thoroughly analyzed the construction of the target vectors with respect to the parameters d and r and concluded that a spread of the probability mass over a wider range of wall candidates decreases the predicted area of the model (higher geometric expressiveness) while simultaneously lowering the accuracy of the model. We did, however, not study the effect of the dimension of the target (total number of walls) on the model outcome, which we believe would result in a similar conclusion and can be investigated in future work.

VII. Conclusion

We proposed a novel approach to fingerprinting-based indoor area localization that does not require a predetermined floor plan segmentation. Our model (DeepLocBIM) directly predicts polygon-areas that contain the position of the user. In contrast to existing segmentation-free works, it leverages the digital building model (BIM model) and thus its predictions are more valuable, since they match with the structure of the building (e.g., rooms). On two independent data sets, we compared DeepLocBIM with a segmentation-free and a segmentation-based approach. It has been shown that the predicted areas of our DeepLocBIM are more fine-grained while maintaining the same level of accuracy compared to the reference models.

We believe that DeepLocBIM provides great potential for indoor area localization. The segmentation-free approach has several benefits: 1) the model can adapt its predicted areas (shape and size) based on the input fingerprint; 2) it erases the demand for resegmentation to alter the model expressiveness; and 3) it avoids the challenge of computing a segmentation that also allows for separation in signal space (RSS). Furthermore, our model is suitable for large-scale multifloor deployments and, in contrast to existing segmentation-free works, incorporates the building structure within the learning phase.

References

[1] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2568–2599, 3rd Quart., 2019.

[2] A. Basiri *et al.*, "Indoor location based services challenges, requirements and usability of current solutions," *Comput. Sci. Rev.*, vol. 24, pp. 1–12, May 2017.

[3] F. Mazhar, M. G. Khan, and B. Sällberg, "Precise indoor positioning using UWB: A review of methods, algorithms and implementations," *Wireless Pers. Commun.*, vol. 97, no. 3, pp. 4467–4491, Dec. 2017.

[4] S. He and S.-H. G. Chan, "Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 466–490, 1st Quart., 2016.

[5] N. Anzum, S. F. Afroze, and A. Rahman, "Zone-based indoor localization using neural networks: A view from a real testbed," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kansas City, MO, USA, May 2018, pp. 1–7.

[6] M. Laska, J. Blankenbach, and R. Klamma, "Adaptive indoor area localization for perpetual crowdsourced data collection," *Sensors*, vol. 20, no. 5, p. 1443, Mar. 2020.

[7] M. Laska and J. Blankenbach, "DeepLocBox: Reliable fingerprinting-based indoor area localization," *Sensors*, vol. 21, no. 6, p. 2000, Mar. 2021.

[8] P. Roy and C. Chowdhury, "A survey of machine learning techniques for indoor localization and navigation systems," *J. Intell. Robot. Syst.*, vol. 101, no. 3, p. 63, Mar. 2021.

[9] R. Y. Chang, S.-J. Liu, and Y.-K. Cheng, "Device-free indoor localization using Wi-Fi channel state information for Internet of Things," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–7.

[10] W. Liu, H. Chen, Z. Deng, X. Zheng, X. Fu, and Q. Cheng, "LC-DNN: Local connection based deep neural network for indoor localization with CSI," *IEEE Access*, vol. 8, pp. 108720–108730, 2020.

[11] A. Mittal, S. Tiku, and S. Pasricha, "Adapting convolutional neural networks for indoor localization with smart mobile devices," in *Proc. 2018 Great Lakes Symp. VLSI*, Chicago, IL, USA, May 2018, pp. 117–122.

[12] X. Song *et al.*, "A novel convolutional neural network based indoor localization framework with WiFi fingerprinting," *IEEE Access*, vol. 7, pp. 110698–110709, 2019.

[13] L. Xiao, A. Behboodi, and R. Mathar, "A deep learning approach to fingerprinting indoor localization solutions," in *Proc. 27th Int. Telecommun. Netw. Appl. Conf. (ITNAC)*, Melbourne VIC, Australia, Nov. 2017, pp. 1–7.

[14] R. H. Jaafar and S. S. Saab, "A neural network approach for indoor fingerprinting-based localization," in *Proc. 9th IEEE Annu. Ubiquitous Comput. Electron. Mobile Commun. Conf. (UEMCON)*, Nov. 2018, pp. 537–542.

[15] K. S. Kim, S. Lee, and K. Huang, "A scalable deep neural network architecture for multi-building and multi-floor indoor localization based on Wi-Fi fingerprinting," *Big Data Anal.*, vol. 3, no. 1, p. 4, Dec. 2018.

[16] M. Ibrahim, M. Torki, and M. ElNainay, "CNN based indoor localization using RSS time-series," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, 2018, pp. 1044–1049.

[17] A. Sahar and D. Han, "An LSTM-based indoor positioning method using Wi-Fi signals," in *Proc. ACM Int. Conf. (ICVISP)*, Aug. 2018, p. 43.

[18] B. Xu, X. Zhu, and H. Zhu, "An efficient indoor localization method based on the long short-term memory recurrent neuron network," *IEEE Access*, vol. 7, pp. 123912–123921, 2019.

[19] S. Yean, B.-S. Lee, and H. L. Oh, "Feature engineering for grid-based multi-floor indoor localisation using machine learning," in *Proc. Int. Conf. Intell. Data Sci. Technol. Appl. (IDSTA)*, Valencia, Spain, Oct. 2020, pp. 142–148.

[20] A. Chriki, H. Touati, and H. Snoussi, "SVM-based indoor localization in wireless sensor networks," in *Proc. 13th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Valencia, Spain, Jun. 2017, pp. 1144–1149.

[21] H.-X. Liu, B.-A. Chen, P.-H. Tseng, K.-T. Feng, and T.-S. Wang, "Map-aware indoor area estimation with shortest path based on RSS fingerprinting," in *Proc. IEEE Veh. Technol. Conf.*, 2015, pp. 1–5.

[22] D. AlShamaa, F. Mourad-Chehade, and P. Honeine, "Zoning-based localization in indoor sensor networks using belief functions theory," in *Proc. IEEE 17th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Edinburgh, U.K., Jul. 2016, pp. 1–5.

[23] J. A. Lopez-Pastor, A. J. Ruiz-Ruiz, A. S. Martinez-Sala, and J.-L. Gomez-Tornero, "Evaluation of an indoor positioning system for added-value services in a mall," in *Proc. Int. Conf. Indoor Position. Indoor Navigat. (IPIN)*, Pisa, Italy, Sep. 2019, pp. 1–8.

[24] J. Wei, X. Zhou, F. Zhao, H. Luo, and L. Ye, "Zero-cost and map-free shop-level localization algorithm based on crowdsourcing fingerprints," in *Proc. 5th IEEE Conf. Ubiquitous Position. Indoor Navigat. Location-Based Services (UPINLBS)*, Dec. 2018, pp. 1–10.

[25] Y. Rezgui, L. Pei, X. Chen, F. Wen, and C. Han, "An efficient normalized rank based SVM for room level indoor WiFi localization with diverse devices," *Mobile Inf. Syst.*, Jul. 2017, Art. no. 6268797.

[26] H. Zhang, B. Hu, S. Xu, B. Chen, M. Li, and B. Jiang, "Feature fusion using stacked denoising auto-encoder and GBDT for Wi-Fi fingerprint-based indoor positioning," *IEEE Access*, vol. 8, pp. 114741–114751, 2020.

[27] A. S. Salazar, L. Aguilar, and G. Licea, "Estimating indoor zone-level location using Wi-Fi RSSI fingerprinting based on fuzzy inference system," in *Proc. Int. Conf. Mechatronics Electron. Automotive Eng.*, Morelos, Mexico, Nov. 2013, pp. 178–184.

[28] J. Zhang, J. Sun, H. Wang, W. Xiao, and L. Tan, "Large-scale WiFi indoor localization via extreme learning machine," in *Proc. 36th Chin. Control Conf. (CCC)*, Dalian, China, Jul. 2017, pp. 4115–4120.

[29] N. Hernández, J. M. Alonso, and M. Ocaña, "Fuzzy classifier ensembles for hierarchical WiFi-based semantic indoor localization," *Expert Syst. Appl.*, vol. 90, pp. 394–404, Dec. 2017.

[30] S. Tang, D. R. Shelden, C. M. Eastman, P. Pishdad-Bozorgi, and X. Gao, "A review of building information modeling (BIM) and the Internet of Things (IoT) devices integration: Present status and future trends," *Autom. Construct.*, vol. 101, pp. 127–139, May 2019.

[31] M. Bassier and M. Vergauwen, "Unsupervised reconstruction of building information modeling wall objects from point cloud data," *Autom. Construct.*, vol. 120, Dec. 2020, Art. no. 103338.

[32] P. Tang, D. Huber, B. Akinci, R. Lipman, and A. Lytle, "Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques," *Autom. Construct.*, vol. 19, no. 7, pp. 829–843, Nov. 2010.

[33] R. Volk, J. Stengel, and F. Schultmann, "Building information modeling (BIM) for existing buildings — literature review and future needs," *Autom. Construct.*, vol. 38, pp. 109–127, Mar. 2014.

[34] N. Li, B. Becerik-Gerber, B. Krishnamachari, and L. Soibelman, "A BIM centered indoor localization algorithm to support building fire emergency response operations," *Autom. Construct.*, vol. 42, pp. 78–89, Jun. 2014.

[35] L. Liu, B. Li, S. Zlatanova, and P. van Oosterom, "Indoor navigation supported by the Industry Foundation Classes (IFC): A survey," *Autom. Construct.*, vol. 121, Jan. 2021, Art. no. 103436.

[36] P. Herbers and M. König, "Indoor localization for augmented reality devices using BIM, point clouds, and template matching," *Appl. Sci.*, vol. 9, no. 20, p. 4260, Oct. 2019.

[37] I. Ha, H. Kim, S. Park, and H. Kim, "Image retrieval using BIM and features from pretrained VGG network for indoor localization," *Build. Environ.*, vol. 140, pp. 23–31, Aug. 2018.

[38] C.-H. Chang, C.-Y. Lin, R.-G. Wang, and C.-C. Chou, "Applying deep learning and building information modeling to indoor positioning based on sound," in *Proc. Comput. Civil Eng.*, Atlanta, GA, USA, Jun. 2019, pp. 193–199.

[39] E. Lohan, J. Torres-Sospedra, H. Leppʹakoski, P. Richter, Z. Peng, and J. Huerta, "Wi-Fi crowdsourced fingerprinting dataset for indoor positioning," *Data*, vol. 2, no. 4, p. 32, Oct. 2017.

[40] S. Vijayanarasimhan, J. Shlens, R. Monga, and J. Yagnik, "Deep networks with large output spaces," Apr. 2015, *arXiv:1412.7479*.

Marius Laska received the B.S. and M.S. degrees in computer science from RWTH Aachen University, Aachen, Germany, in 2016 and 2019, respectively, where he is currently pursuing the Doctoral degree in construction and geoinformatics.

From 2015 to 2019, he was a student assistant, where he has been a Research Assistant with the Geodetic Institute and Chair for Computing in Civil Engineering, RWTH Aachen University since 2019. His research interest includes the application of machine learning for indoor localization, in particular fingerprinting-based area localization with high reliability as well as scalable architectures for real-time stream processing in IoT settings.

Jörg Blankenbach (Member, IEEE) received the Dipl.-Ing. and Dr.-Ing. degrees in geodesy from Technische Universität Darmstadt, Darmstadt, Germany, in 2001 and 2006, respectively.

From 2006 to 2012, he headed a Research Group of Indoor Positioning with the Geodetic Institute, Technische Universität Darmstadt. Since 2012, he has been a Professor and the Director of the Geodetic Institute and Chair for Computing in Civil Engineering and Geo Information Systems, RWTH Aachen University, Aachen, Germany. His research interests are indoor positioning, geosensor networks, and distributed geo-information systems, as well as 3-D surveying and modeling of the built environment in the context of building information modeling.