Tagging the public buildings' main entrance based on OpenStreetMap and binary imbalanced learning

Preprint · June 2020

DOI: 10.13140/RG.2.2.35284.63366

7 authors, including:

Xuke Hu
German Aerospace Center (DLR)
23 PUBLICATIONS **227** CITATIONS
SEE PROFILE

Alexey Noskov
Philipps University of Marburg
30 PUBLICATIONS **69** CITATIONS
SEE PROFILE

Hongchao Fan
Norwegian University of Science and Technology
81 PUBLICATIONS **1,118** CITATIONS
SEE PROFILE

Tessio Novack
Universität Heidelberg
33 PUBLICATIONS **217** CITATIONS
SEE PROFILE

ARTICLE TEMPLATE

Tagging the public buildings' main entrance based on OpenStreetMap and binary imbalanced learning

ARTICLE HISTORY
Compiled June 7, 2020

ABSTRACT
The entrance of buildings is an important feature that connects their internal and external environments. Most frequently, automatic approaches for detecting building entrances are based on street-level images, which, however, are not widely available. To address this issue, we propose a more general approach for inferring the location of the main entrance of public buildings based on the association between spatial elements extracted from OpenStreetMap. In particular, we adopt three binary classification approaches: Weighted Random Forest, Balanced Random Forest, and SmoteBoost to model the association relationship. The features considered in the classification are of two types: (1) intrinsic features derived from the footprint, such as the distance to the centroid of the footprint, and (2) extrinsic features derived from spatial contexts, such as the shortest path distance to the main roads. Extensive experiments have been conducted on 320 public buildings with an average perimeter of 350 meters. The experimental results showed that a mean linear distance error of 21 meters and a mean path distance error of 22 meters were achieved by using the Weighted Random Forest and Balanced Random Forest models, ruling out 90% of the incorrect locations of the main entrance at buildings. Our work finds relevance, for example, in saving pedestrians' way-finding efforts.

KEYWORDS
Main entrance tagging;OpenStreetMap; Imbalanced learning; Random forest

1. Introduction

The entrance of public buildings plays a vital role in connecting outdoor and indoor spaces. Determining the location of the main entrance is essential in many location-based service (LBS) applications, such as way-finding since it is normally the end destination of outdoor way-finding (Zeng and Weber 2015). However, the entrance information is missing on current mainstream map providers, such as Bing Maps and Google Maps. This can lead to several issues (e.g., inaccurate navigation and misleading) when using these map services. For example, when following the planned route by map providers to a certain building, users are often guided to the wrong location, which is far away from the main entrance. Consequently, they need to spend even more efforts to find the main entrance by themselves. Times way-finding efforts can be saved and shorter and simpler routes can be derived if the main entrance of buildings is a mapped feature. This is an unpleasant experience especially for the people with mobility constraints because public buildings are normally complex and of large proportions. Figure 1 shows two real examples when using Google Maps to plan a route to a certain building. Realizing the importance of mapping the building entrance, the OpenStreetMap (OSM) contributors have created a tag to represent the main entrance

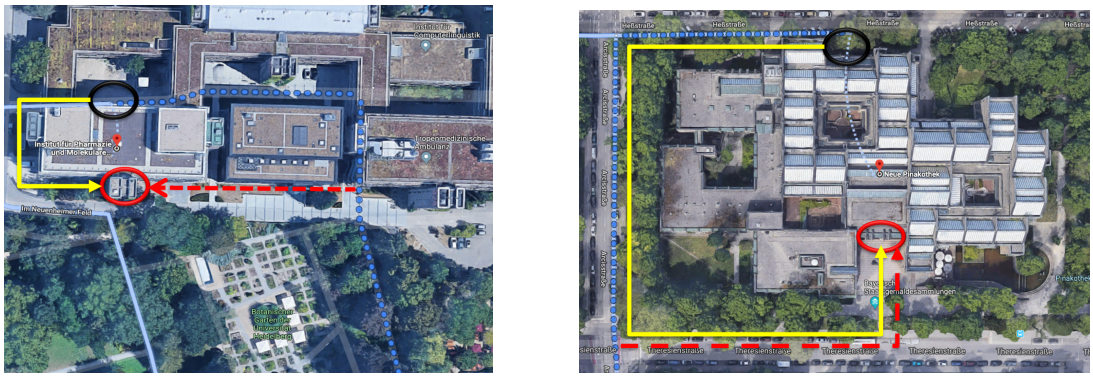


Figure 1. Inaccurate and misleading navigation by Google Maps due to missing of entrance information. Location tagged by black and red circle are planned target point by Google Maps and true main entrance, respectively. The blue dotted line represents the planned path by Google Map. The yellow line shows the extra path taken to find the true entrance to the planned target location. The red dashed line denotes the shortcut that is not found by Google Maps.

as a node with the OSM key 'entrance' and value 'main' (Goetz and Zipf 2011).

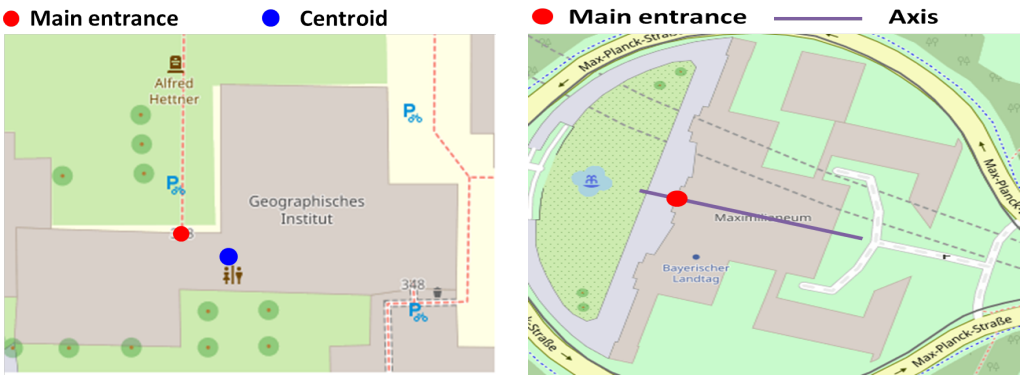
However, to the present date, only a small proportion of buildings on OSM have an entrance tag feature. For instance, in the London area, there are only about 60 buildings that are tagged with the main entrance. This is because it is difficult for volunteers to contribute with the entrance of the building in comparison to other features, such as the buildings' footprints and the venues' names, which can be obtained from personal experience, public information, and Bing satellite imagery. Only the volunteers who are familiar with the building would mark the main entrance on OSM. To overcome this challenge, some automatic solutions have been proposed to identify the entrance of buildings from street-level images (Liu et al. 2014; Kang et al. 2010; Liu et al. 2017) and remarkable tagging results have been achieved. However, the data they leverage limits the applicability of the approach, as the street-level images that cover a wide range of areas are not guaranteed to be available even from Google Street View, which is the largest provider of street view images to date, specially outside developed countries. Furthermore, the entrance of many buildings can not be directly observed from streets due to the existence of obstacles or because the entrance does not face any street.

To mitigate this gap, a more general and applicable main entrance tagging approach for public buildings (e.g., hospital, office building, and museum) is proposed by leveraging OSM, which provides high-quality geospatial information in many regions, such as the Europe and the United States (Hochmair et al. 2013) and is freely accessible. The definition of public buildings might vary as counties. In this study, we follow the specification of the OSM community [1], in which the public building is defined as the building constructed as accessible to the general public with the OSM tag as 'building=public', such as the town hall, police, courthouse, hospital, library, and museum. The reason that we focus only on public buildings rather than private buildings such as residential house, is that their shape is complex and large-scaled and they are the most frequent route destinations. Therefore, to guide users to find the entrance of public buildings is of larger public interest. Besides, this work focuses on the detection of the main entrance, ignoring the possible secondary or ancillary entrances since in many

[1] https://wiki.openstreetmap.org/wiki/Tag:building%3Dpublic

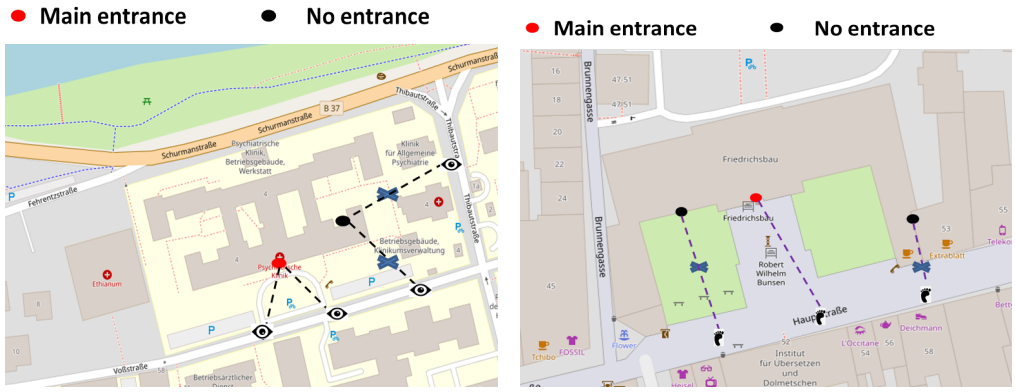
cases the public is not allowed to use secondary entrances, commonly used mostly for special purposes, such as emergency evacuations. Therefore, from the perspective of navigation, the main entrance is more important than the secondary entrance.

The idea of this study is inspired by the fact that there exists strong association between the spatial (particular building) elements in the real world as the buildings are man-made structures that are constructed with plans made by people. That is, given partial spatial elements, the other element can be inferred based on the association relationship (Hu, Fan, and Noskov 2018; Hu et al. 2019, 2020). Intuitively, the entrance is associated with two kinds of spatial elements: (1) The location of the main entrance of a public building is correlated with the shape of its footprint. For instance, the main entrance is located normally near the centroid of the footprint, as shown in Figure 2a. If the footprint is reflection symmetry, the main entrance is very likely located close to the symmetry axis to maintain the symmetric characteristics of the building, as shown in Figure 2b. Another example is that the main entrance sometimes is located at the convex and concave edge of the footprint, which corresponds to the rain-shed, independent vertical passage or entrance foyer. (2) The main entrance of a building is correlated with its surrounding spatial contexts, such as the streets. Generally, the main entrance should be easily accessed and observed from the streets, which often has shorter path distance to the streets and more observable points from the street than the other locations at the footprint, as shown in Figure 3.



(a) Entrance is close to centroid of footprint (b) Entrance is close to axis of symmetric footprint

Figure 2. Location of entrance is correlated with shape of footprint



(a) Entrance is easily observed from roads (b) Entrance is easily accessed from roads

Figure 3. Location of entrance is correlated with spatial contexts of buildings

The main entrance tagging issue can be regarded as the association relationship modeling issue, which is implemented through statistical learning. Specifically, by splitting the footprint into discrete equidistant points (also named samples), the task is thus converted to identify which one is the most likely location of the main entrance (positive). For each point, the corresponding intrinsic and extrinsic features are extracted by measuring the relationship between the samples and the footprint as well as its spatial context, respectively. The proposed approach consists of two stages, namely, model training and entrance tagging. During the training stage, three different classification models are fitted to model the association relationship. They are the Weighted Random Forest (WRF)(Effendy, Baizal et al. 2014), Balanced Random Forest (BRF) (Khalilia, Chakraborty, and Popescu 2011), and the SmoteBoost (Chawla et al. 2002) algorithms. The reason for testing and comparing these models is that they are robust to class imbalance issues. During the tagging stage, the fitted model is used to calculate the probability of assigning each sample in a test building as positive, and the one with the highest probability is chosen as the estimated location of the main entrance.

The main contributions of this work are twofold:

(1). To the best of our knowledge, this work is the first to propose an automatic approach to estimating the physical location of the main entrance of buildings based only OSM data.

(2). Our proposed approach is broadly applicable, as it relies only on OSM data, which is freely accessible and covers a wide range of areas in the world.

The remainder of this paper is structured as follows: In Section 2, we introduce the relevant works. In Section 3, we present the workflow of the proposed approach and give the details of each step. We evaluate the proposed approach through experiments on 320 public buildings in Section 4, and discuss relevant issues in Section 5. Conclusions are drawn in Section 6.

2. Related works

We categorize the entrance detection approaches into two groups: door detection (indoor) and entrance detection (outdoor) since to some degree an entrance is also a door but the detection approach is different.

Door detection: Door detection approaches are widely investigated due to two reasons. First, robots need to recognize the location of doors for autonomous naviga-

tion. Second, indoor reconstruction solutions also need to detect the location of doors to build a complete indoor navigation network for pedestrians. For instance, Murillo et al. (2008) presented a technique for detecting doors using only visual information for robot navigation. The probability distribution is learned in a parametric form from a few reference images in a supervised setting. A model-based approach is used, where the door model is described by a small set of parameters characterizing the shape and the appearance of the object. The geometry of the door is specified by a small number of parameters and the appearance is learned from the reference data. The constraints of man-made environments were used to generate multiple hypotheses of the model and the learned probability distribution was used to evaluate their likelihood. Zhao et al. (2015) proposed a light-weight and broadly applicable door detection approach based on the magnetometer embedded on a smartphone. It analyzes readings from the built-in magnetic sensors since the anomalies or sharp fluctuations of magnetic signals normally happened at doors. Nikoohemat et al. (2017) proposed using mobile laser scanners for data collection. It can detect openings (e.g., windows and doors) in cluttered indoor environments by using occlusion reasoning and the trajectories from the mobile laser scanners. The results showed that using structured learning methods for semantic classification is promising. Recently, Quintana et al. (2018) presented an approach that detects open, semi-open and closed doors in 3D laser scanned data of indoor environments. It integrates the information regarding of both the geometry and colour provided by a calibrated set of 3D laser scanner and a colour camera. The integration of geometry and colour makes it robust to occlusion and variations in colours resulting from varying lighting conditions at each scanning location and different scanning locations.

Entrance detection: Apart from door detection, a couple of automatic methods have also been proposed to detect the entrance of buildings, which is the focus of this work. The traditional ways to detect the entrance is through images analysis. That is, the detection of entrance is treated as the issue of semantic tagging from images. For instance, Liu et al. (2014) proposed a three-stage system that starts with a high-recall entrance candidate extractor, which is followed by classifying candidates based on local image features. The final stage fuses results from multiple views by using Markov chain Monte Carlo to solve a Bayesian inference problem, and to select the best set of entrances that explain the image of a facade. The system achieves a recall of 70% on a challenging data set of urban scene images. Kang et al. (2010) proposed an approach to detecting the entrance of building for robot navigation based on the images that can be collected in real-time by mobile robots during navigation. They adopted a probabilistic model for entrance detection by defining the likelihood of various features for entrance hypotheses. The basic idea is to exclude non-entrance regions in the surface of a building, such as walls and windows, which are extracted from the image of the surface. The reminding region is considered as the candidate of entrance, which is then evaluated by their proposed probabilistic model. Recently, Talebi, Vafaei, and Monadjemi (2018) presented a vision-based method for detecting building entrances with outdoor images. They first converted the RGB image into gray-scale image, from which the vertical and horizontal line segments can be detected by using Line Segment Detector (LSD) algorithm. Then, the regions between the vertical lines were specified and the features including height, width, location, color, texture and the number of lines inside the regions are obtained. Finally, they used some additional knowledge such as door existence at the bottom of the image and a reasonable height and width of a door to decide if a door is detected or not. Different from the aforementioned works that use manually defined features to detect entrance, Liu et al. (2017) proposed

using random forest classifier to perform automatic feature selection and entrance classification. The process of the algorithm is as follows: first, the scene geometry was exploited and the multi-dimensional problem is reduced down to a one-dimensional (1D) problem. Then, a rich set of discriminative image features for entrances was explored according to constructed designs, specifically focusing on properties such as symmetry and color consistency. Lastly, a joint model was formulated in three dimensions (3D) for entrances on a given facade, which enables the exploitation of physical constraints between different entrances on the same facade in a systematic manner to prune false positives, and thereby selected an optimum set of entrances on a given facade. The drawback of these works is that they rely on the stree-level image, which can be obtained from some map providers, such as Google Street View and through the cameras equipped on the robot during navigation. The street-level image does not always contain the entrance of all the buildings since the entrance might not face any street. Meanwhile, Google Street View covers only partial large cities in the world. The robot-based solution is not applicable for pedestrian way-finding, which needs to know the location of the main entrance in advance.

3. Approach

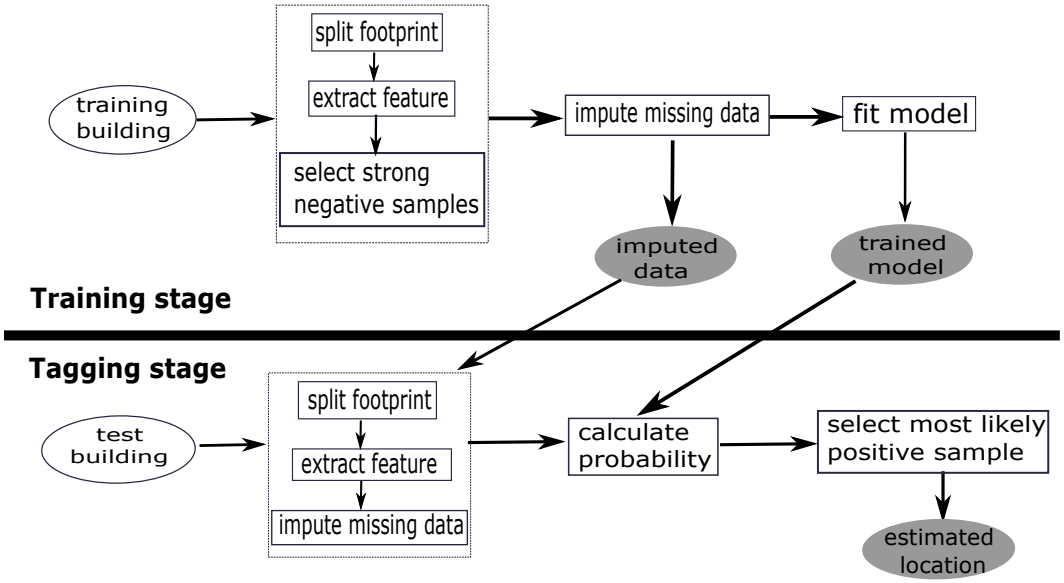


Figure 4. Workflow of proposed approach

As illustrated in Figure 4, the proposed approach consists of two stages: training and tagging. In the training stage, the edges of each building are first split into single points, also named samples. They are then tagged as positive (true main entrance) and negative accordingly. The next step is to extract features for each sample by measuring the relationship between the sample and the footprint (intrinsic features) and the surrounding spatial entities (extrinsic features), such as the distance to the centroid of the footprint and the shortest path distance to the main roads. However, some negative samples are neighbors of the positive sample, which may cause the mis-classification of the positive sample. To solve this issue, only the 'strong' negative samples are used in the training data. The 'strong' negative samples are those whose

physical or feature distance is far away from the positive sample. After collecting the samples from all of the training buildings, the missing data of the training samples would be filled out, which is caused by the lacking of some spatial entities around a certain building. For instance, not all the buildings have main roads around. Finally, a classification model that can deal with the unbalanced class issue is fitted based on the training samples.

In the tagging stage, the footprint of a test building is split into single points and the corresponding features are extracted in the same way as in the training stage. The next step is to impute the missing data by using the strawman strategy (Tang and Ishwaran 2017). Specifically, the missing value of a numerical feature is filled out with the median value of the non-missing values of this feature in the training samples. Likewise, the missing value of a categorical feature is filled out with the most frequent value of the non-missing values of this feature in the training samples. Then, the trained model is used to calculate the probability of assigning each sample to positive or negative. Finally, the one with the highest positive probability among all the samples in a building is chosen as the estimated location of the main entrance. In the following sections, we will elaborate on the key steps of the training stage.

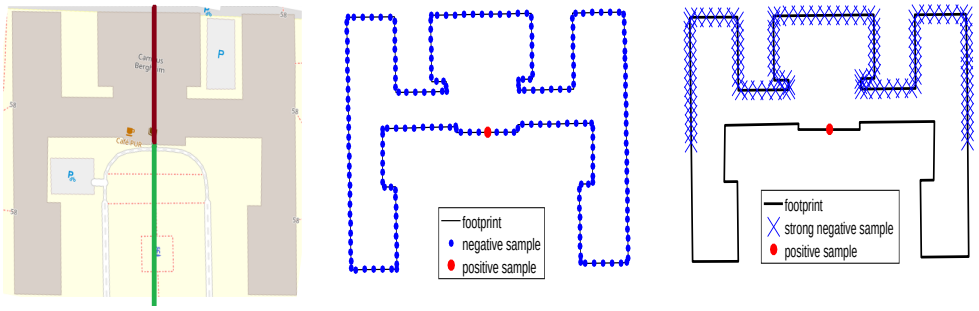
3.1. *Data pre-processing*

The input of the training stage is buildings. For each one, its external edges are first split into smaller segments with an interval of three meters as the width of the main entrance of public buildings is normally around three meters. By doing so, the segment can approximately represent the main entrance. The segment whose length is below three meters is treated as a complete segment. Then, the midpoint of the segment is chosen as a sample (the candidate location of the main entrance). The one whose parental segment contains the true main entrance is tagged as positive, and the others are tagged as negative. We define the edge that contains a sample as the master edge of the sample. The features of each sample is then extracted by measuring the relationship between the sample and the footprint (intrinsic features) and the surrounding spatial entities (extrinsic features), which will be elaborated in the following section. Figure 5a shows the footprint of a building. Figure 5b shows the discretized result, from which we can see: (1) the number of the negative sample is much larger than that of the positive sample (only one); (2) the positive sample is physically surrounded by some negative samples. If we fit a normal classification model with these samples, all the test samples would most likely to be categorized as negative to achieve the highest classification accuracy. However, what we expect is to correctly pick out the positive samples from the negative ones.

To handle the inbalanced data issue, this work adopts three classification models namely, SmoteBoost, Balanced Random Forest, and Weighted Random Forest. To address the second issue, the negative samples that are close to the positive sample in either physical or feature distance are ruled out from the training samples to reduce the interference of the negative samples. That is, only the 'strong' negative samples are preserved. The physical and feature distance thresholds are denoted by P_T and F_T, respectively. The physical distance between two samples is defined as the shortest linear distance along the footprint, as shown in Figure 10. The feature distance is defined as the Euclidean distance of the feature vector of two samples. Note that before calculating the feature distance, each variable in the vector is first normalized by using the Min-Max Normalization method, as shown in Formula 1, limiting the

value of all features to the range of zero and one. Figure 5c shows the selected 'strong' negative samples.

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{1}$$



(a) footprint of a building on OSM (b) discretization of footprint (c) resampling of nagative samples

Figure 5. Process of footprint split and sample extraction

After obtaining positive and 'strong' negative samples for all the training buildings, the straw-man imputation strategy is adopted to deal with the missing data issue in the training samples. Specifically, we fill out the missing value of a numerical feature with the median value of the non-missing values of this feature in the training samples. Likewise, we fill out the missing value of a categorical feature with the most frequent value of the non-missing values of this feature in the training samples. More approaches that impute the missing data will be investigated in our future work, such as KNN, missing forest, and multiple imputation by chained equations (MICE) (Tang and Ishwaran 2017; Deng et al. 2016).

3.2. *Feature extraction*

This section introduces the procedure of extracting features for each sample in a building. Given a building, its footprint and surrounding spatial contexts or entities are obtained from OSM, on which the intrinsic and extrinsic features can be derived, respectively. In total, we define 84 features. The complete features we use can be found in the shared files.

3.2.1. Extrinsic feature

The spatial contexts include *address street*, *main road*, *pedestrian way*, *service way*, *railway*, *bicycle parking area*, *landmark*, and *postbox*. Partial buildings have been tagged with the address street. The key is 'addr_street' in OSM and the corresponding value is renamed as 'addr_street_value' in this work, based on which the address street of the building are retrieved. The key and value of these contexts in OSM are given in Table 1. The relationship between the sample and the spatial contexts can be measured in multiple ways, as shown in the first column of Table 2. Note that, we do not choose the pathways connected to the building as the spatial context since it is too strong features that indicate the location of the entrance, as shown in Figure 6.

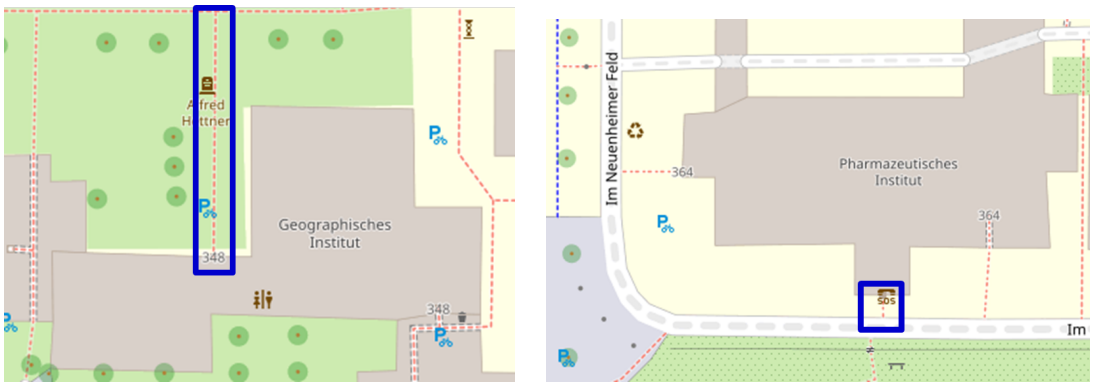


Figure 6. Pathway surrounded by blue rectangles is connected to buildings on OSM with the connection point as the location of main entrance

Before introducing the specific features, we first define the outer perpendicular line (OPL) and inner perpendicular line (IPL) of a sample, which are needed in defining some features. OPL of a sample is the line with the sample as the start point, extending along the line that is perpendicular to the master edge of the sample and deviating from the building. Conversely, IPL is the line with the sample as the start point, extending along the line that is perpendicular to the master edge of the sample and toward the footprint. The OPL and IPL of a sample in Figure 5a are denoted by the green and brown lines, respectively. The following measures are used to define the external feature:

Shortest path distance: It refers to the shortest path distance from a sample to multiple spatial contexts of the same type, such as multiple service ways. Normally, the true sample (main entrance) can be easily accessed (with a shorter path distance than other samples) from address streets and main roads. To calculate the path distance,

Table 1. OSM key and value of spatial entities used to extract external features

	key	value
address street	name	addr_street_value
main road	highway	primary / secondary/ tertiary / unclassified/ residential
pedestrian way	highway	pedestrian
service way	highway	service
railway	railway	rail
bicycle parking area	amenity	bicycle_parking
landmark	artwork_type	sculpture
	tourism	artwork
	historic	memorial
	amenity	fountain
	man_made	water_well
	man_made	flagpole
postbox	amenity	post_box

the path obstacle is first extracted, including building, barrier, grass, water, railway, and garden. Next, the spatial contexts in the form of line segments or polygons are split into points at a certain interval. A small interval will cause a high computational cost of the shortest path while a large value might cause the inaccurate computation of the shortest path distance. Thus, in this study, a 5 meter interval is adopted. The path distance from the sample to these context points is then calculated with the A-star algorithm (Hart, Nilsson, and Raphael 1968), and among them the shortest path distance is obtained.

Turning degree: It refers to the turning degree of the shortest path from a sample to a certain spatial context. It is calculated by dividing the shortest path distance by the euclidean distance from the sample to the target location on the shortest path. The larger the value, the more turnings on the shortest path.

Accessible: It measures if a sample is accessible from a certain spatial context. It can be obtained from the result of the shortest path distance.

Degree of visibility: It measures how easily a sample (candidate entrance) can be observed from certain spatial contexts. Generally, the main entrance is easily observed from main roads. The obstacles that hinder visibility include buildings and barriers defined by the key of 'barrier' on OSM. To calculate the degree of visibility of a sample, the spatial contexts (e.g., main roads) are first discretized into points at a certain interval and the number of the points from which a sample can be directly observed without obstruction is used as the degree of visibility. Likewise, a 5 meter interval is adopted to achieve a balance between the computational cost and accuracy.

Visible: It measures if a sample is visible from a certain type of spatial contexts. It can be derived from the result of the degree of visibility.

Euclidean distance: It measures the Euclidean distance between a sample and its spatial contexts.

Table 2. Extrinsic feature extraction by measuring the relationship between samples and spatial contexts

	address street	main road	pedestrian way	service way	railway	bicycle parking area	landmark	postbox
Shortest path distance (*)	✓	✓	✓	✓	✓	✓		
Accessible	✓	✓	✓	✓	✓	✓		
Turning degree	✓	✓	✓	✓	✓	✓		
Degree of visibility (*)	✓	✓		✓	✓			
Visible	✓	✓		✓	✓	✓	✓	✓
Euclidean distance (*)							✓	✓

The other important extrinsic features are as follows:

Open area (*): It measures the size of an open area before a sample. To calcualte the feature, the OPL of the sample is first obtained, which is followed by searching all the intersection points between the OPL and obstacles. The open area then equals

the shortest Euclidean distance between the intersection points and the sample. The obstacle here refers to the building, grass, main road, barrier, water, and railway.

Distance to buildings(*): It measures the Euclidean distance from a sample to the nearest building. It is calculated in the same way as **Open area**. The only difference is that the obstacle here refers to buildings.

3.2.2. Intrinsic feature

Intrinsic features refer to the features extracted from the building itself, i.e., footprint. Some important intrinsic features are as follows:

Distance to centroid (*): It represents the Euclidean distance from a sample to the centroid of the footprint.

Proportion (*): It measures how close a sample is to the midpoint of its master edge. It is calculated by dividing the distance between the sample and the midpoint of its master edge by the length of the master edge. The value ranges from 0 to 0.5.

Existence of axis: It indicates if the reflection symmetry axis exists since not every building is symmetric. For instance, the footprint in Figure 5b is reflection-symmetric and the axis is the perpendicular bisector of the master edge of the positive sample.

Distance to reflection symmetry axis (*): It represents the perpendicular distance from a sample to the reflection symmetry axis of the footprint if the axis exists.

At intersected edge of axis (*): It indicates if a sample is located at the edge that intersects the axis of the building if the axis exists.

Length of master edge (*): It represents the length of the edge that contains the sample.

Face inner(*): It indicates if the OPL of a sample intersects the other edges of the building (except the master edge). Normally, the OPL of the entrance sample does not intersect the edges of the footprint, such as the positive sample in Figure 5b.

Concavity and convexity: It indicates if the master edge of a sample is concave (0), convex(1), or neither (-1). An edge is defined as convex only when the inner angles of the two endpoints of this edge approximate 90 degrees, while the neighboring two angles approximate 270 degrees. In contrast, an edge is defined as concave only when the two angles of this edge approximate 270 degrees, while the neighboring two angles approximate 90 degrees. For instance, the master edge of the positive sample in Figure 5b is concave.

Opposite shape: It indicates if the opposite edge of the master edge of the sample is concave (0), convex(1), or neither (-1). The opposite edge of an edge is defined as the closest exterior edge of a building, which intersects the perpendicular bisector of the edge.

Note that, for both intrinsic and extrinsic features, the one with the star symbol (*) means that apart from the absolute measurements, the sorting result of measurement of a sample among the total samples in the same building is also treated as features. It measures if one sample is closer to some spatial contexts or easier to be observed from some places than the other samples in the same building. Intuitively, the positive sample (entrance) is closer to the centroid of a building than most of the negative samples. The sorting result of each sample in a building, denoted by $S = \{s_1, s_2, ...s_n\}$ is normalized, denoted by $NS = \{s_i/n\}_{i\in[1,n]}$. s_i denotes the sorting result of i-th sample, ranging from 1 to n, while n denotes the number of samples in a building. In this way, the value of the feature is limited in the range of 0 and 1, making it globally comparable.

3.3. *Classification models for imbalanced data*

As we mentioned before, the positive sample is far less than the negative ones, which causes imbalanced data issues (Sun, Wong, and Kamel 2009). The common ways to deal with the issue include over-sampling the minority class, under-sampling the majority class, and giving more weight to the minority class. This work adopts three different classification models: SmoteBoost, Balanced Random Forest, and Weighted Random Forest, which are the representative methods of the three strategies.

SmoteBoost: It was first proposed by Chawla et al. (2003) for countering imbalance in a dataset, which combines the Synthetic Minority Oversampling Technique (SMOTE) (Chawla et al. 2002) and Adaptive Boost (AdaBoost) (Schapire 2013). Specifically, before each boosting step, a SMOTE resampling calculates new synthetic examples for the minority class. The minority class is over-sampled by taking each minority class sample and introducing synthetic examples from the k minority class nearest neighbors. AdaBoost works to improve the performance of weak learners (poor predictive models, but better than random guessing). It iteratively builds an ensemble of weak learners by assigning a higher weight to samples that the current weak learner misclassified during each iteration. This weight determines the probability that the sample will appear in the training of the next weak learner. For this reason, boosting algorithms like AdaBoost are particularly useful for class imbalance problems because higher weight is given to the minority class at each successive iteration as data from this class is often misclassified. More details of the AdaBoost can be found in (Chawla et al. 2003).

Balanced Random Forest: Balanced Random Forest (BRF) is a variant of the random forest by under-sampling the majority class in building each decision tree. BRF algorithm consists of three steps. (1) For each iteration (building a tree) in random forest, draw a bootstrap sample from the minority class and randomly draw the same number of samples, with replacement, from the majority class. (2) Induce a classification tree from the data to maximum size, without pruning. (3) Repeat the two steps above for the number of trees desired. During the tagging stage, the predictions of all the trees in the forest are aggregated to make the final prediction. More details of the Balanced Random Forest can be found in (Khalilia, Chakraborty, and Popescu 2011).

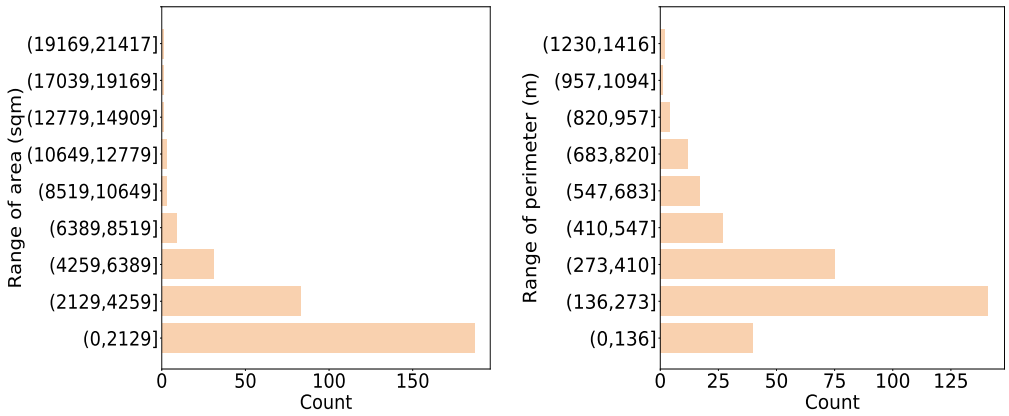
Weighted Random Forest: Weighted Random Forest (WRF) is another variant of random forest, which follows the idea of cost-sensitive learning. That is, a heavier penalty would be placed on the misclassification of the minority class, by assigning the minority class a larger weight (i.e., higher misclassification cost). The class weights are used in two places of the RF algorithm. In the tree induction procedure, class weights are used to weight the Gini criterion for finding splits. In the terminal nodes of each tree, class weights are again taken into consideration. The class prediction of each terminal node is determined by "weighted majority vote"; i.e., the weighted vote of a class is the weight for that class times the number of cases for that class at the terminal node. The final class prediction for RF is then determined by aggregating the weighted vote from each tree, where the weights are average weights in the terminal nodes. More details of Weighted Random Forest can be found in (Effendy, Baizal et al. 2014).

4. Experiments

4.1. *Experimental setting*

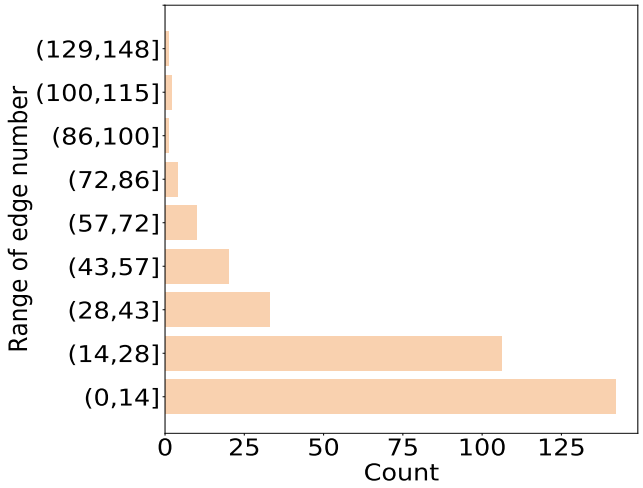
We have collected 320 public buildings from seven German cities: Frankfurt (60), Mannheim (28), Heidelberg (46), Karlsruhe (40), München (44), Stuttgart(38), Berlin (40), and Köln (24). The digital in the parentheses denotes the number of buildings collected in the corresponding city. The true main entrance of the buildings are identified in three ways. (1) On OSM, some buildings have been tagged with the main entrance or entrance[2]. Then, the main entrance or the only entrance node of the building is selected. (2) In some cases, the main entrance can be manually identified by humans from the satellite images and Google Street View. (3) The main entrance can be collected through site-survey if the first two ways fail. In the last two ways, the prior knowledge about the entrance is utilized to distinguish if the entrance is the main entrance or the ancillary entrance when more than one entrance is found. For instance, the logo or information board of an institute normally appears around the main entrance or it is more salient than the ancillary entrance in size and shape. We use IGIS.TK and its spatial data model to export the OSM data of the seven cities into the Spatialite database, from which the corresponding OSM entities around a building are retrieved (Noskov and Zipf 2018). Specifically, the OSM elements (i.e., node, way, and relation) that locate in or intersect with the buffer of the building are retrieved from the database. The buffer takes the main entrance of the building as the center and 150 meters as the radius, which is large enough to contain the associated contexts of the building entrance. A larger buffer, however, might cause the high computational cost since much more contexts would be analyzed. The corresponding SQL script is as *'SELECT elements.id, AsGeoJson(Transform(geom,32630)),keys.txt, vals.txt FROM elements JOIN tags ON elements.id=tags.id JOIN keys ON tags.key = keys.rowid join vals on tags.val=vals.rowid WHERE MbrIntersects(Transform(Buffer(Transform(MakePoint(8.3728, 49.0159, 4326), 32630), 150), 4326),elements.geom)'*. (8.372814, 49.015944) represents the latitude and longitude coordinates of the main entrance of a building that should be modified as buildings when querying the corresponding buffer. Based on the retrieved result, the required OSM entities and spatial contexts of a building can be then extracted.

[2] https://wiki.openstreetmap.org/wiki/Key:entrance



(a) distribution of area

(b) distribution of perimeter



(c) distribution of edge number

Figure 7. Distribution of area, perimeter, and edge number of test buildings

Furthermore, we analyze the distribution of the perimeter, area, and number of edges of the total buildings, which is shown in Figure 7. The x-axis denotes the count of the range of attributes. In each range, the open parenthesis means the start of the range is exclusive and the close bracket means the end of the range is inclusive. Thus, (0,2129] in Figure 7 (a) denotes the building whose area is larger than 0 and smaller than or equals 2129 square meters. We can observe that the shape and size of the buildings vary greatly. Then, the spatial contexts of the buildings and the symmetric buildings (with axis) are analyzed. The occurrence frequency of different spatial contexts and the symmetric building are shown in Figure 8. We can see the missing data issue is quite serious with only the frequency of the main road, service way, and address street over 0.7, making the classification task much challenging. To know which feature is important in recognizing the main entrance, we measured the importance of each feature (84 in total) by calculating how much the accuracy decreases when the feature is excluded in the random forest. From which, the top 20 most significant features are picked out, and their normalized weights are shown in Figure 9. 'centroid_sort', 'proportion', 'to_centroid', 'main_dis_sort', 'b_oa', 'service_dis_sort', 'service_ratio', 'oa_sort', 'oa', 'address_dis_sort', 'b_oa_sort', 'main_ratio', 'main_vis_sort',

'main_vis', 'service_vis', 'service_dis', 'main_dis', 'service_vis_sort', 'address_ratio', and 'oppo_shape' denote the ***sort of distance to centroid***, ***proportion***, ***distance to centroid***, ***sort of shortest path distance to main road***, ***distance to nearest building***, ***sort of shortest path distance to service ways***, ***turning degree of shortest path distance to service ways***, ***sort of open area***, ***open area***, ***sort of shortest path distance to address street***, ***sort of distance to nearest building***, ***turning degree of shortest path distance to main road***, ***sort of visibility degree from main road***, ***visibility degree from main road***, ***visibility degree from service way***, ***shortest path distance to service way***, ***shortest path distance to main road***, ***sort of visibility degree from service way***, ***sort of turning degree of shortest path distance to address street***, and ***opposite shape***, respectively. They play the most significant role in identifying the location of the main entrance. The axis related features are not ranked among the top 20 as we expected because only a small proportion of buildings are symmetric.

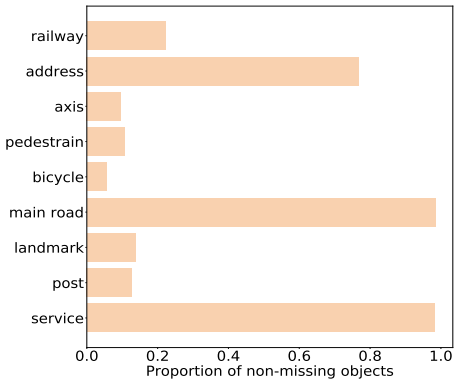


Figure 8. Occurrence frequency of spatial contexts and symmetric buildings in test buildings

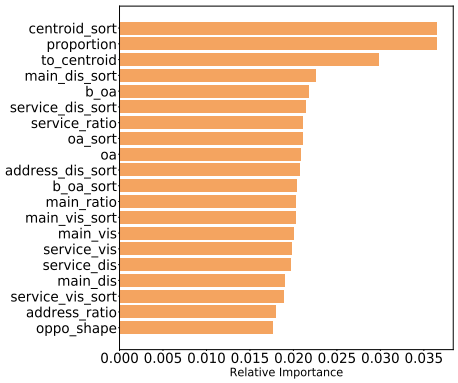


Figure 9. Importance of top 20 features

4.2. *Tagging accuracy*

In this experiment, we compare the three classification models for the imbalanced class issue and the general random forest model. A couple of important parameters need to be set for our proposed solutions to achieve the optimal performance. Specifically, the physical distance threshold (P_T) and the feature distance threshold (F_T) that are used to select the 'strong' negative samples are set to 24 (m) and 0.04, respectively. For the WRF approach, the important parameters include the number of trees (w^t), the maximum depth of the tree (w^d), and the weight of the minority class compared to the majority class (w^w), which are set to 80, 12, and 160 (160:1), respectively. For the BRF approach, the key parameters include the number of trees (b^t) and the maximum depth of the tree (b^d), which are set to 140 and 14, respectively. For the SmoteBoost approach, the key parameters include the number of new synthetic samples per boosting step (s^s), the maximum number of estimators (s^e) at which boosting is terminated, and the number of the nearest neighbors that are used to generate new samples for a minority class sample (s^n), which are set to 130, 90, and 4, respectively. For the general RF approach, the important parameters include the number of trees (r^t) and the maximum depth of the tree (r^d) , which are set to 110 and 14, respectively. These approaches

are implemented based on scikit and the imbalanced-learn package of Python.

The five-fold cross-validation is used to evaluate the approaches based on 320 public buildings. That is, the 320 buildings are divided into five test groups with each containing 64 buildings. In each test group, the 64 buildings are treated as the test set, and the remaining 256 buildings are treated as the training set, in which the location of the main entrance is known. We measure the deviation between the true entrance and the estimated entrance in two ways. The first is the shortest linear distance between them along the perimeter of a building polygon. The second is the shortest path distance from the estimated entrance to the true entrance. Note that, due to the existence of obstacles such as barriers and buildings, the path distance between two locations might be much larger than their linear distance, as shown in Figure 10.

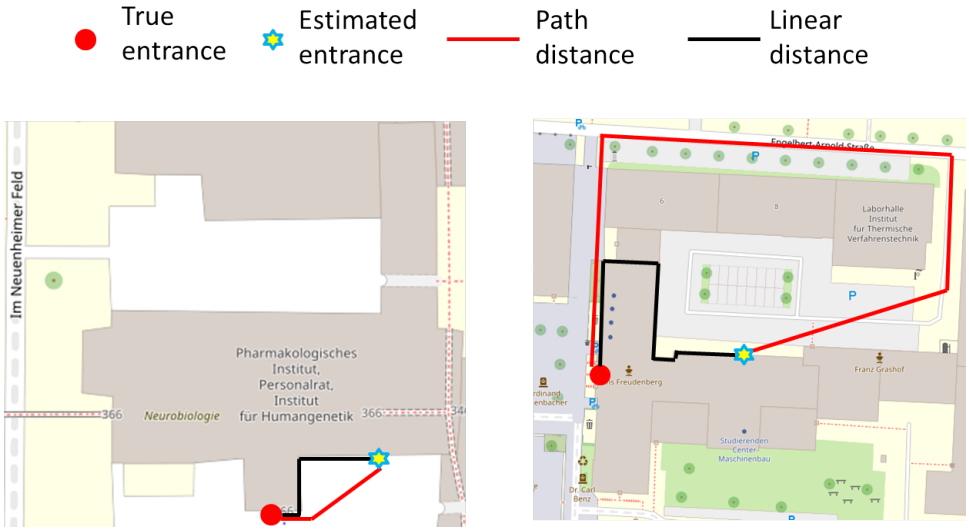


Figure 10. Two kinds of distance errors. In the left figure, the path distance is smaller than the linear distance. In the right figure, the path distance is much larger than linear distance due to the obstruction of buildings

In the Appendix, we present the tagging results of partial testing buildings by using the four models. In the figures, the red square denotes the position of the true entrance, while the brown upper-pointing triangle, the yellow star, the light blue diamond, and the blue right-pointing triangle denote the estimated position of the entrance by WRF, BRF, RF, and SmoteBoost, respectively. The complete data set, python code, and tagging results have been uploaded online. Figure 11 shows the cumulative linear distance error of the total five test groups. We can see WRF and BRF achieve the best tagging result with an average error of around 21 meters. 30% of the buildings are correctly tagged with the linear distance error at 0 meters, and in 80% of the cases, the distance error is below 30 meters. For SmoteBoost and the general random forest approaches, the mean error is around 35 meters. BRF and WRF can better deal with the imbalanced class issue than the SmoteBoost and RF approaches in this context.

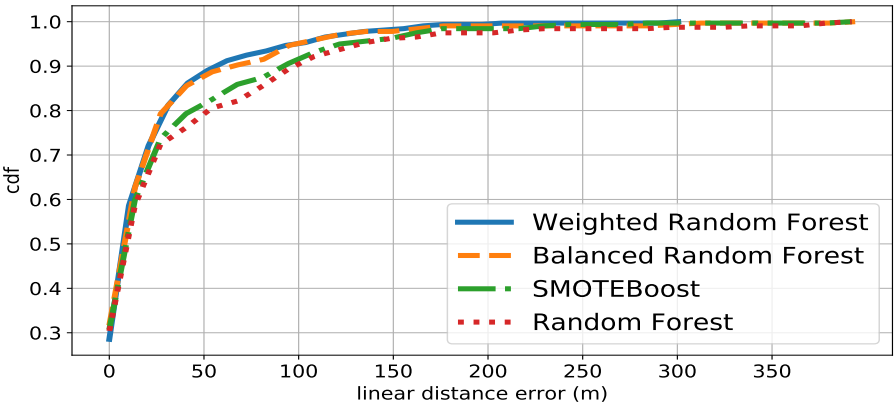


Figure 11. CDF of linear distance error of four classification models

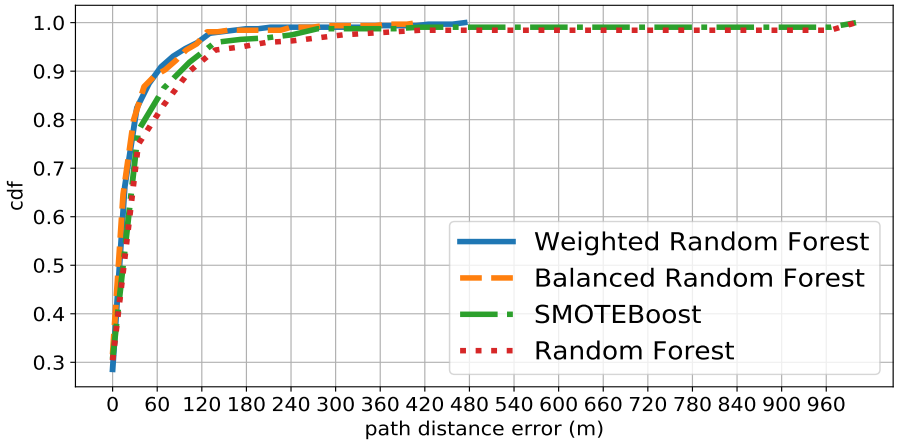


Figure 12. CDF of path distance error of four classification models

However, the liner distance error between the estimated and the true entrance does not reflect the actual walking distance that users need to take from the estimated entrance to the true entrance due to the existence of obstacles, including buildings and barriers (e.g., fence) in this context, as shown in Figure 10. Therefore, we further calculate the shortest path between the estimated and true entrance for the five test groups. If the true entrance is unreachable from the estimated entrance, the shortest path distance is set to 1000 meters. Figure 12 shows the CDF of the path distance error of the four approaches. We can see, BRF and WRF still achieve a promising result, with a mean error at 22 meters, and in 80% of the cases, the path distance error is below 30 meters. However, for SmoteBoost and the general RF approaches, the path distance error becomes larger at 38 and 46 meters, respectively, compared to their liner distance errors. We believe that a distance deviation at around 30 meters would not cause the failure of finding the true entrance because humans have powerful spatial cognition capability (Foo et al. 2005). For instance, pedestrians can easily find the entrance when they are following the route to the estimated entrance if the estimated and the true entrance is not far away.

Furthermore, we analyzed the test buildings whose linear distance error is over 60 meters. We found that three reasons mainly cause the large tagging error. The first

is inaccurate or incomplete OSM data. For instance, the building in Figures (k) and (u) of the Appendix have a tagging error over 60 meters. A fence in front of the estimated entrance location of Figure (k) by WRF is missing on OSM, leading to the estimated entrance easily accessed from roads. Likewise, a deep hole in front of the estimated entrance location of Figure (u) is missing on OSM. The second reason is that the ancillary entrance sometimes shows similar patterns to the main entrance such that the model misclassifies the ancillary entrance as the main entrance, as shown in Figures (s) and (ad) of Appendix, where the estimated location by WRF is close to the ancillary entrance. The third reason is that there are always numbers of exceptional buildings that do not follow the general layout principles of the main entrance.

Finally, we analyze the ranking of positive probability assigned to the true entrance sample among the total samples in a building. The ranking result are grouped into two types. The first is the absolute ranking result among all the samples with the value ranging from 1 to N, where N represents the number of samples in a building. Figure 13 shows the CDF of the absolute ranking result achieved by the four approaches. Still, BRF performs the best. In 55% of the cases, the true positive sample is ranked among Top 4. In 75% of the cases, it is ranked among Top 10. The second is the relative ranking result, which considers the varying number of samples in test buildings. It is calculated by dividing the absolute ranking result by the total number of samples in the corresponding building, limiting the value in the range of zero to one. Figure 14 shows the CDF of the relative ranking result achieved by the four approaches. Likewise, BRF performs the best. In 50% of the cases, the true entrance sample is ranked among the top 2%. In 74% of the cases, it is ranked among the top 10%. The ranking result looks promising, which proves that the trained models (i.e., BRF and WRF) are robust and effective.

To achieve the best classification accuracy, the general RF algorithm would mainly learn the patterns of the negative samples and classify nearly all the samples as negative since the negative ones are much more than positive ones. It pays little attention on the positive samples. Therefore, it achieved the worst tagging accuracy. The SmoteBoost approach uses the neighboring positive samples of a positive sample to synthesize new positive samples to keep the balance between the positive and negative samples. It to a certain degree can mitigate the imbalance issue. However, the synthesized positive samples normally contain noise, which would decrease the final tagging accuracy. BRF undersamples the negative samples while WRF gives the minor class more weight in classification. Both of them can deal with the imbalance issue and meanwhile do not introduce noisy samples. Therefore, they achieve better tagging performance than SmoteBoost and RF.

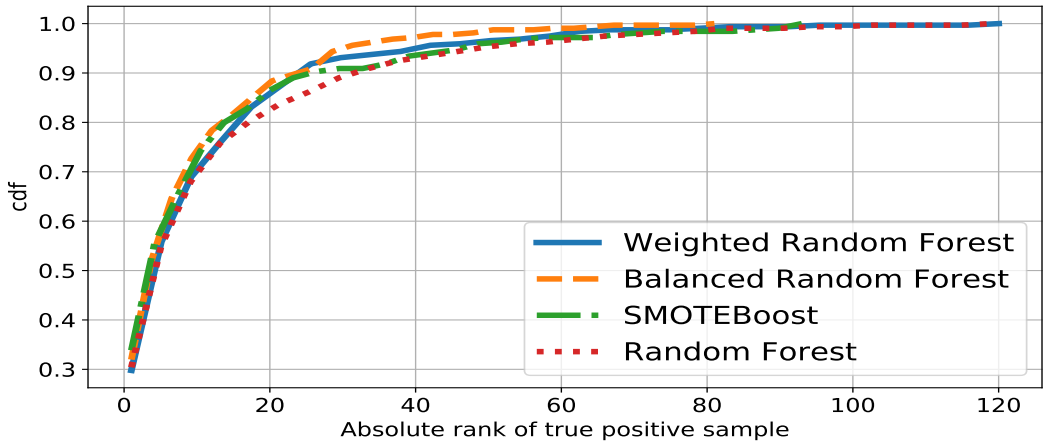


Figure 13. CDF of absolute ranking result of estimated positive probability of true entrance by four classification models

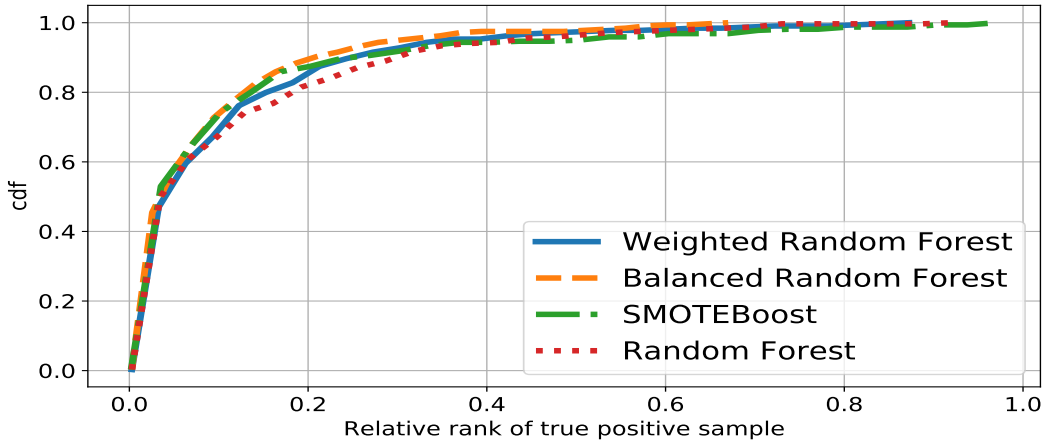
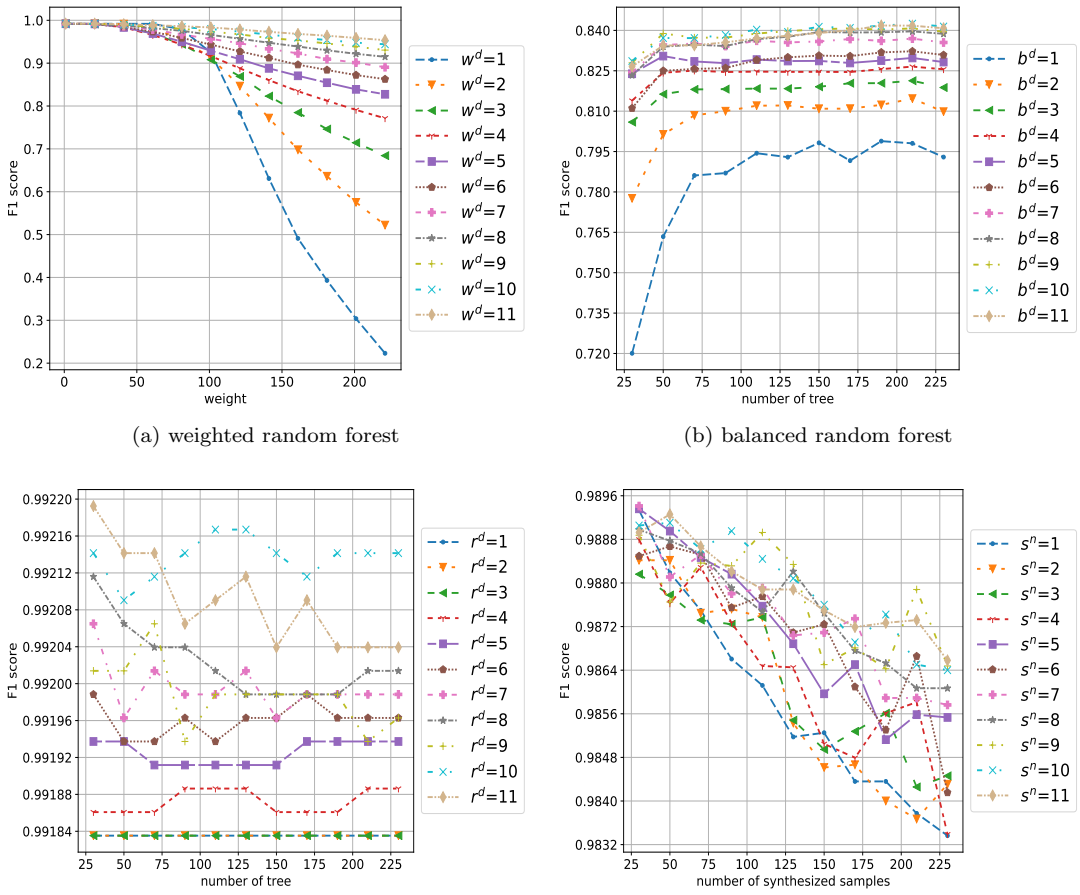


Figure 14. CDF of relative ranking result of estimated positive probability of true entrance by four classification models

4.3. *Impact of key parameters on F1 score and tagging error*



(a) weighted random forest

(b) balanced random forest

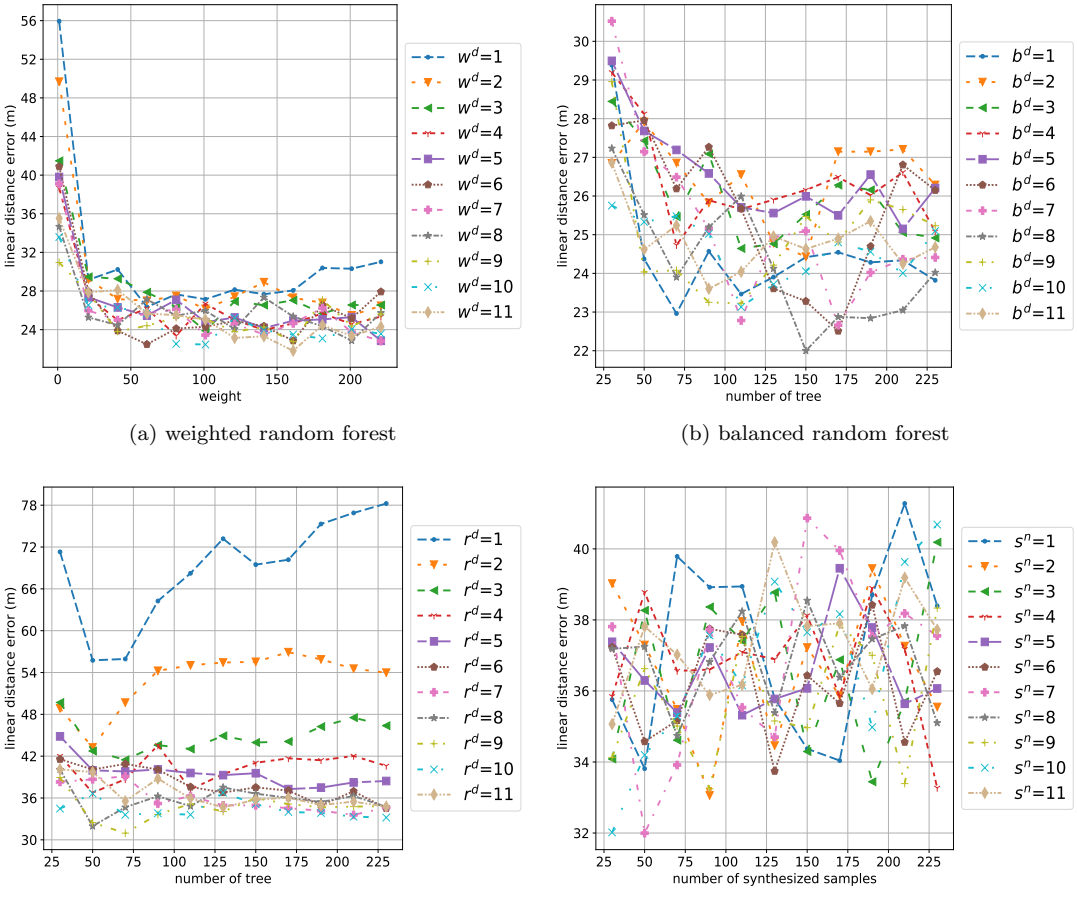
(c) random forest

(d) smoteboost

Figure 15. Impact of key perameters on F1 score of different models

The impact of the key parameters of the four classification algorithms on the F1 score is first analyzed, which is shown in Figure 15. The F1 score is calculated by counting the total true positives, false negatives, and false positives, without considering the imbalance issue. From the figure, we can see that in average RF achieves a higher F1 score than the other three algorithms. The model would classify most of the examples as negative to achieve the highest classification accuracy, which leads to a high F1 score. The tree number (r^t) and the maximum depth of the tree (r^d) have a small impact on the F1 score since the imbalance issue remains. For the SmoteBoost approach, s^e is set to 210. A downward trend can be observed as the increase of the generated samples per positive sample (s^s). As the number of positive samples is approximating that of negative samples, the model would classify more samples as positive. This leads to the decrease of the F1 score considering the fact that the negative samples in the data set are far more than the positive samples. For WRF approach, w^t is set to 70. Likewise, the F1 score shows a substantial downward trend as the increase of the weight (w^w) since more negative samples are classified as positive. For BRF approach, the negative samples are under-sampled. Thus, an equal weight is assigned to the negative and positive samples, which lead to a lower F1 score compared with the SmoteBoost and RF algorithms. It can be also observed that as the increase of the tree number (b^t) and maximum tree depth (b^d), the F1 score increases and remains stable when the two parameters are over 100 and 8, respectively. In conclusion, it is difficult to evaluate

the performance of the classification models through the F1 score when the class is highly imbalanced especially in our issue where only the relative probability matters. Therefore, the impact of the key parameters of the four classification algorithms on the linear distance error is also analyzed, which is shown in Figure 16. In general, WRF and BRF achieve a lower tagging error than RF and Smoteboost algorithms. For the WRF approach, w^t is set to 70. As the increase of the weight (w^w), the tagging error is decreasing and remains stable when it is over 100. A larger weight assigned to the positive samples can make the model pay more attention to the positive samples. Likewise, the tagging error decreases as the increase of the maximum tree depth (w^d) and remains stable when it is over 9. The deeper the tree, the more features could be utilized. The BRF approach achieves a similar tagging performance to WRF and the change of the tree number and maximum depth does not affect the tagging error dramatically although a slight downward trend can be observed as the increase of the tree number. Compared to BRF, the RF approach is dramatically affected by the tree number and maximum depth. The tagging error decreases as the increase of r^d and remains stable when the value is over 7. For Smoteboost, s^e is set to 210. No stable trend can be observed from the result of the Smoteboost algorithm with the change of (s^s) and (s^n). This is mainly because the synthesizing process can mitigate the imbalance issue but meanwhile introduce incorrect positive samples to the training set.



(a) weighted random forest

(b) balanced random forest

(c) random forest

(d) smoteboost

Figure 16. Impact of key perameters on liner distance error of different models

5. Discussions

Main entrance assumption: One of the assumptions of the proposed solution is that there is one and only one main entrance in a public building. This is due to two reasons. First, in most of the cases, this assumption holds. Second, it is challenging to detect a variable number of main entrances in a public building if we are uncertain how many main entrances exist. However, when collecting the test buildings, we also found that a public building could be comprised of multiple departments with each having one house number and one main entrance. Such buildings are beyond the scope of this study. However, this will be dealt with in the future work considering the house number tagged on OSM since each house number corresponds to a main entrance. That is, multiple main entrances can be identified from a building if the tagged location of the house number is known.

Fusion of OSM and satellite imagery: As we have mentioned in the experimental section, the tagging error is often caused by missing or incomplete data in OSM. This greatly reduces the applicability and robustness of the proposed solution. To mitigate the issue, in the future, we plan to use the satellite imagery (e.g., from Bing map) to provide more cues about the possible locations of the main entrance. For instance, in Figure 17, an open space is in front of the main entrance, which can be identified from the satellite imagery. However, by using only the data from OSM, a big tagging error is produced, as shown in Figure (v) of Appendix. One of the cues of the impossible entrance position is the front green space, as shown in Figure 18, which can be observed from the satellite imagery. However, with only the OSM data, the estimated entrance by BRF is located at the green space, as shown in Figure (ab) of Appendix. The possible solution is to combine the manually defined features extracted from OSM, and the features automatically learned from the satellite imagery with deep learning in an integrated model. The other reason that the satellite imagery should be introduced is because in certain countries, such as China, the OSM data is poor with coarse road networks and little building information except the footprint and name. In this case, the satellite imagery can play dominant roles in predicting the building entrance.

Variation of building style in time and space: We tested our approaches with the buildings in only Germany. The buildings from the other countries are not considered, such as in Asian counties where the building styles might be totally different and the entrance location might also vary dramatically. Therefore, we cannot guarantee that the trained model can be applied to predict the main entrance of public buildings in other countries. However, we still believe this work is valuable since a local model such as for Germany is still useful, considering the large number of buildings in Germany. To achieve a globally applicable model, the spatial location properties (e.g., country and continent) of buildings could be added to the feature set and the buildings across different countries and continents are trained together. By doing so, the trained model can adapt to the change of spatial location. In this study, the construction time of the buildings is ignored. However, it can also be introduced to make the model more general that can accurately predict the main entrance of buildings across both space and time given abundant annotated data.

Figure 17. Blue square indicates an open space where the main entrance is located

Figure 18. Yellow square denotes a green space where the main entrance is not likely to be located

6. Conclusion

To mitigate the misleading and inaccurate navigation issues caused by the missing main entrances of public buildings on current map providers (e.g., Google Maps and OSM), we proposed a broadly applicable main entrance tagging approach based only on the association between spatial elements extracted from OSM. Three classification algorithms have been applied to model the association relationship and deal with the imbalanced class issue, namely WRF, BRF, and SmoteBoost. Experimental results show that WRF and BRF have a low tagging error in both linear distance and shortest path distance errors, which we believe can greatly save pedestrians' effort in finding the main entrance. We also found the most frequent tagging error is normally caused by inaccurate and incomplete OSM data. Realizing this interesting finding, we will investigate the possibility of automatically reporting erroneous data on OSM based on the tagged entrance since the big tagging error might be related to erroneous OSM data. Apart from this, in the future, we plan to combine the satellite imagery to provide further evidences about the possible location of the main entrance to mitigate the large tagging error and to overcome the poor OSM data challenge faced in certain countries.

7. Data and codes availability statement

The data and codes that support the findings of this study are available in entrance_tagging with the identifier at the private link https://figshare.com/s/00612ebbc369a980bd7b

References

Chawla, Nitesh V, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. "SMOTE: synthetic minority over-sampling technique." *Journal of artificial intelligence research* 16: 321–357.

Chawla, Nitesh V, Aleksandar Lazarevic, Lawrence O Hall, and Kevin W Bowyer. 2003. "SMOTEBoost: Improving prediction of the minority class in boosting." In *European conference on principles of data mining and knowledge discovery*, 107–119. Springer.

Deng, Yi, Changgee Chang, Moges Seyoum Ido, and Qi Long. 2016. "Multiple imputation for general missing data patterns in the presence of high-dimensional data." *Scientific reports* 6: 21689.

Effendy, Veronikha, ZK Abdurahman Baizal, et al. 2014. "Handling imbalanced data in customer churn prediction using combined sampling and weighted random forest." In *2014 2nd International Conference on Information and Communication Technology (ICoICT)*, 325–330. IEEE.

Foo, Patrick, William H Warren, Andrew Duchon, and Michael J Tarr. 2005. "Do humans integrate routes into a cognitive map? Map-versus landmark-based navigation of novel shortcuts." *Journal of Experimental Psychology: Learning, Memory, and Cognition* 31 (2): 195.

Goetz, Marcus, and Alexander Zipf. 2011. "Extending OpenStreetMap to indoor environments: bringing volunteered geographic information to the next level." *Urban and regional data management: UDMS annual* 2011: 47–58.

Hart, Peter E, Nils J Nilsson, and Bertram Raphael. 1968. "A formal basis for the heuristic determination of minimum cost paths." *IEEE transactions on Systems Science and Cybernetics* 4 (2): 100–107.

Hochmair, Hartwig H, Dennis Zielstra, Pascal Neis, PN Hartwig, H Hochmair, and D Zielstra. 2013. "Assessing the completeness of bicycle trails and designated lane features in OpenStreetMap for the United States and Europe." In *Transportation Research Board Annual Meeting*, .

Hu, Xuke, Lei Ding, Jianga Shang, Hongchao Fan, Tessio Novack, Alexey Noskov, and Alexander Zipf. 2020. "Data-driven approach to learning salience models of indoor landmarks by using genetic programming." *International Journal of Digital Earth* 1–28.

Hu, Xuke, Hongchao Fan, and Alexey Noskov. 2018. "Roof model recommendation for complex buildings based on combination rules and symmetry features in footprints." *International Journal of Digital Earth* 11 (10): 1039–1063.

Hu, Xuke, Hongchao Fan, Alexey Noskov, Alexander Zipf, Zhiyong Wang, and Jianga Shang. 2019. "Feasibility of Using Grammars to Infer Room Semantics." *Remote Sensing* 11 (13): 1535.

Kang, Suk-Ju, Hoang-Hon Trinh, Dae-Nyeon Kim, and Kang-Hyun Jo. 2010. "Entrance detection of buildings using multiple cues." In *Asian Conference on Intelligent Information and Database Systems*, 251–260. Springer.

Khalilia, Mohammed, Sounak Chakraborty, and Mihail Popescu. 2011. "Predicting disease risks from highly imbalanced data using random forest." *BMC medical informatics and decision making* 11 (1): 51.

Liu, Jingchen, Thommen Korah, Varsha Hedau, Vasu Parameswaran, Radek Grzeszczuk, and Yanxi Liu. 2014. "Entrance detection from street-view images." In *IEEE International Conference on Computer Vision and Pattern Recognition Workshop (CVPR), Columbus*, .

Liu, Jingchen, Vasudev Parameswaran, Thommen Korah, Varsha Hedau, Radek Grzeszczuk, and Yanxi Liu. 2017. "Entrance detection from street-level imagery." Oct. 24. US Patent 9,798,931.

Murillo, Ana Cris, Jana Košecká, Jose Jesus Guerrero, and Carlos Sagüés. 2008. "Visual door detection integrating appearance and shape cues." *Robotics and Autonomous Systems* 56 (6): 512–521.

Nikoohemat, S, M Peter, S Oude Elberink, and G Vosselman. 2017. "Exploiting Indoor Mobile Laser Scanner Trajectories for Semantic Interpretation of Point Clouds." *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences* 4.

Noskov, Alexey, and Alexander Zipf. 2018. "Open-data-driven embeddable quality management services for map-based web applications." *Big Earth Data* 2 (4): 395–422.

Quintana, Blanca, Samuel A Prieto, Antonio Adán, and Frédéric Bosché. 2018. "Door detection in 3D coloured point clouds of indoor environments." *Automation in Construction* 85: 146–

166.

Schapire, Robert E. 2013. "Explaining adaboost." In *Empirical inference*, 37–52. Springer.

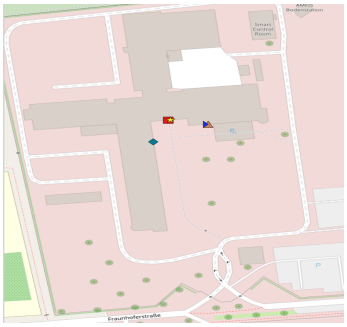
Sun, Yanmin, Andrew KC Wong, and Mohamed S Kamel. 2009. "Classification of imbalanced data: A review." *International Journal of Pattern Recognition and Artificial Intelligence* 23 (04): 687–719.

Talebi, Mehdi, Abbas Vafaei, and Amirhassan Monadjemi. 2018. "Vision-based entrance detection in outdoor scenes." *Multimedia Tools and Applications* 77 (20): 26219–26238.

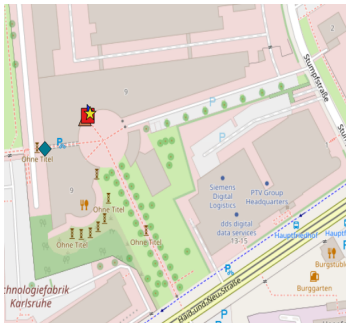
Tang, Fei, and Hemant Ishwaran. 2017. "Random forest missing data algorithms." *Statistical Analysis and Data Mining: The ASA Data Science Journal* 10 (6): 363–377.

Zeng, Limin, and Gerhard Weber. 2015. "A pilot study of collaborative accessibility: How blind people find an entrance." In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*, 347–356. ACM.

Zhao, Yiyang, Chen Qian, Liangyi Gong, Zhenhua Li, and Yunhao Liu. 2015. "LMDD: Lightweight magnetic-based door detection with your smartphone." In *2015 44th International Conference on Parallel Processing*, 919–928. IEEE.

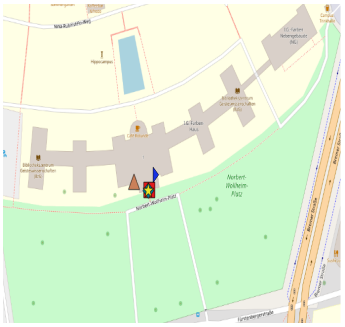


(a)



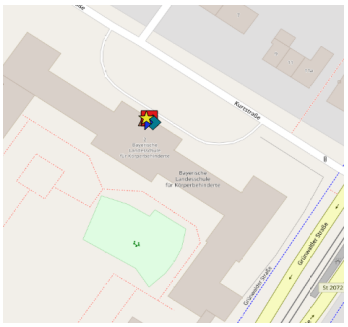
(b)

(c)



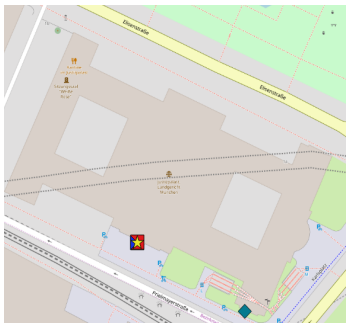
(d)

(e)



(f)

(g)

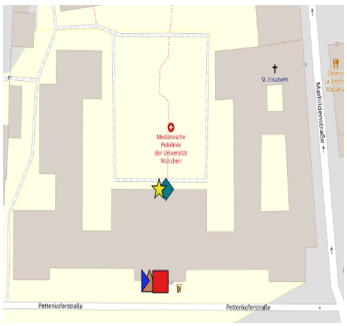


(h)

(i)

(j)

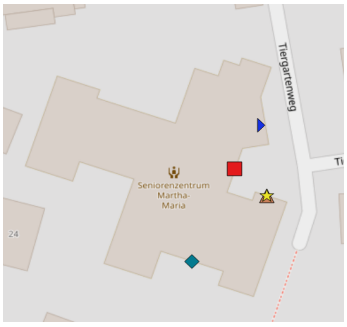
(k)



(l)

(m)

(n)



(o)

(p) (q) (r)

(s) (t) (u)

(v) (w) (x)

(y) (z) (aa)

(ab) (ac) (ad)

Figure 0. Tagging result of partial test buildings