



Digital Urban Twins for heavy rain events - An open source QGIS plugin for machine learning classification of residential buildings using CityGML with additional datasets

Alexander Bong¹, Christian Clemen¹

¹University of Applied Sciences Dresden, Faculty of Spatial Information, Germany
alexander.bong@htw-dresden.de - ORCID  0009-0006-3919-6065
christian.clemen@htw-dresden.de - ORCID  0000-0002-5807-7698

Keywords: Digital Twin, Heavy Rain, City Model, Residential Buildings, Classification, Potential Damage.

Abstract

Extreme weather events such as heavy rains are an increasing challenge. The potential impact of flooding on residential buildings can be simulated using digital twins. However, when using geometric-semantic information from diverse data resources, such as 3D city models, zoning or cadastre, the data must be carefully selected and programmatically prepared for the simulation. In this study, we present how a use-case driven classification was generated for the residential buildings in the city of Dresden, which is used to estimate the damage potential. The research focuses on both the supervised building classification with a neural network and the open source software framework. Data management is done with the 3DCityDB in PostgreSQL. QGIS is used for visualisation and user interaction. The Python-plugin automatically classifies more than 70,000 residential buildings based on 37 residential building classes. The hierarchical classification is challenging due to the ground truth sample size of about 21,000 and the heterogeneous distribution of the samples. The core of the method is the training and validation utilising random forest as machine learning method. With the developed toolset, classification results can be visually checked in a subsequent step using QGIS. Additionally, the classification, might be corrected manually for individual buildings using mobile mapping data, if necessary. Eventually, the assigned classes are fed back into the official CityGML city model as a new attribute, enabling a realistic damage potential analysis, in a free and publicly available 3D-WebGIS platform. The project is funded under the Smart Cities pilot programme of Germany.

1. Introduction

1.1 Motivation

Due to climate change, extreme weather events, such as heavy rainfall, are becoming increasingly frequent. These cause varying degrees of damage to buildings. The damage can be simulated using digital twins. However, in this simulation, a distinction must be made between different classes of buildings. Each class has a specific combination of characteristics and therefore reacts differently to heavy rain events. A correct and comprehensive simulation of the damage potential is only possible if each building can be assigned to a corresponding class. This is to be carried out for the residential buildings in the city of Dresden. In 2025, there are more than 90,000 residential buildings in the urban area. Manual mapping would take a long time. Part of the Smart Cities project - Digital Twin for Heavy Rainfall will be an application that automatically classifies the buildings (Fig. 1). This includes open geodata that is merged in a database. The implementation as a plugin for the QGIS software allows the employees of the city of Dresden to carry out this classification again with new data. It is also possible to check the results of each step and make manual corrections. The software will then be made freely available for use and adaption in other areas of Germany and internationally.

1.2 Scope Statement

This work is not basic research, but a descriptive case study. A new classification is developed in relation to the city of Dresden. The application does not represent a universal solution for all cities or for any arbitrary city. However, the procedure can be adapted to other areas if the relevant data is available and the attributes of the buildings are recorded.

Geospatial attributes and feature-vectors for machine learning can be adapted for different locations and scenarios.

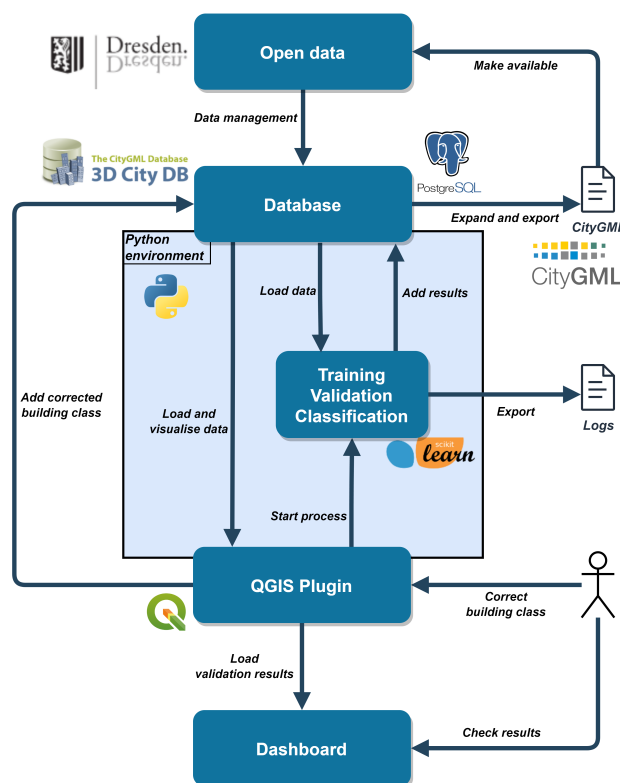


Figure 1. System architecture of the developed tool to classify residential buildings.

1.3 Research Questions

Aspects of the classification of residential buildings in Dresden are analysed in detail. In particular, the following research questions are considered and answered:

- How can residential buildings in Dresden be classified according to their construction type and age?
- Which characteristics can be used for classification?
- How can the classification be implemented automatically for all residential buildings in Dresden?
- Which reliability of the classification can be achieved?
- How can the tool be adapted to other cities?

2. Related Research

The use of 3D city models based on the CityGML standard of the Open Geospatial Consortium (OGC) is becoming increasingly important. The 3D City Database makes it easier to develop and expand a city model, as described in (Yao et al., 2018). It provides a relational database schema for managing city objects. For example, buildings with their geometry and semantic attributes can be stored centrally. This can form the basis for classifications.

In urban areas, there is often many heterogeneous data available. This data must be combined to be used effectively in various applications. The processing of geometric information at a low level and semantic information at a higher level has been addressed by (Sideris et al., 2019). Using a semantic model, the authors merge the data to make it usable for machine learning. The data is then used in various scenarios to test several classification approaches. Based on the metrics Accuracy, Specificity, Precision, Recall, F1 Measure, and G-Mean, the authors recommend the random forest classifiers as decision tools for urban planning tasks. These are also robust against deviating observations. However, human intuition is categorised as the ultimate decisive factor for such problems.

Machine learning can also be used to predict building characteristics. The study by (Lei et al., 2024b) addresses this problem using spatial buildings and the streets surrounding them. According to the authors, graph neural networks cover the spatial relationships of buildings to their environment better than models such as random forest do. In the paper, a model was developed that can predict certain building features, such as the number of storeys, as well as solve classification tasks. The authors compare their approach with random forest models in three experiments. The building storeys, building type, period of construction and material are determined in the cities of Boston, Melbourne and Helsinki. In comparison, they achieve slightly better results with their newly created model.

Determining the exact age of residential buildings is another discipline of classification. Using 3D GIS (Geographic Information Systems) models and machine learning, (Biljecki and Sindram, 2017) have developed an approach for estimating the year of construction of buildings with a Level of Detail 1 (LOD1). Random forest models from the scikit-learn library were trained using various attributes such as building height, number of floors, volume and number of neighbouring buildings. The city model of Rotterdam was chosen for this purpose. Half of the

datasets were used as training datasets, the other half as validation datasets. The authors note that categorisation into a decade is very possible if all attributes are available for each building.

The necessity for a consistent survey of building attributes is also postulated by (Lei et al., 2024a). Morphological characteristics are just as important for classifications in various application scenarios. In particular, expanding the data to include attributes of visual perception opens up new possibilities. The authors also used the scikit-learn library to train random forest models using information from building images. They have extended the existing CityJSON dataset of the city of Amsterdam. The authors mention that it is important to extend digital urban twins and 3D city models with social components. They plan to investigate new use cases in the future.

An automatic classification of building floor plans was investigated by (Hecht, 2014). The author tested various machine learning algorithms and preferred the use of random forest models as a non-linear classification method. These models have the highest generalisation capability and the shortest runtime. The analyses are made with various open geodata. Hecht emphasises that geometric, semantic, topological and statistical attributes are necessary for reliable classifications. The author continues to criticise the studies considered up to this point with a maximum of five classes, as it is possible to determine more classes. A differentiation into more classes offers a differentiated view of the settlement structure and is therefore desirable. With a sufficient amount of training data, an accuracy of over 90 % is achieved for nine out of eleven classes. This shows that reliable categorisation is not always possible for all building types.

The categorisation of buildings to determine potential damage has already been analysed by (Vetter et al., 2024). The authors have developed an approach that uses open geodata and creates parametric models. Georeferenced raster data and CityGML models are used. The models are used to determine the building age classes and floor plan geometries of the buildings. The buildings are then categorised using a semi-automatic approach. The authors describe the limited availability of data on the load-bearing structure of the buildings as the main challenge. The models work without the use of machine learning. However, the potential for expanding the possibilities through the additional use of machine learning is recognised.

The categorisation into building classes can be used not only for simulations of heavy rainfall events. It is also possible to carry out energy analyses, as shown by (Kaden et al., 2012). The authors have integrated data from solar potential analyses and the energy supply into the CityGML-compliant Energy Atlas of Berlin. The 3D city model serves as a link between the different ontologies. It is thus enriched with information relevant to environmental and energy planning. Based on the correlation between building characteristics and consumption information, the authors describe the use case of estimating the heating energy demand. The authors show that existing 3D city models can serve as a basis for merging open geodata and enable new use cases.

Further application scenarios are the prediction of hourly heat consumption in residential buildings and the expected short-term consumption of the buildings. To investigate this (Tognoli et al., 2023) trained, tested and applied two models to 500 buildings in Switzerland. In addition to building data in CityGML format, the authors also used weather data.

Both models were optimised by combining different regression models. The models achieve very good results. The first model can predict the heating requirements for any building. The second model can predict the heating demand for a group of buildings based on framework conditions. Overall, the authors have developed a workflow that can be adapted to other application scenarios.

Flooding is causing more damage to residential buildings due to climate change and urban development. New flood resistance technologies may be able to reduce the impact of flood damage. To analyse this (Golz et al., 2014) used such technologies in a GIS-based flood damage simulation model to support the evaluation of these strategies. The authors extended their study by some steps (Schinke et al., 2016). They used the high resolution of GIS and the characteristic properties of the individual buildings. Their synthetic model enables the spatial damage and risks to each building to be analysed. To demonstrate and validate their approach, they present a case study in Valencia, Spain.

3. Initial Data

The source data is provided by the City of Dresden and the Dresden Office for Geodata and Cadastre. Data and concepts from previous research projects are also accessed. The fundamental building data is freely accessible and can also be used subsequently. In addition, three data sets on monuments and newly built houses as well as a data set from a property service provider are used to include the age of buildings.

3.1 Open Data Dresden

The City of Dresden provides different data in the OpenData Portal Dresden. This includes a dataset of the city's parcels with their geometry and parcel numbers and a dataset of the construction types. The construction types are offered in a block map. It should be noted that the construction attributes are not managed on a parcel-by-parcel basis. Both data sets are used in this project in CSV format and geometrically intersected. This means that all parcels receive a string value for the building construction. This includes, for example, A12 for detached houses or semi-detached houses or C11 for a perimeter block development.

The 3D city model in Level of Detail 2 is also downloaded as CityGML format from the OpenData portal. Building functions as the keys for residential, commercial and public buildings are updated with ALKIS building data, due to the model update intervals. ALKIS is the official cadastral information system in Germany. It contains parcel data in connection with building footprints.

3.2 Classification Concept

The classification of residential buildings is based on a classification concept developed by the Faculty of Civil Engineering of the University of Applied Sciences Dresden (Golz et al., 2014) and (Schinke et al., 2016). Table 1 explains the urban structure types of residential buildings in Dresden.

The concept includes a residential building matrix that shows 42 categories for possible combinations of building age and building type for residential buildings in Dresden (Fig. 2). The combination of structure type and building age level results in the residential building type. A building that is classified as EE3,

| Building type | Description |
|---------------|---|
| EE | Detached single-family house |
| HH | Backyard house |
| LW | Agriculturally characterised building |
| LWS | Agriculturally characterised building with stable use |
| ME | Detached apartment house |
| ER | Single-family house in a row |
| MRG | Apartment block in closed row development |
| MRO | Apartment block in open row development |

Table 1. Building types of the residential buildings in Dresden.

| Building Type | Urban Structure Type | free standing buildings (with one main entrance) | | | | | row standing buildings (each with one main entrance) | | |
|--|----------------------|---|-----|------------|------|-----|---|------------|------|
| | | single unit | | multi unit | | | single unit | multi unit | |
| | | EE | HH | LW | LWS | ME | ER | MRG | MRO |
| before 1870 timber frame construction | 1 | EE1 | | LW1 | | ME1 | ER1 | MRG1 | MRO1 |
| before 1870 brickwork | 2 | EE2 | HH2 | LW2 | LWS2 | ME2 | ER2 | MRG2 | MRO2 |
| 1870-1918 brickwork | 3 | EE3 | HH3 | LW3 | LWS3 | ME3 | ER3 | MRG3 | MRO3 |
| 1918-1945 mainly brickwork | 4 | EE4 | HH4 | LW4 | LWS4 | ME4 | ER4 | MRG4 | MRO4 |
| 1945-1990 brickwork | 5 | EE5 | | | LWS5 | ME5 | ER5 | | MRO5 |
| 1970-1990 prefab. concrete building | 6 | | | | | ME6 | | | MRO6 |
| after 1990 mainly brickwork | 7 | EE7 | | | | ME7 | ER7 | MRG7 | MRO7 |

Figure 2. Building typology of the residential buildings in Dresden based on (Golz et al., 2014) and (Schinke et al., 2016).

for example, is therefore a detached single-family house from the period between 1870 and 1918.

Each category is characterised by its typical design and generates different damage potentials during heavy rainfalls. Of over 90,000 residential buildings, 21,000 have already been mapped during on-site mapping in different regions of the city. The previous mapping campaign has already shown that not all categories are consistently represented in the mapped data. In addition, adjustments must be made with regard to class categorisation. It is not possible to accurately differentiate between agricultural buildings and those with an additional use as stables on the basis of the external appearance of the buildings alone. In addition, not all building ages are available in the recorded data, leaving a total of 30 categories to be assigned. This means that this study does not cover all possible types of residential buildings in Dresden.

4. Data Processing

All residential buildings in Dresden are held in the 3D City Database¹. This free and open-source spatial relational database is used for data storage, data management and visualisation of 3D city models. The data described in section 3 are compiled in a newly developed problem-specific schema. The database is used in the open-source object-relational database system PostgreSQL². The free, open-source software QGIS³ is used to visualise and analyse the geodata. The application, developed in the presented research project, is a plugin for QGIS and uses the object-oriented programming language Python⁴ to

¹ <https://github.com/3dcitydb> (Version 5.0.0)

² <https://www.postgresql.org/> (Version 16)

³ <https://qgis.org/> (Version 3.34.10 Prizren)

⁴ <https://www.python.org/> (Version 3.12.5)

perform calculations and load the data from the database into QGIS.

4.1 Hierarchical Classification

A hierarchical classification allows decisions to be simplified and specified step by step. The existing categories are integrated into several levels of a hierarchy. The first step is to decide whether the residential building under consideration is an apartment block, a detached house or another class. Multi-family houses and single-family houses are then given the additional information as to whether they are single or in a row. These MR, ME, ER and EE classes are then assigned a building age category. Numerous tests have shown that the most reliable classification for row multi-family houses is to differentiate between an open and a closed construction method on the last level. All other classes in the first level are subdivided into rear houses and agricultural buildings and are then also assigned a building age level (Fig. 3).

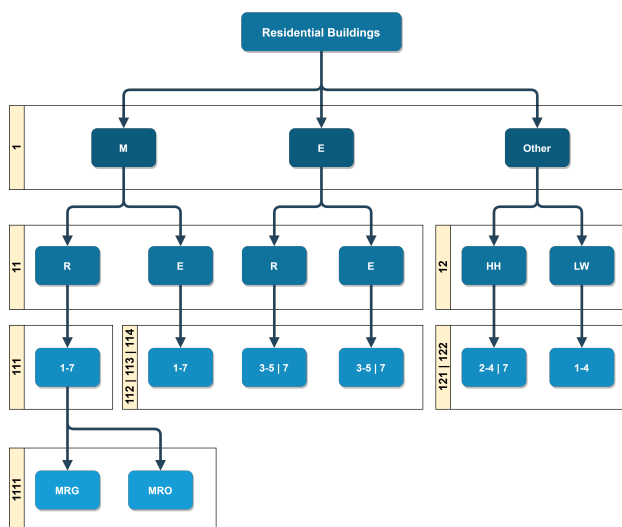


Figure 3. Hierarchy for the classifications of residential buildings.

4.2 Relevant Features

For classification, meaningful semantic and geometric attributes must be selected from the existing data to populate the feature vector for machine learning. In their entirety, they must clearly differentiate the individual classes or groups in the respective levels. Fifteen attributes were selected for this purpose, which were either taken directly or calculated.

These include:

- geometric attributes
 - ridge height [m]
 - eaves height [m]
 - storey height [m]
 - slope of the largest roof surface [degree]
 - size of the base area [m²]
 - width of the convex envelope of the base area [m]
 - length of the convex envelope of the base area [m]
 - number of vertices of the base area

- semantic attributes
 - building age
 - number of storeys
 - number of roof surfaces
 - roof type [categorical enumeration]
 - development type identified on the block map [categorical enumeration]

Topological characteristics are also determined. For this purpose, neighbourhood relations are used for a radius of 100 m around the building in question.

These are:

- building density [n buildings / 100 m radius]
- average building footprint [m²]
- minimum distance to neighbouring buildings [m]
- predominant class [categorical enumeration]

4.3 Feature Engineering

A static evaluation was carried out for each class and hierarchy level to analyse the attribute values. As the classes are not normally distributed in the given sample data, the significance level was calculated using the median. The calculated and collected attributes can also be used to determine ratios using feature engineering. These can provide robust results when individual attribute values deviate greatly from the median of the respective class. For example, the volume of the building and the ratio of building height to building floor area are calculated. If, for instance, the floor area of a building deviates significantly from the median of all buildings in the respective category, it is still possible to reliably determine which category the building belongs to. For this purpose, the volume and the ratio of floor area to height can be used to determine the category. In total eight ratios are calculated using feature engineering. The filtering of the city database and the calculation of the neighbourhood relations and the feature engineering attributes are done with an office notebook (Processor: 13th Gen Intel(R) Core(TM) i7-13700H with 2.40 GHz, RAM: 32.0 GB). This process lasts about three hours for the dataset of Dresden.

5. From Training to Newly Classified Buildings

Random Forest is an established machine learning-based classification model. In this project the initial data is available in tabular form, offering low complexity with numerical and categorical attributes. Furthermore, the feature space is consistent across the entire dataset. Due to the clearly defined classification problem, Random Forest was chosen over Graph Neural Networks and Gradient Boosting. The building classes are determined using the collected attributes for training random forest models. A model is trained and validated for each hierarchy level. The functions of the open-source library scikit-learn⁵ are used. The software package provides algorithms for the selection of the best random forest model, training and metrics for validation of all models. All data is predivided into training data with 17,000 buildings, validation data with 4,000 buildings and classification data with 70,000 buildings and stored in new relations.

⁵ <https://scikit-learn.org/stable/> (Version 1.6.1)

5.1 Model Training

The split between training data and validation data is randomised in a 80:20 ratio each time training is started from scratch and no existing model is used. Thus, 17,000 mapped residential buildings serve as the training data set and 4,000 buildings serve as the validation data set. Categories with zero or only one building in the given training dataset are excluded. The training is performed using GridSearchCV (Grid Search Cross Validation) to use the optimal model for each hierarchy level. After multiple tests, the hyperparameters in table 2 result in the best training of the models with this particular data set.

| Hyperparameter | Value range |
|-------------------|---------------------|
| n_estimators | [150, 250, 350] |
| max_depth | [15, 25, None] |
| min_samples_split | [5, 10] |
| min_samples_leaf | [2, 4] |
| criterion | ['gini', 'entropy'] |

Table 2. Best hyperparameter for all random forest models with their value ranges.

The parameters define value ranges for the number of trees and the maximum depth of the trees. A higher number of trees can lead to better model performance with a longer calculation time. A greater depth can achieve better results, but also increases the risk of overfitting. Overfitting is the strong adaptation of the model to the training data. This means that the model is less sensitive to changes in the input data. Also, it's determined by the number of data sets required for a leaf node and for further division and depth. The determination of the optimal parameters by the models results in consistently low values. This means that even those classes with a very small number of samples are well covered. The respective model remains flexible, but is also more susceptible to overfitting if noise occurs in a minority class. The hyperparameter criterion is set to the values 'gini' and 'entropy'. The parameter evaluates the quality of the distribution in a decision tree, with the aim of producing a homogeneous distribution of classes. The value 'gini' favours frequently occurring classes to avoid random misclassifications. In contrast, the use of 'entropy' tends to favour rare classes. All models use the value 'gini' as the best parameter. The use of other hyperparameters such as bootstrap, max_features or class_weights did not change the results. Due to the significantly higher computing time, these are not considered in the model training.

The training is performed with the numeric and categorical attributes mentioned above. The use of categorical attributes requires a label encoder to make them processable for the random forest model. In addition, each model determines the importance of the attributes used for training and discards those that fall below a threshold. Highly correlated attributes as building_footprint and length_footprint or eaves_height and storeys_above_ground are also identified and removed. The results of the training are written to a log file for each hierarchy level. The model files themselves are also saved as *.pkl files. To improve model performance, weights are also introduced for individual attributes per level. The weights are adjusted manually in order to control the importance according to the differentiation based on experience. The results of the changed values are not significant. In addition, the hyperparameter weights are already very well estimated by the software library. As manual adjustment carries the risk of subjective overfitting, we do not

use it. In total, training the models for each hierarchy level takes about an hour on the mentioned office laptop.

5.2 Model Validation

Validation is also performed with the respective random forest model, depending on the hierarchy level. The scikit-learn library offers numerous metrics to check the quality of the models. Reports are generated for each level, showing the values determined. In addition, the performance of all models over the entire process is determined. The validation logs for each hierarchy level are also saved. A visual check of the individual buildings is provided by the automatic transfer of the data to QGIS (Fig. 4). Residential buildings shown in grey serve as training data, correctly validated buildings are shown in green and incorrectly validated buildings are shown in red. All other colours represent the predicted building classes for damage potential.

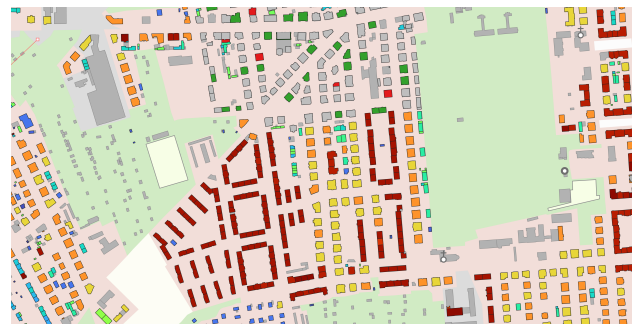


Figure 4. Visualisation of training, validation and classification data sets with QGIS for evaluation of the random forest models.

The results are also summarised in a dashboard (Fig. 5). A key parameter is the accuracy, which is used to calculate the proportion of correctly assigned classes out of the total number of buildings to be validated:

$$\text{accuracy}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} 1\{\hat{y}_i = y_i\} \quad (1)$$

where $1\{\cdot\}$ = indicator function
(1 if condition is true, 0 otherwise)
 n_{samples} = total number of samples
 \hat{y}_i = predicted label for sample i
 y_i = true label for sample i

Other important parameters are the F1 score and the weighted F1 score. Both scores evaluate the overall performance of the respective model. They represent the ratio of correctly assigned classes (true positives) to the sum of correct, incorrectly correct (false positives) and incorrectly incorrect (false negatives) assignments:

$$F_1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \quad (2)$$

where TP = true positives
 FP = false positives
 FN = false negatives

Both parameters represent the harmonic mean of the precision and recall metrics and are therefore robust to unbalanced classes. The weighted F1 score also takes into account the relative frequency of the class in the data set and thus provides realistic values for the multi-class problem used in this project. The model training aims to obtain values above 70 % for good results and above 80 % for very good results. Critical values below 60 % should be avoided in any case. The quality requirements apply to both, the individual levels and the overall classification.

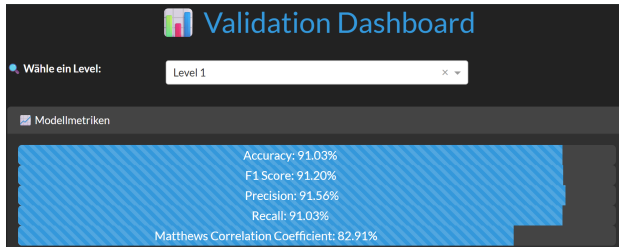


Figure 5. Dashboard of the validation results for Level 1.

The trained random forest models for the nine hierarchy levels achieve the following accuracies and F1 scores during the validation process, as shown in table 3:

| Level | accuracy | F1 score |
|----------------|----------|----------|
| 1 [M/E/Other] | 91 % | 91 % |
| 12 [HH/LW] | 95 % | 95 % |
| 121 [HH] | 100 % | 100 % |
| 122 [LW] | 59 % | 60 % |
| 111 [MR] | 71 % | 71 % |
| 112 [ME] | 74 % | 73 % |
| 113 [ER] | 87 % | 86 % |
| 114 [EE] | 68 % | 68 % |
| 1111 [MRG/MRO] | 84 % | 84 % |

Table 3. Accuracies and F1 scores of each random forest model for the hierarchy levels.

The dashboard also displays the Confusion Matrix (Fig. 6). The matrix shows the number of classes predicted and the number of classes actually identified in a heat map. The diagonal of the matrix shows all correctly predicted data sets.

Confusion Matrix

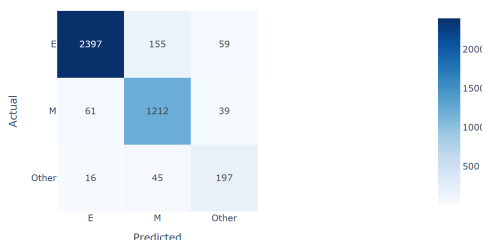


Figure 6. Confusion Matrix of Level 1 with M for apartment houses, E for single-family houses and Other for backyard houses and agriculturally characterised buildings.

To evaluate the overall performance of all models across all hierarchical levels, the correctly and incorrectly assigned classes are also aggregated. This allows the accuracy of the overall classification to be determined. The results of the analysis of the classified residential buildings in Dresden are shown in table 4.

| Metric | Value |
|----------------------|-------|
| True Positives | 5884 |
| True Negatives | 19750 |
| False Positives | 487 |
| False Negatives | 493 |
| End-to-End Accuracy | 81 % |
| Model match | 1962 |
| Direct assignments | 1407 |
| Correctly classified | 3369 |

Table 4. Metric of the whole process over all levels.

It can be seen that misclassifications are less common than correct classifications. The entire classification process across all hierarchy levels achieves an end-to-end accuracy of 81 %. This result is due to direct assignments and model matches.

5.3 Building Classification

The classification of the remaining 70,000 residential buildings in Dresden, which are available in CityGML format, is also based on the hierarchical classification of the individual levels. The random forest models generated in the training process are applied and intermediate results are saved in the 3D city database in the corresponding relation. Buildings for which the additional datasets already provide a value for building age are not classified into the building age group hierarchy levels. These buildings are assigned a known age with a confidence level of 1 for this step. Classifying the buildings through each hierarchy level takes about half an hour on the mentioned office laptop. For each result at each level, the certainty of the classification is also written to new attributes of the relation. A summary evaluation of these provides further information on the performance of each model (Fig. 7).

Confidence Report

Level: 1

```

=====
Target value: M
> 0.9: 2757
0.8 - 0.9: 4912
0.7 - 0.8: 2735
0.6 - 0.7: 1944
0.5 - 0.6: 1621
< 0.5: 1595
Target value: E
> 0.9: 2
0.8 - 0.9: 357
0.7 - 0.8: 665
0.6 - 0.7: 1337
0.5 - 0.6: 1832
< 0.5: 4327
Target value: Other
> 0.9: 29
0.8 - 0.9: 537
0.7 - 0.8: 3521
0.6 - 0.7: 11733
0.5 - 0.6: 19212
< 0.5: 13328
=====

```

Figure 7. Confidence report of Level 1, presenting the total numbers per confidence interval for the predicted building class.

For example, for Level 1 it can be seen that the certainty of the classifications varies greatly depending on the class. For the target values *E* and *Other*, only a few residential buildings can be

classified with a high or very high certainty of more than 0.8. In addition, almost 27 % of the buildings at this level are classified with a certainty of less than 0.5, which is equivalent to a random decision. Overall, only more than 21 % of the buildings are reliably classified (above 0.7). This shows that the classifications still need to be checked by experts and corrected if necessary. To take this into account in the developed plugin, there is a section for manual correction of the classified buildings. In Fig. 8 the procedure for such an adjustment is shown.

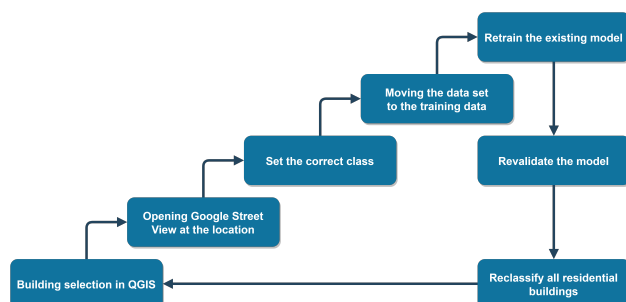


Figure 8. Manual correction of the class of every classified building.

First select the building in question using the QGIS plugin. This will automatically open the Google Street View application in the preferred browser at the location of the building. This requires Google Street View data at the location. The editor can estimate the class of the building based on the images and enter the correct class via the plugin if necessary. The data set is then added to the training data set and the existing models can be retrained. The patterns learned from the original training sessions are used to improve them. This semi-automated process allows the training data set to be significantly expanded in a short time. This is especially important for classes with a very small number of samples. This semi-automatic mapping of additional buildings is intended to increase the quality of the random forest models and thus the reliability of the classification. The results of the application of this process still need to be evaluated in the future. After checking and improving the quality of the classification, the assigned classes are written as a new attribute in the original data set in CityGML format, which takes about half an hour. This adds an attribute to the 3D city model of the city of Dresden and makes it usable for the application of a digital heavy rain twin.

6. Conclusion

This project has used and validated a matrix of residential buildings in the city of Dresden. This answers the first research question:

- How can residential buildings in Dresden be classified according to their construction type and age?

The residential building matrix divides the buildings into 37 classes based on seven construction types and seven construction ages. This differentiation is adapted to the building types found in Dresden. The distinctions result in clear separations for the subsequent damage potential analysis in heavy rain simulations. The buildings are categorised according to their specific characteristics:

- Which characteristics can be used for the classification?

The developed tool offers the possibility to prepare existing open geospatial data for classifications using freely available applications. Using the 3DCityDB schema allows the possibility to work with several dialects of the CityGML standard. The current 3D model of the city of Dresden already provides many semantic features of the buildings. In addition to the given geospatial data sets, geometric attributes, topological relationships between neighbouring buildings and statistical attributes are calculated for each feature vector. Digital city models are filtered by the tool according to the required attributes and saved together with the building geometry in a relation. Note that highly correlated attributes must be excluded during training. In addition, weak attributes can be ignored for the classification. The large number of buildings to be classified requires the automation of the process. This work also answers the following question:

- How can the classification be implemented automatically for all residential buildings in Dresden?

The scikit-learn library provides extensive functions that are used to classify and validate the data. The classification is performed hierarchically. For each hierarchy level a random forest model is trained. Out of 90,000 residential buildings in Dresden, 17,000 have already been mapped and are used as training data. The optimal random forest model and the evaluation of the used attributes could easily be parameterised. The results of the training are validated with another 4,000 already mapped buildings:

- Which reliability of the classification can be achieved?

Various metrics are visualised in a dashboard. The focus is on the accuracy, the F1 score and the weighted F1 score. The overall performance of differently parameterised machine learning models is also categorised on the basis of correct classifications (true positives) and correct non-classifications (true negatives). In addition, the confidence of the classification for each building is saved in the database. The project shows that different data must first be harmonised, as the attribute names and the coordinate reference systems used are different. It is also important to ensure that sufficient training data is available in each class to achieve a reliable and correct classification. An extension of the training data through human intervention is absolutely essential. In addition, the individual classifications must continue to be checked by an expert employee. Finally, the classes determined for each building are added to the original CityGML data. This makes it possible to analyse the damage potential in the event of heavy rainfall. The workflow is customised to the development of the city of Dresden:

- How can the tool be adapted to other cities?

The developed source code can be adapted to other cities if the matrix of residential buildings and thus the hierarchies of the classification are adapted to the respective development. In addition, a 3D city model must be available and the available attributes must be verified. Categorical attributes must be able to be clustered into unique categories. Furthermore, they must be processed with the label encoder in order to be used for random forest models. Numerical attributes can be used directly, but consistent value ranges are required for better training and classification results. The basic procedure and the applications used can then be adopted.

7. Limitations and Outlook

The tool developed for the classification of residential buildings can be used for the city of Dresden. The residential building matrix needs to be adapted for any other type of settlement structure. It should also be noted that the application depends on open geospatial data, such as the 3D city model with fully populated attributes. The different building classes can be found in different frequencies in Dresden. The training data set includes for example less than 400 backyard houses but more than 3,000 detached apartment houses. In addition, the differences between the building types are not always clearly recognisable on the city model. As a result, some random forest models produce unsatisfactory classification results. Due to the poor results for some classes so far, either many manual online corrections have to be made or the in-place mapping has to be extended significantly, especially for rare classes. The implementation as a plugin for the QGIS software offers simple visualisation options, but also restricts the full use of the scikit-learn library. It is not possible to set all possible hyperparameters or use additional model variants. This type of implementation also limits performance. Development as a standalone application is recommended. In the following steps, the classification is improved by manual corrections and the training data set is extended. Once the classification of all residential buildings in Dresden has been completed with good results, the building class attribute is added to the input data in CityGML format. They will then become part of the heavy rain simulations in the Smart City model project and the damage potential can be calculated.

Acknowledgements

DeepL has been used for English grammar checking.

The ‘Environmental monitoring/digital (heavy rain) twin’ project (grant number 14675631) is being carried out in cooperation with the city of Dresden. It is funded under the Smart Cities pilot programme of the German Federal Ministry of Housing, Urban Development and Construction and the Kreditanstalt für Wiederaufbau (KfW).

Gefördert durch:



aufgrund eines Beschlusses
des Deutschen Bundestages



Open source and research data publication

The following link from our Github repository includes the QGIS Python plugin, machine learning algorithms and quality evaluation tools:

https://github.com/dd-bim/Building_classifier

The results of the training, validation and classification are also published on Zenodo (DOI 10.5281/zenodo.15799484). The link provides all the log files and quality reports that form the quantitative basis of this descriptive case study research paper. It also contains a shapefile with the initial results of classifying all residential buildings of the city.

References

- Biljecki, F., Sindram, M., 2017. ESTIMATING BUILDING AGE WITH 3D GIS. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-4/W5, 17–24.
- Golz, S., Schinke, R., Naumann, T., 2014. Assessing the effects of flood resilience technologies on building scale. *Urban Water Journal*, 13.
- Hecht, R., 2014. *Automatische Klassifizierung von Gebäudegrundrissen: ein Beitrag zur kleinräumigen Beschreibung der Siedlungsstruktur*. IÖR-Schriften, Rhombos-Verl, Berlin.
- Kaden, R., Krüger, A., Kolbe, T. H., 2012. Integratives Entscheidungswerkzeug für die ganzheitliche Planung in Städten auf der Basis von semantischen 3D-Stadtmodellen am Beispiel des Energieatlases Berlin.
- Lei, B., Liang, X., Biljecki, F., 2024a. Integrating human perception in 3D city models and urban digital twins. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, X-4/W5-2024, 211–218.
- Lei, B., Liu, P., Milojevic-Dupont, N., Biljecki, F., 2024b. Predicting building characteristics at urban scale using graph neural networks and street-level context. *Computers, Environment and Urban Systems*, 111, 102129.
- Schinke, R., Kaidel, A., Golz, S., Naumann, T., López-Gutiérrez, J., Garvin, S., 2016. Analysing the Effects of Flood-Resilience Technologies in Urban Areas Using a Synthetic Model Approach. *ISPRS International Journal of Geo-Information*, 5(11), 202. <https://www.mdpi.com/2220-9964/5/11/202>.
- Sideris, N., Bardis, G., Voulodimos, A., Miaoulis, G., Ghazanfarpour, D., 2019. Using Random Forests on Real-World City Data for Urban Planning in a Visual Semantic Decision Support System. *Sensors*, 19(10), 2266.
- Tognoli, M., Peronato, G., Kaempf, J. H., 2023. A Machine Learning Model for the Prediction of Building Hourly Heating Demand from CityGML Files: Training Workflow and Deployment as an API.
- Vetter, J. Z., Neuhäuser, S., Rosin, J., Stolz, A., 2024. A Categorization and Parametric Modeling Approach Using Open Geodata Enabling Building Vulnerability Assessment. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, X-4/W5-2024, 309–316.
- Yao, Z., Nagel, C., Kunde, F., Hudra, G., Willkomm, P., Donaubaue, A., Adolphi, T., Kolbe, T. H., 2018. 3DCityDB - a 3D geodatabase solution for the management, analysis, and visualization of semantic 3D city models based on CityGML. *Open Geospatial Data, Software and Standards*, 3(1), 5.