Assessing implicit knowledge in BIM models with machine learning

ABSTRACT

The promise, which comes along with Building Information Models, is that they are information rich, machine readable and represent the insights of multiple building disciplines within single or linked models. However, this knowledge has to be stated explicitly in order to be understood. Trained architects and engineers are able to deduce not explicitly stated information, which is often the core of the transported architectural information. This paper investigates how machine learning approaches allow a computational system to deduce implicit knowledge from a set of BIM models.

INTRODUCTION

The adoption of Building Information Modelling (BIM) constitutes a radical shift in the way models in the building and construction industry are described. Traditional representations for architectural knowledge, such as architectural drawings, 3d models, technical descriptions and spreadsheets are transitioning into semantically-rich information models. Building related information no longer exists within discrete entities, but is kept in an interlinked context. BIM authoring tools for Design and Construction and Facility Management systems provide now semantic information about elements and spaces within buildings, their constituting elements, as their interrelation and performance. Currently huge efforts are put in place to create links between buildings, on local, regional and international level, through e.g. standardisation committees (buildingSMART 2014) or the Geospatial communities (Geospatial Media 2014), but as well through international research project, such as DURAARK (Durable Architectural Knowledge) (DURAARK 2015), a three year EU funded project on the creation and maintenance of semantic links between representations of buildings.

Building Information Models, in formats such as a Revit database or IFC, have become the bond that connects disciplines by streamlining data exchange and connecting the construction with the operational phases of a building lifecycle. Building related knowledge is herein represented in an object oriented way, holding building element geometry, properties, and its interrelation to other objects. These objects can be part of the described building, but in addition relate to external objects or other sources of information, including building element libraries. Information can be related to physical entities, like a wall, as well as to intellectual or organisational constructs, for example spaces or organisations. Hence, the model can support many facets of the construction phase, and in addition guide the building's operation with Facility Management tools or the planning of retrofitting tasks.

The new class of information is directly machine-interpretable, as it conforms to a structured schema. The use of BIM models in current practice is however predominantly focussed on explicit information, such as property values, augmented with aggregate functions for the extraction of quantity information and clash detection based on geometrical inference (Tamke et al. 2014a). BIM models hold however information that is not explicitly stated, but lies implicit in the interrelation between the entities within a single model or in the interrelation of a large variety of models. And while years of practice train a building professional to immediately apprehend the functions of a space by means of merely symbolic two-dimensional representations, this information can currently not be assessed by machines. We ask, how these implicit second order descriptors can be assessed and whether this approach holds the potential to describe the qualitative aspects of a building.

STATE OF THE ART

This paper presents experimental approaches directed at extracting implicit data from building models. It shares its interest with parallel research into the assessment of architectural models. This is for instance concerned with parametric models (Davis 2014), the way these are set up and the complexity they address. The descriptors for this, such as dimensionality or cyclomatic complexity, stem from Computer Science and are produced by algorithms reading the parametric models. Stasiuk and Thomsen (2014) investigate open-ended processes of discovery and categorical description of form-found design models. Rather than looking at the properties that constrain the form-finding process, the study uses machine learning algorithms, such as k-means clustering, to categorize the models by means of emerging properties. Machine learning is particular interesting to the here described approach, as it is able to address the great variety that it is inherent in architectural designs and identify similarities rather than conformity.

Approaches for explicitly querying building models can be qualified based on their intended use, their underlying technology and expressiveness of the query language, their amount of abstraction from underlying data and various other measures. A language like BimQL (Mazairac and Beetz 2013) provides means to extract entity instances from IFC models using the entity and attribute names defined in the IFC EXPRESS schema, as such it is an effective tool for people to extract explicit information. Other methods are developed for the purpose of conformance checking. Examples of this include Solibri Model Checker (Solibri 2014) which checks models for conformance to BIM standards or clashes between elements using hardwired constraints, or the mvdXML checker (Zhang et al. 2015), which checks for the conformance of a certain model to a Model View Definition, a construct that imposes additional constraints for validity onto an IFC file in addition to the constraints as dictated by the schema. The expressiveness of the mvdXML language however does not include querying for implicit relations between different instances, for example the spacing between two columns, as it has no notion of binary operators. Other approaches express topological relations, such as containment or adjacency, and implement these in query languages (Daum and Borrmann 2014). Such an approach enables to query for binary relational aspects of the model. Within the DURAARK project, metadata extraction utilities are provided (Beetz et al. 2014) to automatically extract literal values from IFC files according to a metadata schema. This has been extended to determine Level of Development information by assessing calculated attributes based on geometric detail (Tamke et al. 2014b).

APPLICATIONS OF MACHINE LEARNING

The querying approaches presented above have in common that they are tailored to specific scenarios with predefined outcomes as they allow for no novelty or discovery. Conversely, the promise of machine learning is the extent to which it is able to make predictions and detect patterns. In order to make such predictions and find such patterns, typically, metrics about actual buildings would be collected, such as the measured energy efficiency. These real-world metrics can then serve as a label to the set of building models and, hence, patterns can be unravelled between the building configurations described in the models and their measured performance.

In this paper both supervised and unsupervised machine learning (Mitchell 1997) of BIM models will be discussed. An unsupervised learning approach will be presented to show anomalies in building models. Unsupervised approaches work without the premise of aforementioned labels. More speculatively, it can be seen as a method to reduce the failure costs in the construction industry by flagging uncommon situations that might need additional checks or coordination. These anomalies could include unusual large overhangs or other situations that constitute an unusual confluence of several building elements.

In addition, a supervised machine learning approach based on a neural network is discussed. It is able to classify floor plans according to its intended function. Such a system can be seen in the light of a large archival framework for building models in which information pertaining to intended function and use is often fragmented and incomplete. This paper argues that supervised learning can be used to complete missing attributes in such a dataset. Such an approach can be extended to classify based on other criteria, such as iconic and exemplary architectural edifices, based on geometrical descriptors. Both these attributes are common in the world of archival, but seldom explicitly documented for newly built artefacts.

Both supervised as well as unsupervised learning algorithms typically assess a sample, in this case a building, by means of a set of features that describe the building. How such features can be extracted automatically from building models is described in the next section.

IMPLEMENTATION

The IFC Machine Learning platform presented in this paper is built on top of the DURAARK IFC metadata extractor (Beetz et al. 2014) (Krijnen 2015). This utility is able to extract literal values, aggregates and derived values from IFC SPF files. The extraction utility presents a simplified Domain Specific Language (DSL) that enables users with little programming experience to map query paths pointing to literal values in the IFC file to keys in a metadata schema. Furthermore, the DSL provides functions to compute aggregates, such as to count the extent of a list or sum or concatenate attribute values. The DSL is a subset of the Python scripting language and is in fact executed as a regular Python program. In addition, serialization formats are provides to output the data in a format suitable for further processing. For example, in the context of the DURAARK workbench, the extracted values are written into a linked data RDF graph. This version of the extractor has been published under an open source license (Krijnen 2015a). For the Machine Learning platform, an additional output format has been added to output Comma Separated Value files.

What follows now are short excerpts of the data extraction system to highlight some key aspects of its use. For example, to query the unique identifiers of all wall elements one could invoke the following statement:

```
csv_formatter << [
        file.IfcWall.GlobalId >> "ifc_ml:wall_identifiers"
]
```

Listing 1: Code required to output wall identifiers

```
   wall_identifiers
0  2O2Fr$t4X7Zf8NOew3FL9r
1  2O2Fr$t4X7Zf8NOew3FLIE
2  2O2Fr$t4X7Zf8NOew3FLPP
   ...
```

Listing 2: Output from the program provided in Listing 1

```
csv_formatter << [
        file.IfcWall >> count >> "ifc_ml:wall_count"
]
```

Listing 3: Code required to output the number of walls in an IFC file

```
wall_count
57
```

Listing 4: Output from the program provided in Listing 3

The hypothesis in this paper is that geometrical properties of the building elements and their relations can be used to build an implicit architectural knowledge model that can be assessed with Machine Learning. Therefore, in order to extract geometrical quantities that are derived from the body representation of IfcProducts, the DSL has been extended with geometrical operators. In addition, to calculate the minimal distance between these products, the concept of binary functions is introduced that signify a relationship with other products in the file. The geometrical and topological analysis functionality is implemented on top of IfcOpenShell (Krijnen 2015b) and pythonOCC (Paviot 2014).

```
elems = file.IfcProduct
    >> segment(by_entity)
    >> segment(by_attribute("GlobalId"))

csv_formatter << [
       elems >> shape_area >> "ifc_ml:area",
       elems >> shape_volume >> "ifc_ml:volume",
       elems >> shape_gyradius >> "ifc_ml:gyradius"
]
```

Listing 5: A more elaborate example that lists geometrical attributes for all building elements

segment_0	segment_1	**area**	**gyradius**	**volume**
IfcBeam	2OrWItJ6z...	9.17	1.78	0.04
IfcSlab	1CZILmCaH...	53.89	2.10	3.81
IfcSpace	0BTBFw6f9...	67.50	1.80	33.51
IfcWallSt...	0dxE1Sy6n...	23.64	1.20	1.64
...	...	...	...	...

Listing 6: Output from the program provided in Listing 5

UNSUPERVISED LEARNING: ANOMALY DETECTION

As an unsupervised method, outlier detection is applied to the geometrical attributes of the elements in a model. Outliers are the samples that deviate from an observed median area. In this experiment the duplex apartment model is used (NIBS 2013). The building element samples are segmented according to their entity type (wall, window, and so on). For each of these element types, an elliptical envelope is fit through the sample data. For this purpose the scikit-learn toolkit (Pedregosa et al. 2011) has been used. For demonstration purposes, a two-dimensional plot of the data is shown below, where the dimensionality of the data is reduced by using the ratio between the geometrical attributes. The plot highlights that are is a clear centre in which most of the samples reside and that there are clear outliers outside of this centre area. The samples are coloured according to their Mahalanobis distance to the centre. This distance metric accounts for the distribution of the data, contrary to a Euclidian distance would and is a common measure for classification purposes. Contours of the elliptical boundary are shown in dashed lines.

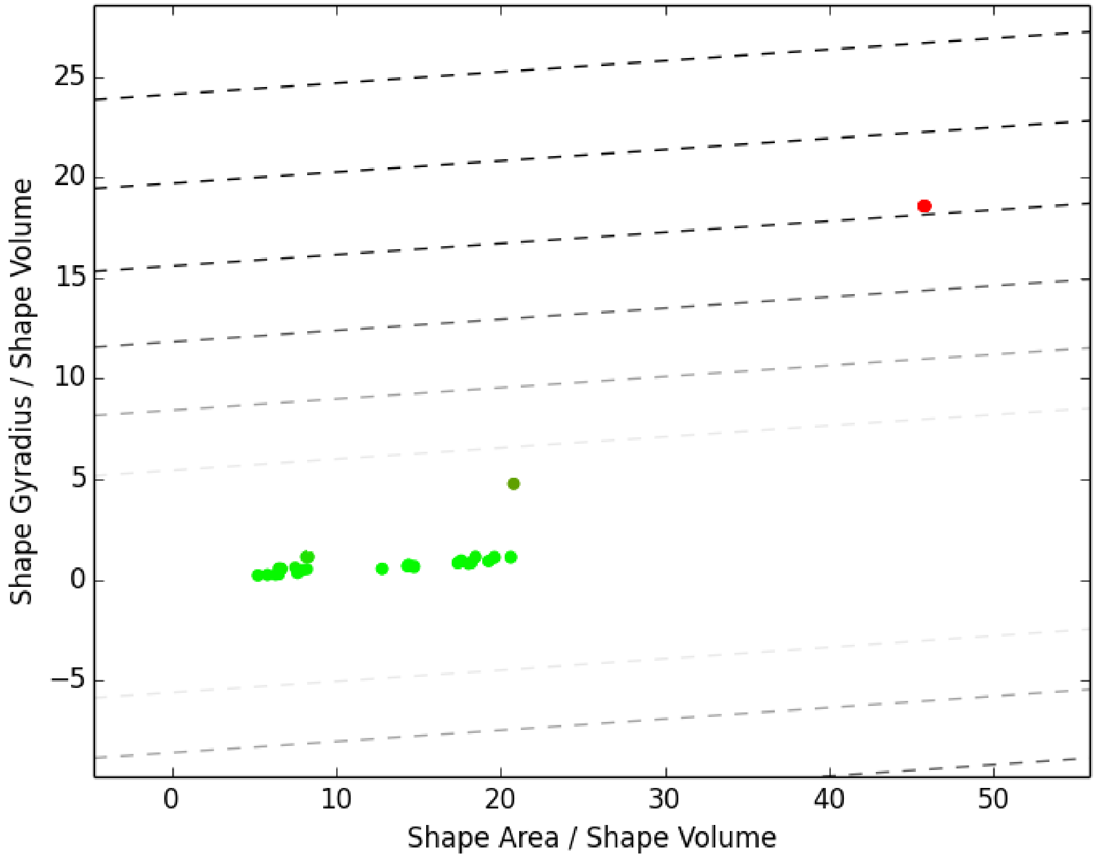

Figure 1: IfcWallStandardCase anomalies in the Duplex Apartment model according to their geometrical attributes

	Mahalanobis distance
0iEHWY1$XA8eQeeULq4jpl	2.00135121249
0jf0rYHfX3RAB3bSIRjmpw	87.4188172025
1aj$VJZFn2TxepZUBcKphf	2009.27365845
3Y4YRln2r91vflHcHE5ITm	38142.0133829
...	...

Listing 7: An excerpt of the wall GlobalIds with their Mahalanobis distance

Visually, the same information can be represented by colour coding the model elements, as can be seen in Figure 2. In this overview it can be assessed that the ten elements with the highest distances are in fact miscategorised to be wall elements. If one inspects the definition of a wall in the IFC schema one can see that these elements do not fulfil a role in bounding or subdividing the construction work. In fact, they appear to be narrow and horizontal structural members and therefore could have been more suitably classified as a beam.

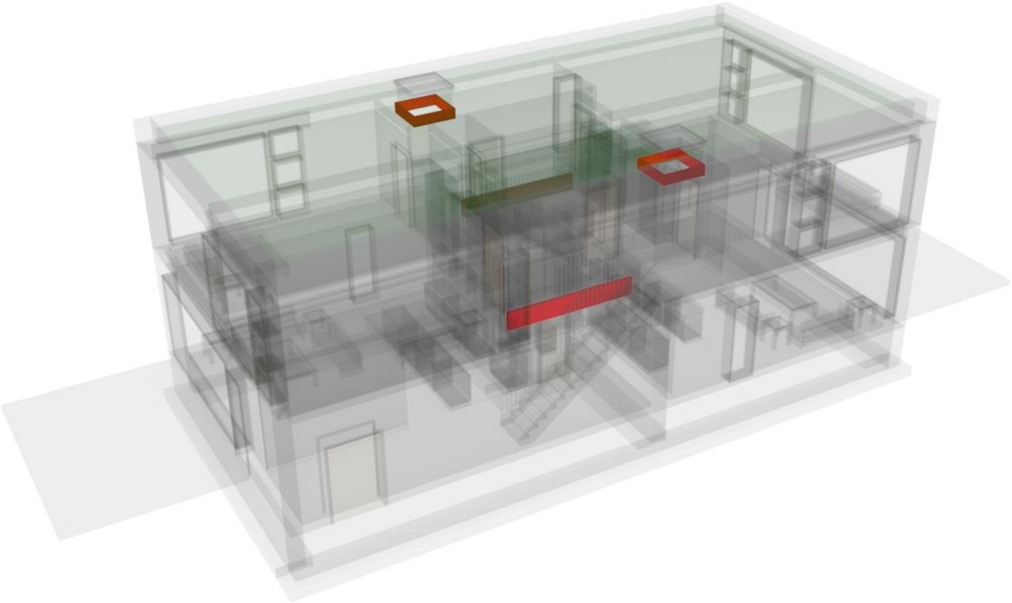

Figure 2: Ten elements in the duplex model that are misclassified as wall found by anomaly detection

To summarize, by looking at a single model, the algorithm has been able to identify the geometrical essence of a wall and found elements that deviate from this idea. It seems that these are likely misclassified elements. In building projects misclassifications can be problematic as they might result in elements escaping from being verified the appropriate person, in this case perhaps by a structural engineer, as he might inspect only a subset of the model corresponding to a structural view, to which architectural wall elements do not belong.

A more precise classification of anomalies can be obtained by training the algorithm with a dataset that is known to be correct. In this way outliers do not contaminate the dataset and, therefore, a more precise decision boundary can be obtained. Furthermore, other algorithms can be applied that pose fewer restrictions on the distribution and modality of the data. The elliptical envelope method, which has been used in this example, works best on Gaussian, symmetric and unimodal distributions of the feature vectors.

It is assumed that this line of reasoning can be extended to more sophisticated discoveries if relational parameters to other nearby products are introduced. In such a way, clearance areas, for example which can be found near stairs or doors, can be assessed. Or the load-bearing characteristics of a column can be schematized as an element that directly connects to an element to the top and to the bottom of it. For these advanced relational concepts the definition of orientation of elements is crucial. These experiments are to be done in follow-up research.

SUPERVISED LEARNING: NEURAL NETWORKS

BIM models are able to represent many facets of a building, in addition to geometrical and relational information, for example by using predefined and extensible property set or reference to external data sources. However, datasets from practice (DURAARK 2015) show that BIM models typically contain this information only partly and have heterogeneous levels of information. Metadata records that might exist for one building might be absent for the other. As such, it is worthwhile to investigate whether Machine Learning can act as a means to supplement or complete this information. In this experiment, individual building elements are no longer subject of examination, but instead building storeys, a common aggregate structure in building models are assessed in order to classify them according to their function and intended use. This is metadata that is typically not semantically available in an IFC building model, but is very relevant in a digital archive. By looking at the spatial configuration of a floor plan, a neural network is trained to differentiate between residential and institutional facilities.

The dataset consists of publicly available models augmented with models that have been aggregated over the course of the DURAARK project. An overview of the models, separated by floor and divided by their residential or non-residential labels is provided in Figure 3.

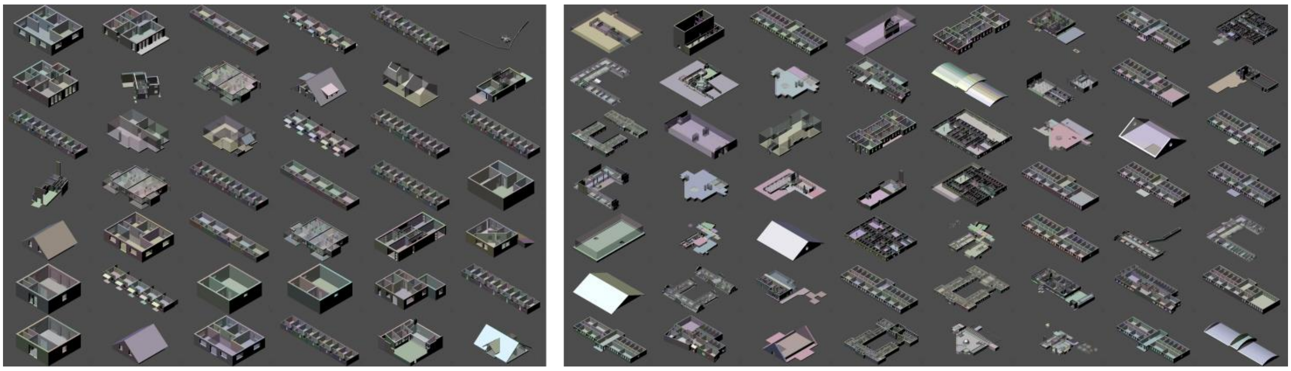

Figure 3 Residential (left) and Non-residential (right) building storeys in the dataset

The geometrical attributes that are used to assess the floor plans are provided in Listing 8. As a measure for the compactness of a space, the radius of gyration (gyradius) is used. Alternatives to this metric include for example the ratio between surface area and bounding volume (Corney et al. 2002), but it is assumed that for this purpose a high gyradius specifically signifies long branching corridors. To be precise, in this specific case, the gyradius refers to the radius around the vertical axis through the centre of mass of the solid volume. It is a measure for the distribution of mass around this vertical axis. This implies that for a shape that describes a network of corridors this measure will be higher than for a cylinder, even if they bear the same volume, the latter being the most compact 2.5D shape. An overview of the gyration radius and centre of mass for two distinct solid shapes can be seen in Figure 4.

Space - wall volume ratio	Measures the amount of spatial segregation
Space - slab volume ratio	Measures the amount of spatial segregation in vertical direction
Doors per space	Measures the connectivity of spaces
Average space volume	Measures the size of the spaces
Space volume variance	Measures the extent to which spaces vary in size
Average space gyradius	Measures the compactness of the spaces
Space gyradius variance	Measures the extent to which the compactness of the spaces varies
Column wall ratio	Measures whether walls or columns are used for load bearing

Listing 8: Features incorporated in the floor plan assessment

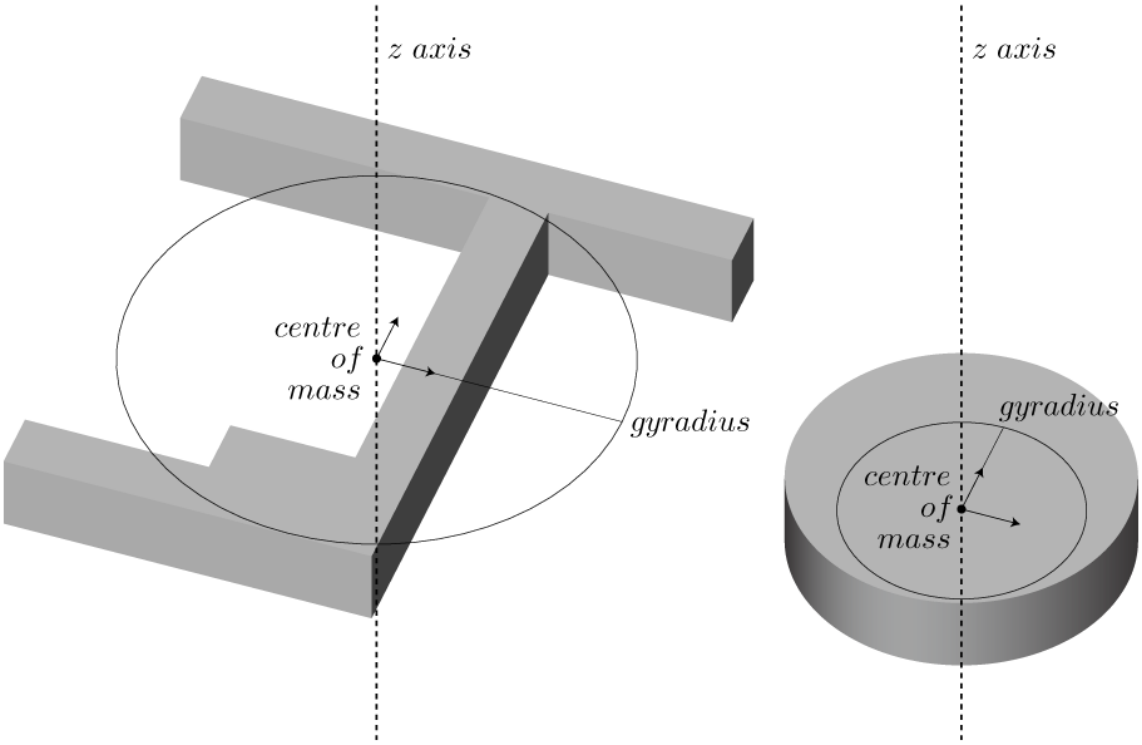

Figure 4 Two distinct solids of identical volume with their centre of mass and gyradius

During the training phase of a neural network a (locally) optimal configuration of connections between neurons will be formed that pertains to the situation at hand. This is implemented by means of a gradient-descent based optimization algorithm, called back-propagation. Such an algorithm functions best on normalized features so that the range of values for different features roughly corresponds. Thus, the geometrical attribute values are normalized and scaled so that they all cover, more or less, the same range and their median values are close to zero. An example of two features before and after feature scaling is provided in Figure 5. Also, notice how these two features already exhibit some grouping in terms of the labels, mostly the variance in space volumes appears to be much higher for the non-residential facilities.

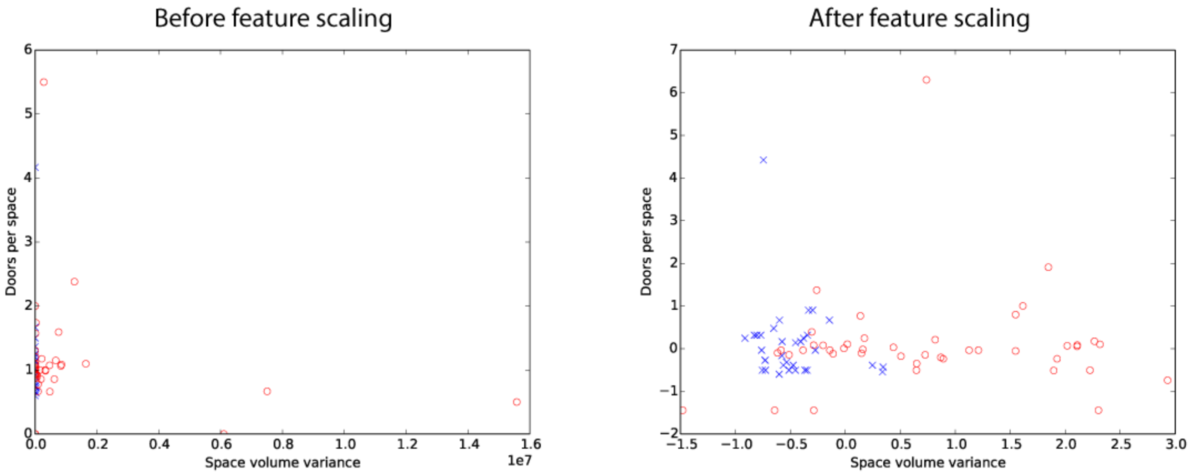

Figure 5 Two features before and after scaling

According to best practices, the labelled dataset is divided into three distinct subsets: a training set (70%), a cross-validation set (20%) and a test set (10%). The training set acts as the input for the back-propagation algorithm and determines the weights of the network edges between the neurons. In order to pick a network that pertains to a general solution and not just specifically to the training set input, the optimal network is selected based on its misclassification error on the cross-validation set, which has not been used to build the network. In this particular case a network with bias units and without hidden layers proved to result in the lowest misclassification error on the cross-validation set. Both these concepts in general make the network more capable to fit complicated problems, as the number of neurons increases. The network has been trained for 4000 iterations. The remaining final 10% of the models provide an indication of the actual performance of the network on unseen cases. The models are provided in Figure 6. According to the manual labelling process there is a single model that is misclassified, it is a non-residential floor layout categorized as a residential floor plan. In defence of the algorithm, one might argue that it is a difficult case, as it is the cellar of an historic building for which there are no other samples in the training set. The neural network has been implemented in the Python programming language using PyBrain (Schaul et al. 2010).

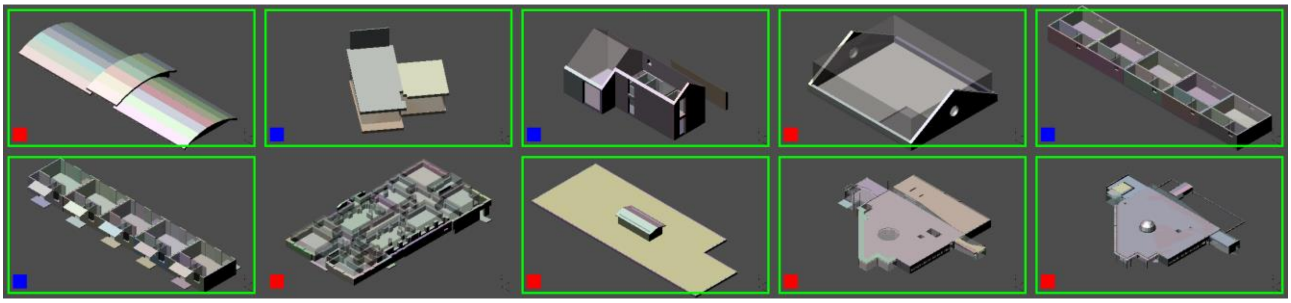

Figure 6 Results on the test set with all successful classifications marked in green

To conclude, it has been shown that a neural network has been able to differentiate floor plan layouts into two categories based on their geometrical appearance. The authors assume that such classification can be augmented with metrics from spatial connectivity graphs, including aspects like centrality and clustering coefficients. With a bigger dataset and more features, future research will indicate whether it is possible to distinguish between more categories of building functions. In this example, the neural network has been formed to some extent by trial and error, which is not uncommon in this domain. A more systematic and exhaustive application is of neural networks is beyond the scope of this paper and will have to be extended upon in future research. From both machine learning examples in this paper one can draw that the geometrical nature of the elements that constitute a BIM model provides insight into the nature of these elements on their own and into the assembly they form as a whole.

CONCLUSION

The paper demonstrates the application of both supervised and unsupervised machine learning methods with Building Information Models. The implicit attributes and qualifications can currently not be found by means of traditional computational approaches in architectural practice. Results from machine learning on architectural datasets provide a relevant alternative view to explicit querying mechanisms and provides useful insights for more informed decisions in the design and management of buildings.

This paper presents initial research into the use of machine learning to create architectural insights. The feasibility of the approach was proven on rather basic examples. Future research will have to indicate how this can be extended and generalized into other areas for instance for the more complete quality assurance of BIM models. The search for misclassified elements is here a starting point into further research, which might even address the assessment and classification of designs and architectural qualities.

In particular relational characteristics on the level of the building element, such as clearance areas and typical confluences of specific element types, might yield a richer comprehension of implicit knowledge and would further exploit the relational nature of an IFC file. On the level of building stories, graph measures relating to spatial connectivity might be vital to develop a more detailed understanding on the exact spatial configuration and therefore a more precise functional classification.

The computational approaches, which are currently used in practice to assess BIM models, build upon explicit rules (Solibri 2015) built by experts. Machine learning might enable less technical users to query complex BIM datasets for highly practice and project specific insights.

REFERENCES

Beetz, J. et al., 2014. D3.3 Semantic Digital Archive Prototype. [Online]
Available at: http://duraark.eu/wp-content/uploads/2014/08/DURAARK_D3_3_3.pdf

buildingSMART, 2014. buildingSMART. [Online]
Available at: http://www.buildingsmart.org/

Corney, J. et al., 2002. Coarse filters for shape matching. IEEE Computer Graphics and Applications 22 (3), pp. 65-74.

Daum, S.; Borrmann, A., 2014. Processing of Topological BIM Queries using Boundary Representation Based Methods. Advanced Engineering Informatics 28 (4), pp. 272-286.

Davis, D., 2014. Quantitatively Analysing Parametric Models. International Journal of Architectural Computing 12 (3), pp. 307-320.

DURAARK, 2015. DURAARK Project. [Online]
Available at: http://duraark.eu/ [Accessed May 2015].

Geospatial Media & Communications, 2014. GeoBim. [Online]
Available at: http://geo-bim.org/ [Accessed May 2015].

Krijnen, T., 2015a. DURAARK/pyIfcExtract. [Online]
Available at: https://github.com/DURAARK/pyIfcExtract [Accessed May 2015].

Krijnen, T., 2015b. IfcOpenShell. [Online]
Available at: https://ifcopenshell.org [Accessed May 2015].

Mazairac, W.; Beetz, J., 2013. BIMQL – An open query language for building information models. Advanced Engineering Informatics 27 (4), pp. 444-456.

Mitchell, T., 1997. Machine Learning. New York City: McGraw Hill.

NIBS, 2013. buildingSMART alliance Common Building Information Model Files and Tools - National Institute of Building Sciences. [Online]
Available at: http://www.pythonocc.org/ [Accessed May 2015].

Paviot, T., 2014. pythonOCC, 3D CAD/CAE/PLM development framework for the Python programming language. [Online]
Available at: https://github.com/DURAARK/pyIfcExtract [Accessed May 2015].

Pedregosa, F. et al., 2011. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research 12, pp. 2825-2830.

Schaul, T. et al., 2010. PyBrain. Journal of Machine Learning Research 11, pp. 743-746.

Solibri, 2015. Solibri Model Checker. [Online]
Available at: http://www.solibri.com/products/solibri-model-checker/ [Accessed May 2015].

Stasiuk, D.; Thomsen, M. R., 2014. Learning to be a Vault - Implementing learning strategies for design exploration in inter-scalar systems. Newcastle upon Tyne, England, UK, sn, pp. 381-390.

Tamke, M. et al., 2014a. D7.7.1 - Current state of 3D object processing in architectural research and practice. [Online]
Available at: http://duraark.eu/wp-content/uploads/2014/06/duraark_d7.7.1.pdf

Tamke, M. et al., 2014b. Building Information Deduced - State and potentials for Information query in Building Information Modelling. Newcastle upon Tyne, England, UK, sn, pp. 375-384.

Zhang, C.; Beetz, J.; Weise, M., 2014. Interoperable validation for IFC building models using open standards. ITcon Vol. 20, Special Issue ECPPM 2014 - 10th European Conference on Product and Process Modelling, pp. 24-39.