

Employing outlier and novelty detection for checking the integrity of BIM to IFC entity associations

Bonsang Koo^a, Byungjin Shin^a, Thomas F. Krijnen^b

^aDepartment of Civil Engineering, Seoul National University of Science and Technology, South Korea

^bDepartment of the Built Environment, Eindhoven University of Technology, The Netherlands

E-mail: bonsang@seoultech.ac.kr, byungjin0826@seoultech.ac.kr, t.f.krijnen@tue.nl

Abstract – Although Industry Foundation Classes (IFC) provide standards for exchanging Building Information Modeling (BIM) data, authoring tools still require manual mapping between BIM entities and IFC classes. This leads to errors and omissions, which results in corrupted data exchanges that are unreliable and compromise the interoperability of BIM models. This research explored the use of two machine learning techniques for identifying anomalies, namely outlier and novelty detection to determine the integrity of IFC classes to BIM entity mappings. Both approaches were tested on three BIM models, to test their accuracy in identifying misclassifications. Results showed that outlier detection, which uses Mahalanobis distances, had difficulties when several types of dissimilar elements existed in a single IFC class and conversely was not applicable for IFC classes with insufficient number of elements. Novelty detection, using one-class SVM, was trained *a priori* on elements with dissimilar geometry. By creating multiple inlier boundaries, novelty detection resolved the limitations encountered in the former approach, and consequently performed better in identifying outliers correctly.

Keywords –

BIM; IFC; Outlier detection; Novelty detection

1 Introduction

Models based on the Building Information Modeling (BIM) paradigm are increasingly being used for multiple applications, including clash detection, building code compliance, design quality assurance, constructability analysis, design and construction coordination. Many of these applications require specialized software, which in turn require BIM models to be exported and exchanged between them.

The Industry Foundation Classes (IFC) plays a pivotal role in enabling interoperability, allowing entity and relationship data to be exchanged seamlessly between applications. However, the IFCs does not necessarily guarantee that the integrity of the data is

maintained. For example, major BIM authoring tools, although abide by the Model View Definitions (MVD) in exporting data, still allows individual components and relationships to be mapped to IFC classes manually, and are thus susceptible to human errors and omissions.

Moreover, BIM models are becoming larger and more complex. Depending on the Level of Development (LOD), the number of elements in a single model can range from 1,000's to 10,000 components. Manually checking the integrity of IFC entity and relationship mapping can quickly become intractable, as the size and complexity of the models increases.

This research addressed this issue by applying machine learning techniques to identify errors or omissions in the data integrity of IFC models. Specifically, using geometric features, anomaly detection techniques are applied to determine whether BIM components have been properly mapped to their correct IFC classifications.

The research used existing work performed by from [1] as its initial point of departure. [1] proposed using 'outlier detection' to check the correct classification of individual elements. To verify the scalability of this approach, we first performed outlier detection on two BIM models with increasing components, from which we identified specific limitations. Subsequently, we explored an alternative anomaly detection approach, namely, 'novelty detection' which proved to be more effective in identifying potential misclassifications.

The research conducted herein is anticipated to enhance the robustness of IFC usage, and contribute to the proliferation of machine learning techniques in the AEC domain, including areas of quality control and regulation compliance.

The rest of the paper is structured as follows. Section 2 provides background on the state of IFC development and the need for IFC integrity checking. Section 3 provides an overview of the research methodology, while Section 4 describes the results and limitations of using outlier detection on two architectural BIM models. Section 5 describes how novelty detection was incorporated, and tested on the same architectural models.

Section 6 compares the results and discusses the implications of the research.

2 Motivating Background

2.1 The need for IFC integrity checking

Major BIM authoring tools provide functionalities to selectively map BIM elements to their corresponding IFC classes. These tools offer default settings that have prespecified mappings of the most common elements to their corresponding IFC classes. However, such is not the case for more obscure IFC classes, which requires manual settings, which may lead to errors and omissions.

Also, many authoring tools allow the use of generic libraries, which do not pertain to a specific IFC class. For example, Revit allows the creation of custom families or model-in-place components, which does not map to specific IFC classes, unless specified otherwise by the model developer.

Furthermore, authors of BIM models may not care for the strict designation of IFC classes, or even the BIM elements themselves, and loosely use families of their choosing or even customized ones.

Such misclassifications, whereas trivial in small BIM models, may become difficult to detect manually once BIM models become large and with detailed level of developments.

In a collaboration environment, where project stakeholders individually develop BIM models, such misclassifications can cause severe interoperability issues between them.

The next section briefly introduces tools and standards that have been developed to ensure IFC model integrity checking, followed by a summary of their limitations.

2.2 Existing approaches for IFC integrity checking

Although IFCs provide a standardized format to share BIM information, the complexity in its schema often requires a domain expert experienced in STEP and EXPRESS to verify its integrity. Practitioners without such knowledge find it difficult to readily employ IFC based models for everyday use [2]. Consequently, several advancements have been made that supports the checking of IFC file formats and support users in ensuring their integrity.

2.2.1 Tools for checking the integrity of IFC data structures

buildingSMART International (bsI), the main

organization that manages and develops IFC standards, provides the Information Delivery Manual (IDM) and Model View Definition (MVD) that allow processes to be formalized and generate subsets of IFC entities and relationships. MVDs such as the Coordination View and COBie are used extensively in the industry.

bsI also provides an 'ifcDoc' tool to check the consistent and computer-interpretable definition of MVDs as legitimate subsets of the IFC specification with enhanced definition of concepts.

[3] developed an 'mvdXML Checker' to evaluate the integrity of IFC files, while the commercial software 'Solibri Model Checker' is used widely in the industry to check the conformity of IFC class and entity data between BIM authoring tools. The National Institute of Standards and Technology (NIST) provides the 'IFC File Analyzer,' which enables a semi-automated approach to verify IFC classes and relationships in an IFC-SPF file.

2.2.2 Development of BIM Query Languages

The complexity of the IFC format has also created the need for BIM Query Languages. These languages allow the development of SQL based statements using {SELECT, FROM, WHERE} command constructs to query BIM models and IFC-SPF files. Initially, query languages based on EXPRESS and EQL were explored Tauscher et al., 2016. Later, BIMQL [4] and QL4BIM [5] were developed, which were customized exclusively for BIM/IFC models, as well as general purpose query languages using ifcOWL and SPARQL [6]. Such advances allowed specific querying of IFC-SPF files, which could be utilized to check for their integrity.

2.2.3 BIM Modeling Standards

A more macro-level approach has been where AEC institutions have provided standards and guidelines for working with BIM models and IFC formats. Namely, the American Institute of Architects (AIA)'s provides the 'Documents E203 and G202: Building Information Modeling Protocol Exhibit' [7], while UK's Construction Industry Council (CIC) provides the 'BIM Protocol.' Similar attempts have been developed for Korea in the form of 'KBIMS'¹.

Such standards provide guidance in standard work breakdown structures, the Level of Development (LOD) of BIM models, libraries templates, and project management matrices to ensure that interoperability is maintained throughout the project life cycle and between project stakeholders sharing multiple BIM models. These guidelines can assist in ensuring that BIM/IFC models are correctly mapped to ensure their integrity.

¹ Korea Building Information Modeling Standards

2.3 Limitations of existing approaches

The variety of tools for evaluating IFC entities and relationships improves the deciphering of IFC data, but it is still by and large a manual process. Similarly, BIM query languages can improve the checking ability but essentially has not been developed for ensuring IFC integrity. The standards and guidelines encourage the correct associations and mapping of elements through best practices, but do not necessarily guarantee them. Thus, there still exists a need for tools or methods directly dedicated to IFC integrity checking, and is the main objective of this research.

3 Research Methodology

This research explored two approaches for identifying misclassifications of BIM elements to IFC classes using anomaly detection techniques: outlier detection and novelty detection.

The first approach is adopted from [1]. Krijnen and Tamke, here after referred as “Krijnen’s approach”, explored the use geometric features of individual components to detect misclassifications using outlier detection. However, at least in the paper, the approach is not fully validated in terms of its scalability to generic BIM models. They only provide an example implementation to one IFC class (i.e., walls) within a single BIM model. Thus, the first step in our research involved identifying potential limitations in their approach by applying their implementation to multiple BIM models with larger number of BIM elements and IFC classes.

Based on these results, a second approach was devised, which primarily used novelty detection as an alternative to address the limitations found in Krijnen’s approach. We selected novelty detection as it allows multiple boundaries for inliers, whereas outlier detection typically is limited to a single inlier boundary. Also, while Krijnen’s approach limited the analysis to elements within a single BIM, the novelty detection approach was implemented to learn features from multiple BIM models.

4 Verification of Krijnen’s approach using outlier detection

This section provides an overview of Krijnen’s approach, and two cases of its applications to BIM models. Results of the case studies were used to identify limitations of the approach and provide the basis for implementing novelty detection.

4.1 Overview of Krijnen’s approach

As shown in Figure 1, given a single BIM model,

outlier detection is performed on individual IFC classes (i.e., `ifcWallStandardCase`, `ifcWindow`, `ifcDoor`, etc.) to detect potential model components that are geometrically dissimilar to other components of the same class. The geometries are used as features for the analysis, and include the area, volume, radius of gyration, orientation from top and bottom of the individual components. The premise is such that elements of the same IFC class should have similar geometric features, and thus a misclassified component will stand out and be detected as an outlier.

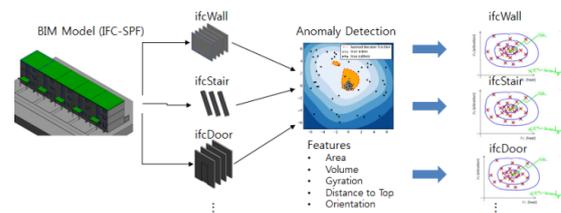


Figure 1. Process overview for outlier detection

[1] developed and applied a suite of open source Python packages to implement their approach. Specifically, `ifcOpenShell` [8] allows the manipulation and query of IFC entities directly from IFC-SPF files. `PythonOCC` [9] is also used to extract individual geometric features needed for outlier detection. Finally, the packages from `scikit-learn` toolkit are used for implementation of the outlier detection algorithm [10].

[1] provides an example by applying their implementation on the wall elements (i.e., `ifcWallStandardCase`) of a duplex apartment BIM model [10]. Their approach identified several elements which are classified as walls but should have been classified as a beam or opening. The results are visually demonstrated using a contour plot (i.e., elliptical envelope) and highlighting the misclassified elements (i.e., outliers) in the BIM model.

The outlier detection used in Krijnen’s approach calculates the Mahalanobis distance of the geometry features to detect outliers in individual IFC classes.

The Mahalanobis distance is widely used in lieu of Euclidean distance in identifying outliers for multivariate datasets [11]. A problem with multivariate data is the effect of covariance between the variables, which cannot be resolved using the Euclidean distance. The Mahalanobis distance overcomes the problem by calculating and using the eigen vectors to transform the main axes of the variables, in effect negating their correlations.

The Mahalanobis distance is calculated using the following equation,

$$D(\vec{x}) = \sqrt{(\vec{x} - \vec{\mu})S^{-1}(\vec{x} - \vec{\mu})} \quad (1)$$

where, $\vec{x} = (x_1, x_2, x_3, \dots, x_n)^T$ is the vector of observed data; $\vec{\mu} = (\mu_1, \mu_2, \mu_3, \dots, \mu_n)^T$ is the vector of average of the observed data, and S is the covariance matrix.

The calculated distance D can be visualized using an Elliptical Envelope as shown in Figure 2. The blue dashed ellipses represent statistically equivalent contours (i.e., boundaries), based on which outliers and inliers are distinguished.

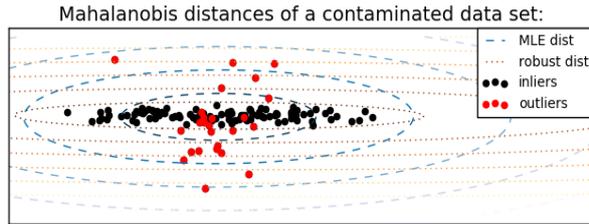


Figure 2. Output diagram using the Elliptical Envelope method for anomaly detection

4.2 Case Studies

Krijnen's approach was applied to two BIM models and their results are described as follows.

4.2.1 Case study 1: Duplex apartment model

The first model is the duplex apartment model, which was used in Krijnen's initial work. The BIM model has 159 elements (i.e., subtypes of IfcBuildingElement), which include beams, coverings, walls slab and roof. Table 1 details the results of performing the outlier detection for the individual IFC classes. Of the 159 elements, 9 outliers were detected in the walls (ifcWall, ifcWallStandardCase), slab (ifcSlab), window (ifcWindow) and doors (ifcDoor).

The following section describes the individual results for each of these classes.

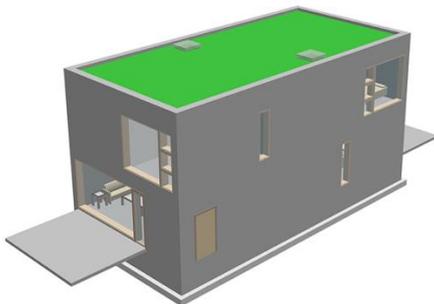


Figure 3. The duplex apartment BIM model

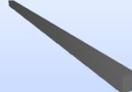
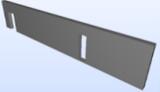
Table 1. Results of outlier detection for the duplex apartment BIM model

IFC Class	# of totals	# of inliers	# of outliers	Analysis of Results
ifcBeam	8	8	0	-
ifcCovering	13	13	0	-
ifcWall (incl. ifcWallStandardCase)	57	54	3	1 outlier represents misclassification, in which, a beam is misclassified as ifcWall 2 outliers are walls, but have openings
ifcSlab	21	18	3	1 outlier has different geometry than inliers 2 outliers are slabs, but have openings
ifcRoof	1	1	0	-
ifcFooting	7	7	0	-
ifcWindow	24	23	1	1 outlier has different height to inliers
ifcDoor	16	14	2	2 outliers have different width than inliers
ifcStair	4	4	0	-
ifcRailing	4	4	0	-
ifcMember	4	4	0	-
Total	159	150	9	

4.2.2 ifcWall, ifcWallStandardCase

The outlier detection identified three entities from 57 wall elements. Table 2 shows the samples of the BIM elements, with their corresponding Mahalanobis distance values. One of these elements was identified as a beam, as shown in Table 2. Thus, this demonstrates a successful detection of a misclassified element. However, the other two elements are walls with openings. These elements are walls, and the approach has identified them as outliers due to their dissimilar geometry from the most generic wall instances.

Table 2. Summary of inliers and outliers for wall elements

Inliers	Outliers
	 

# of items	54	1	2
M dist	2.32	11.08	14.74

4.2.2.1 ifcSlab

Three outliers were detected from 21 of the slab elements (Table 3). In this case, however, these are misclassifications as one of the elements are actually slabs with differing geometry and the latter two are also slabs with openings.

Table 3. Summary of inliers and outliers for slab elements

	Inliers	Outliers	
			
# of items	18	1	2
M dist	2.84	16.05	19.05

4.2.2.2 ifcWindow and ifcDoor

The outlier detection identified a single window and two doors as outliers (Table 4 & 5). Again, these were identified due to their dissimilar shape compared to the majority of the windows and doors.

Table 4. Summary of inliers and outliers for wall elements

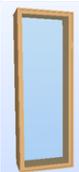
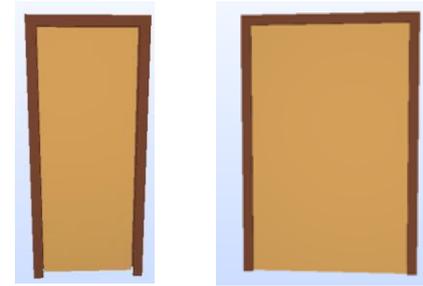
	Inliers	Outliers
		
# of items	24	1
M dist	3.40	16.22

Table 5. Summary of inliers and outliers for door elements

Inliers	Outliers
---------	----------



# of items	16	2
M dist	3.45	11.80

4.2.3 Case study 2: Medical Clinic model

A BIM model of a Medical clinic, provided by [12], was used for the second case study. The model was chosen as it was a larger model than the duplex model, with a total of 1,232 BIM elements. Also, the walls and doors comprised the most number of elements, having more dissimilar elements within their respective classes. Thus, it provided a good candidate to determine the performance of the outlier detection.



Figure 4. The Medical Clinic BIM model

As shown in Table 6, outliers were only found in the walls and doors.

The following sections describes the results with respect to the individual IFC classes with identified outliers.

Table 6. Results of outlier detection for the Medical Clinic BIM model

IFC Class	# of totals	# of Inliers	# of Outliers	Analysis of Results
ifcDoor	254	230	24	Classifiers doors in curtain walls as outliers
ifcRailing	9	9	0	-
ifcSlab	3	3	0	-
ifcStair	9	9	0	-
ifcWall	1080	1025	55	Classifies walls with openings or
StandardCase				

				walls with curvatures as outliers
ifcWindow	58	58	0	-
Total	1,413	1,334	79	

4.2.3.1 ifcWallStandardCase

55 outliers were found out of the 1,080 wall elements. As shown in Table 7, 53 of the elements were found to be dissimilar as these had openings, while two of them were curved walls.

4.2.3.2 ifcDoor

24 outliers were identified out of the 254 doors. As shown in Table 8, 20 of the outliers were doors with glass panes, while the other four were doors inside curtain walls.

Table 7. Summary of inliers and outliers for wall elements

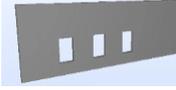
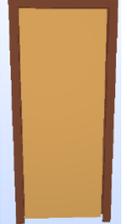
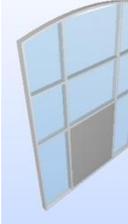
	Inlier	Outlier	
			
# of items	1025	53	2
M dist	0.96	40.95	29.84

Table 8. Summary of inliers and outliers for door elements

	Inlier	Outlier	
			
# of items	230	20	4
M dist	0.13	40.95	34.54

4.3 Summary of the Results

The following summarizes the main limitations of Krijnen's approach identified from the two cases.

- Krijnen's approach classifies elements that are

dissimilar to the most numerous element as outliers. This problem becomes accentuated as the number of elements in classes increases. This problem stems from the fact that the Mahalanobis distance assumes a Gaussian distribution of the data, and thus is not suited for multi-modal distributions. That is, the Mahalanobis distance identifies a single inlier boundary and thus any other elements are taken to be outliers.

- On the other hand, Krijnen's approach will be limited when there are too few elements in a single class, as the outlier detection algorithm does not have enough elements to detect an inlier boundary.
- A more practical limitation is that Krijnen's approach uses a single BIM model, and thus the 'learning' is lost *a posteriori* analysis.

5 Approach using novelty detection

The second approach used was novelty detection. Novelty detection uses a training set that is not polluted by outliers, and is interested in detecting anomalies in new observations. Novelty detection can identify multiple inlier boundaries and can first be trained using datasets prior to the detection of outliers. Thus, it allows the use of data from multiple BIM models, a feature which was utilized in its implementation.

5.1 One-class SVM

Novelty detection can be implemented using one-class Support Vector Machines (SVM) [13]. SVM are a type of supervised learning used either for regression or linear classification. SVMs are referred to as large margin classifiers because the underlying algorithm attempts to identify the hyperplane that best represents the largest separation, or margin, between two classes.

SVMs can also be used as a nonlinear classifier when used with kernels, which are similarity functions that enable the computation of new features as to manual selection (e.g., high order polynomials).

When using SVM's for novelty detection, it is not possible to know *a priori* the type of outliers that may arise, and thus difficult to comprise a training set. In such cases, one-class SVM is used, in which the data set only includes inliers, and is thus a form of semi-supervised learning [14].

5.2 Novelty detection implementation

Figure 5 shows the overall process used to implement novelty detection.

- Model elements from three BIM models (i.e., the Duplex, Clinic and a third residential model) were classified and stored separately with respect to their

IFC classes.

- The datasets were then used to train individual one-class SVM's.
- All one-class SVM's used a non-linear Radial Basis Function (RBF) for their kernels.
- Each trained model was further refined by modifying the hyperparameters, μ and γ to find the values resulting in the highest true negative rates.

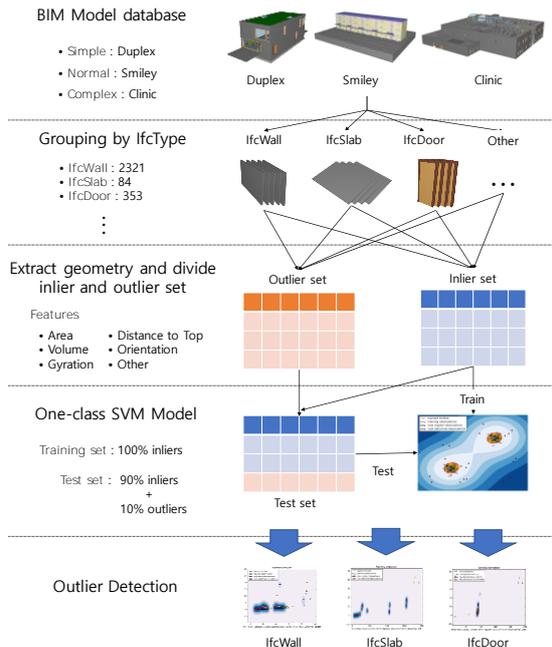


Figure 5. Process overview for novelty detection

5.3 Validation

Validation was performed for each IFC class by using a test set where 10% of the data included outliers (i.e., elements of different IFC entities), and measured whether the one-class SVM correctly detected them.

An example is provided for *ifcWallStandardCase*, which included 2,321 wall elements. As shown in Figure 6, the trained one-class SVM identifies two major boundaries for the walls. Given the test set, it successfully identifies outliers (depicted as yellow points), as abnormal observations. By tuning the model's hyperparameters ($\mu=0.3$, $\gamma=0.1$), the model achieved a true negative rate of 0.983 (Table 9).

Table 10 provides the results for the other IFC classes, especially those that encountered misclassifications using outlier detection. The true negative rates demonstrate that outliers were correctly identified for each of these classes.

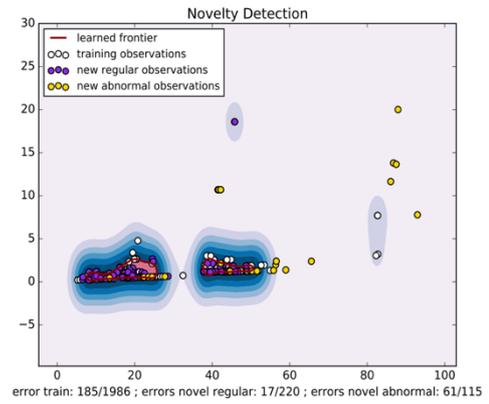


Figure 6. Novelty detection results for *ifcWallStandardCase*

Table 9. Results of novelty detection for the wall elements with tuned hyperparameters

μ	γ	Precision	Recall	TNR	Accuracy
0.1	0.1	0.970	0.908	0.470	0.887
0.1	0.3	0.978	0.903	0.609	0.888
0.3	0.1	0.999	0.698	0.983	0.712
0.3	0.3	0.983	0.698	0.774	0.817

Table 10. Results of novelty detection for other IFC classes

IFC Class	# of Total (outlier)	Prec-ision	Recall	TNR	Accuracy
IfcWall StandardCase	2321 (115)	0.97	0.91	0.47	0.89
IfcSlab	92 (8)	1.00	0.55	1.00	0.59
IfcCovering	388 (35)	1.00	0.91	1.00	0.95
IfcDoor	289 (26)	1.00	0.88	1.00	0.89
IfcWindow	134 (12)	1.00	0.80	1.00	0.81

6 Comparison of the two approaches

As shown in Table 10, results of the novelty detection approach addressed the main limitations identified in the outlier detection approach. Because novelty detection creates multiple boundaries, it could classify different types of elements as inliers, while correctly identifying other class elements as outliers. By creating data sets

using elements from multiple BIM models, the approach also did not suffer from insufficient numbers of data points, as was the case with outlier detection.

Figure 7 shows an ROC curve that compares the accuracy of the two approaches, when applied to all elements of the `ifcWallStandardCase` class. The area under the curve (AUC) is 0.848 for novelty detection, which demonstrates higher performance to outlier detection, whose value is 0.665.

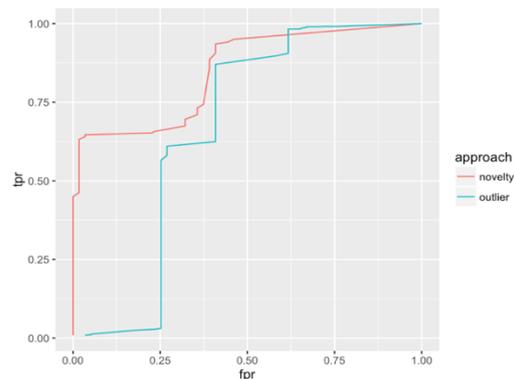


Figure 7. ROC curves for outlier and novelty detection using `ifcWallStandardCase`

7 Conclusions

The IFCs provide a critical role in ensuring the interoperability of BIM models. Ensuring their integrity is thus a fundamental premise for enabling collaboration within the BIM framework. This research examined two techniques in anomaly detection for checking potential errors and misclassifications in the mapping of individual BIM elements to IFC entities. Results showed that novelty detection was superior in terms of overcoming the limitations identified in outlier detection, especially in terms of the ability to train one-class SVM's to identify multiple boundaries of elements within the same IFC class. However, both approaches are limited in restricting features to geometry and not utilizing the semantic relationships between elements within a BIM model. Future research will attempt to address this need by adopting structuring learning techniques (e.g., conditional random fields) to enhance the classification capabilities of these algorithms.

Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2016R1D1A1B03931198)

References

- [1] Krijnen T. and Tamke M. Assessing implicit knowledge in BIM models with machine learning. *Modelling Behaviour*, Springer, 2015
- [2] Tauscher E. and Smarsly K. Generic BIM queries based on the IFC object model using graph theory. *The 16th International Conference on Computing in Civil and Building Engineering*, Osaka, Japan, 2016
- [3] Zhang C., Beetz J. and Weise M. Interoperable validation for IFC building models using open p standards. *Journal of Information Technology in Construction*, 20(2): 24–39, 2015.
- [4] Mazairac W. and Beetz J. BIMQL – An open query language for building information models. *Advanced Engineering Informatics*, 27(4): 444–456, 2013.
- [5] Daum S. and Borrmann A. Processing of Topological BIM Queries using Boundary Representation Based Methods, *Advanced Engineering Informatics*, 28(4): 272–286, 2014.
- [6] Krijnen T. and LAHM Léon Berlo, V. Methodologies for requirement checking on building models:a technology overview. *13th International Conference on Design & Decision Support Systems in Architecture and Urban Planning*, Eindhoven, Netherlands, 2016.
- [7] American Institute of Architects. AIA Document E203; G202: Building Information Modeling Protocol Exhibit. 2008.
- [8] Krijnen T. IfcOpenShell, On-line: <https://ifcopenshell.org>, Accessed: 01/05/2015.
- [9] Paviot T. Pythonocc, 3D CAD/CAE/PLM development framework for the python programming language. On-line: <https://github.com/DURAARK/pyIfcExtract>, Accessed: 01/05/2014.
- [10] Pedregosa F. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*. 12: 2825–2830, 2011.
- [11] Roy D. M., Delphine J-R. and Désiré L. The Mahalanobis distance. *Chemometrics and Intelligent Laboratory Systems*, 50:1–18, 2000.
- [12] William E. Common Building Information Model Files and Tools. On-line: https://www.nibs.org/?page=bsa_commonbimfiles, Accessed: 01/01/2014
- [13] Bernhard S. Platt J. C. Shawe-Taylor J. and Smola A.J., et al. Estimating the support of a high-dimensional distribution. *Neural computation* 13(7): 1443-1471, 2001.
- [14] Escalante H. J. A comparison of outlier detection algorithms for machine learning. *In the Proceedings of the International Conference on Communications in Computing*, pages 228-237, Hawaii, USA, 2005.