Suggested Citation: Wu, J., Akanbi, T., and Zhang, J. (2022). "Constructing Invariant Signatures for AEC Objects to Support BIM-based Analysis Automation Through Object Classification." Journal of Computing in Civil Engineering, 36(4), 04022008.
For Final Published Version, please refer to ASCE Database Here: https://doi.org/10.1061/(ASCE)CP.1943-5487.0001012

Constructing Invariant Signatures for AEC Objects to Support BIM-based Analysis Automation through Object Classification

Jin Wu, S.M.ASCE[1]; Temitope Akanbi, S.M.ASCE[2]; and Jiansong Zhang, Ph.D., A.M.ASCE[3]

Abstract

Building Information Modeling (BIM) object classification is a key step in supporting the full automation of architecture, engineering, and construction (AEC) domain tasks such as cost estimation and building code compliance checking. Machine learning approach is designated to address any classification task without requiring the domain knowledge to be explicitly or manually specified in detail. The success of machine learning, however, relies on the quality and suitability of input features. In order to support seamless interoperability of BIM applications, the authors have proposed invariant signatures that uniquely define each AEC object and capture their intrinsic properties. In this paper, the authors combine the use of invariant signatures together with machine learning approach, to address BIM object classification. The developed invariant signatures include geometric signatures, locational signatures, and metadata signatures. To test the robustness of their use as machine learning features, the authors created a new BIM object dataset with 1,900 AEC objects in five major categories of building elements, including beams, columns, footings, slabs, and walls. The data were manually annotated by independent annotators to ensure the quality. Among those AEC objects in the dataset, 1,330 objects (70% of

[1] Graduate Student, Automation and Intelligent Construction Lab (AutoIC), School of Construction Management Technology, Purdue University, 401 N Grant Street, West Lafayette, IN 47907. Email: wu1275@purdue.edu; Tel: +1-301-275-1272.
[2] Graduate Student, Automation and Intelligent Construction Lab (AutoIC), School of Construction Management Technology, Purdue University, 401 N Grant Street, West Lafayette, IN 47907. Email: email: takanbi@purdue.edu; Tel: +1-224-655-8524.
[3] Assistant Professor, Automation and Intelligent Construction Lab (AutoIC), School of Construction Management Technology, Purdue University, 401 N Grant Street, West Lafayette, IN 47907 (corresponding author). Email: zhan3062@purdue.edu; Tel: +1-765-494-1574; Fax: +1-765-496-2246.

the data) were used as training/development data and 570 objects (30% of the data) were used as testing data. The authors extracted the predefined invariant signatures as features and tested the robustness of them in AEC object classification using different machine learning algorithms. The best-performing algorithm achieved 99.6% F1-measure in the testing data, which outperformed the state of the art (94.9% F1-measure). As a demonstration of the value of such object classification, a comparative experiment was conducted to take off quantities of walls from a student apartment complex, both using the state-of-the-art commercial software and using the object classification-based automation. Consistent results were found between these two quantity takeoff methods, whereas using object classification-based automation further saved time and manual efforts significantly (saved 98.1% of the loading and object selecting time). These results showed that the use of proposed invariant signatures and machine learning algorithms in BIM applications is promising.

Introduction

Building Information Modeling (BIM) is "a data-rich digital representation cataloging the physical and functional characteristics of design and construction" (GSA 2007). Using BIM technology, architecture, engineering, and construction (AEC) models can be built virtually with accurate digital representations. BIM serves different stakeholders at different life cycle phases, such as architects at the design phase and contractors at the construction phase (Eastman et al. 2011). Comparing to a traditional manual process, BIM allows better analysis and control of a project, such as in cost estimation and progress control. These computer-generated models can support the construction, fabrication, and procurement activities of a construction project with accurate information in a digital format, which saves time and effort by supporting the automation of construction engineering and management tasks.

For such automation support, the data of building information models (BIMs) must contain accurate information in their digital representations. A small error in the BIM data may result in malfunctions in the construction processes such as material misuse or dimension mismatch, therefore resulting in costly construction failure and/or rework. To prevent such error from happening, it is critical to generate semantically correct BIM data at each phase of the construction project, which requires an effective checking of the model in different phases. One of the key challenges in such a checking process is the classification/labeling of objects in BIM, i.e., annotating the BIM objects with their correct categories. For BIM data used in practice, however, misclassification or lack of classification of objects is prevalent. For example, in a bridge model provided to Ma et al. (2018), an abutment was misclassified as a beam, seven bearings were misclassified as column, two safety barriers were misclassified as beam, and four shear keys were misclassified as column. “Before the bridge model could be used in the BMS (bridge management system), these objects need to be correctly classified” (Ma et al. 2018). Furthermore, new semantic classes may need to be added depending on the task at hand. For example, in using BIM data for cost estimation and other construction tasks, objects in BIM may need to be classified into categories following the established construction classification systems such as UniFormat, MasterFormat, OmniClass, and UniClass (Afsari and Eastman 2016). To label the data correctly, a certain level of expertise is required. Labeling the data manually is both labor-intensive and error-prone. Although some efforts have been conducted to develop methods that can automatically label BIM data (Ma et al. 2018, Koo et al. 2019), the results of automated labeling still need to be manually checked in practical use, to ensure the quality of the labels. To achieve full automation for different BIM tasks such as quantity takeoff (QTO), structural analysis and code compliance checking, the BIM object classification method still needs further research and development to cover more generic building elements with higher accuracy and interoperability.

Object classification has been studied extensively in computer science. BIM object classification has also attracted the attention of the AEC research community in recent years. Different methods have been used in these studies such as machine learning (Koo et al. 2019; Bloch and Sacks 2018) and rule-based approaches (Ma et al. 2018; Wu and Zhang 2018). Machine learning approaches were found to be more powerful than rule-based approaches such as in space classifications (Bloch and Sacks 2018). Existing machine learning algorithms for object classification can achieve high accuracy in different settings. However, there is a lack of systematic investigation of machine learning algorithms for use in BIM object classification which, to a certain extent was due to the lack of labeled BIM objects data in sufficient amount with distinguishing features that are publicly available. To address this gap, the authors proposed a new set of features based on invariant signatures of AEC object for BIM object classification, created a new BIM object dataset, and tested them using various machine learning algorithms. Invariant signatures of an AEC object are “a set of intrinsic properties of the object that distinguish it from others and that do not change with data schema, software implementation, modeling decisions, and/or language/cultural contexts” (Wu et al. 2021). The authors further tested such object classification in supporting QTO of wall objects comparatively with the traditional manual approach using commercial software.

In the experiments, the authors chose to use the Industry Foundation Classes (IFC) data format. IFC format is open and neutral (BuildingSMART 2018). It aims to solve the interoperability of different BIM software. Different IFC versions have been released, such as IFC2, IFC2x3, and IFC4, and it is still under development to higher versions to better support standardized data representation. IFC4 was accepted as an International Organization for Standardization (ISO) standard (ISO 2013). For the representations of the models, IFC schemas were written in EXPRESS, which is a data modeling language for product data to facilitate data representation and exchange. IFC is under fast development. However, currently, the most widely used version is still IFC2x3, which is henceforth the version that the authors chose to conduct their experiments on.

To collect IFC data for the experiments, the authors searched existing open IFC repositories such as the "Open IFC Model Repository" (Dimyadi and Henderson 2012), which provided 105 IFC model instances, and the NBS National BIM Library (2018) that provided 6,660 IFC model instances. These existing IFC data facilitated open BIM investigation and are helpful and useful in many research activities, such as providing building elements data for BIM visualization development. However, these data repositories did not provide a systematic dataset tailored for object classification, which would require analyzed building models and verified labels for each element of that building. To address this data need, the authors collected data through existing BIM data models from different sources, such as the IFC official website by BuildingSMART and proprietary BIM authoring tools' default collections. Furthermore, to achieve the object classification development goal, a labeled dataset is needed. The authors invited independent annotators to manually label the collected data and discussed among themselves any disagreement. If any disagreement could not be resolved through discussion, the majority vote mechanism was used to decide the label to adopt.

For the task of BIM object classification, in this paper, the authors explored the potential of combining invariant signature-based features and a machine learning approach. In addition to the selection of machine learning algorithms, the selection of features also plays an important role in the performance of machine learning models. With a reported strong performance that was higher than 90%, the feature set used by Koo et al. (2019) seems sufficient. However, exploring a wider range of features could potentially further improve the classification performance, which is crucial in BIM-based automation applications. To gain more insights about a better set of features, the authors did a systematic feature engineering and feature selection. The authors proposed to use invariant signatures to feed into machine learning models. Invariant signatures can distinguish AEC objects from each other (Wu et al. 2021). By nature, it is expected to suit well the task of AEC object classification. The developed invariant signatures can be divided into three main categories: geometric, locational, and metadata. Geometric signatures define the spatial information, mainly about shape, for each individual object such as its height and width; locational signatures depict the position and orientation of each individual object, in reference to the locations of other objects; metadata signatures capture detailed data structure used by each individual object, which may or may not provide much geometric/locational information. To demonstrate the value of BIM object classification enabled by invariant signature-based features, the authors also tested the proposed method in QTO tasks of 60 AEC objects from two building models.

Background

Shared dataset for research

Open datasets are the cornerstones of many research studies and provide a platform for comparison and collaboration. In computer vision, Zhou et al. (2017) developed a method for Scene Parsing through the use of the ADE20K Dataset (Zhou et al. 2018). They introduced and analyzed the ADE20K dataset, which contains diverse annotations of senses, objects, and parts of objects. Consistent annotations of the images were created following a labeling protocol, and the dataset was larger and more diverse compared to many other image datasets, such as COCO (Lin et al.2014) and ImageNet (Russakovsky et al. 2015). The method proposed by Zhou et al. (2017) used a Cascade Segmentation Module to parse the images, that could remove image content and synthesize images automatically. Without the MPCONF196 dataset, the ADE20K dataset, and other related image datasets, such research developments would not have been possible or as successful. In natural language processing, Alamoudi and Alghamdi (2021) developed a deep learning method to conduct sentiment classification and analysis based on the Yelp Review data. Without the public data shared by Yelp, such research would not be as successful. In our AEC research community, published datasets also started to increase in recent years such as in construction site images (Han and Golparvar-Fard 2015) and part-of-speech tagged building codes (Xue and Zhang 2020a,b), among others.

In the same spirit of shared data and research tasks, the authors developed a dataset with 1,900 labeled IFC building objects from five models, which are discussed in detail in the Proposed Method Section later.

Existing methods for BIM (AEC) object classification

The ISO-registered IFC data standard facilitates BIM interoperability by allowing a "one-to-many" information flow between different BIM applications, which enables the mapping between one central model and representations in various applications, therefore brings flexibility into BIM representation. However, such flexibility also creates challenges in data consistency. An important factor in ensuring IFC data consistency and integrity is a correct object classification.

Object classification distinguishes BIM from 3D computer-aided design (CAD) by carrying the semantic information of objects in a building model (Ma et al. 2018). The semantic information, especially the entity type (e.g., beam, column) of the AEC object, plays an important role in other BIM applications. The misuse of entity type could lead to costly errors in later BIM tasks, such as underestimating construction cost and schedule due to incorrect material information. To prevent such misuse, AEC object classification can be used to detect and correct this error at the design phase. Ma et al. (2018) proposed an integrated approach to classify AEC objects that combined domain knowledge of geometric features and pairwise relationships of 3D objects into a tailored matching algorithm. Their algorithm can process various complex 3D geometries and compile a knowledge base in civil engineering. In addition, in their experiment, Ma et al. (2018) achieved 100% accuracy in their two bridge models and provided a knowledge matrix of the objects. In comparison, Wu and Zhang (2018, 2019a) proposed a seven-step method to develop BIM object classification algorithms following a rule-based approach using solely geometric information. Wu and Zhang (2018, 2019a) showed a rule-based algorithm that could automatically label BIM objects with five categories, including beam, column, footing, slab, and wall, with high computational efficiency.

Distinct from the rule-based approaches taken by Ma et al. (2018) and Wu and Zhang (2018, 2019a), Koo et al. (2019) proposed a classification method using a machine learning approach, more specifically using the support vector machines (SVM). Machine learning is gaining popularity and helped solve many practical problems (Kuang and Xu 2018, Seeliger et al. 2018). In AEC domain, for example, Son et al. used machine learning algorithm to improve concrete detection accuracy from 83% to 93%. Yogesh and Vanajakshi used machine learning (SVM) to improve vehicle detection accuracy from 85% to 99%. For AEC object classification, Koo et al. (2019) proposed a feature set that could map IFC objects to selected IFC classes with an average F1-score of 94.9% for eight classes. While their feature set was working and SVM was one of the most robust supervised machine learning algorithms, different features and different algorithms may outperform the SVM proposed by Koo et al. (2019) for AEC object classification, which is representing the state of the art, to push automated BIM object classification closer to the implementation level. To support a better interoperability, the authors proposed to use invariant signatures as features to help solve this AEC object classification problem while supporting BIM interoperability, because invariant signatures are platform-independent and stable. With the invariant signatures-based features, the authors systematically experimented with five promising categories of machine learning algorithms for AEC object classification on their collected data from the newly created data repository.

Feature engineering

To fully explore the potential of machine learning algorithms, a systematic feature engineering is needed.

Feature engineering is one of the critical steps to ensure that the machine learning algorithms can generate good models to achieve the desired classification results. In addition, Feature engineering is one of the most time-consuming and challenging tasks in data mining (Zhang et al. 2018). According to the Occam's Razor (Bethel 2009), the fewer features used, the more robust machine learning algorithms can potentially be, so the main task in feature engineering is to select a small set of features that maintain a good performance in the target machine learning task. Koo et al. (2019) used a feature set consisted of certain geometric information and relational information. The authors proposed to use a different set of features with more geometric information, selected based on invariant signatures of AEC objects and a systematic feature engineering.

Machine learning algorithms

Machine learning plays an important role in the success of many modern computing technologies such as artificial intelligence. There are traditional statistical methods such as linear regression and logistic regression, and new methods that require large computational efforts such as convolutional neural networks (CNN) (Kuang and Xu 2018) and generative adversarial networks (GAN) (Seeliger et al. 2018). Machine learning algorithms can be categorized into two types – supervised and unsupervised. Supervised algorithms require ground truth as input, which is the correct labels of the data. Unsupervised algorithms do not have such prerequisite. Based on the purpose of application, machine learning algorithms can also be categorized by tasks, e.g., classification, regression, and clustering. Kotsiantis (2007) reviewed major machine learning algorithms for the classification task: logic-based algorithms, including decision trees and rule-based algorithms; perception-based algorithms, including logistic regression (Fraix-Burnet et al. 2014), neural networks (Cartwright 2015, Hugh and Nawwaf 2018, Angermueller et al. 2016, Zeng et al. 2019, David et al. 2016, Ryan 2017, and Mahanta 2017), and radial basis function networks; statistical learning algorithms, including naive Bayes and Bayesian network (Kotsiantis 2007, Quinlan 1986, and Jensen 1996); instance-based learning, including nearest neighbors and k-nearest neighbors (Veropoulos et al. 1999); and SVM. Kotsiantis (2007) summarized the properties and method of each type of algorithms and provided analysis of each of them. For supervised object classification, multiple existing algorithms provide accurate results for different cases. Some of the most promising ones include logic-based algorithms, perception-based techniques, statistical learning algorithms, instance-based algorithms, and SVM (Kotsiantis 2007). In addition to those machine learning algorithms, boosting method can increase the overall accuracy by reducing variance (Trevor 2009) and therefore can be used to develop new machine learning algorithms. For example, Random Forest (Trevor 2009) is a machine learning algorithm that is built on substantial modification of the bagging method which in turn is also known as bootstrap aggression to reduce the variance of an estimating function. The bagging method works well for tree structures. Random forest pushes that further to build a large collection of de-correlated trees and take the mean prediction of the many trees in classification results using the bagging method.

Quantity takeoff (QTO)

The cost estimates of construction projects highly depend on the accuracy of the quantities taken off for each individual cost component (Vitasek and Matejka 2017; Alshabab et al. 2017). BIM tools are faster and more reliable to use than the traditional manual QTO approach and can help provide the necessary accuracy required for QTO (Santos et al. 2017). A few researchers have developed BIM-based QTO processes and methods that vastly improved both the accuracy and the time efficiency involved in generating QTO. For example, Akanbi et al. (2020) developed a new data-driven algorithm development method for developing QTO algorithms that process IFC-based BIMs designed in different BIM authoring tools and workflows. Han et al. (2017) proposed a method that improves the accuracy of QTO by extracting geometric dimensions from a proprietary 3D model. Mandava and Zhang (2016) developed a method for generating QTO by leveraging the cartesian points of building objects in an IFC-based BIM. Liu et al. (2016) proposed an ontology-based semantic method to extract QTO information from models. Choi et al. (2015) developed a BIM-based method that computes schematic QTO, improving the accuracy of early design stage cost estimates. Although BIM can provide these benefits in improving the accuracy of QTO results and time efficiency of QTO processes, accurate object classification is further needed to avoid misrepresentation or misuse of BIM data, which is the main barrier faced by fully automated QTO and cost estimation. The authors' proposed method in this research, aims to extract all needed quantities fully automatically using invariant signatures-based object classification.

Proposed Method

For the development of fully automated QTO methods utilizing invariant signatures and machine learning, a comprehensive and robust dataset is required. The data should contain all needed common AEC objects including beams, columns, footings, slabs, and walls. The data also needs to be labeled consistently. After data collection, there are three main steps in the proposed method: (1) construct invariant signatures, (2) apply invariant signatures to object classification, and (3) apply object classification to QTO.

Construct invariant signatures

As described in the background section, the AEC object classification can potentially be better addressed with invariant signatures for seamless interoperability. In the first step, the authors propose to construct invariant signatures in three sub-types: geometric signatures, locational signatures, and metadata signatures. Geometric signatures capture the geometric information of common shapes, such as rectangle and cylinder. Locational signatures capture the position information of an object, including the absolute position, relative position, and orientation. Metadata signatures capture representation-level information, especially the representation used in IFC, e.g., the average number of vertices among faces in a boundary representation (Brep).

Apply invariant signatures to AEC object classification

After constructing the invariant signatures, the authors propose to test the robustness of these properties in two BIM tasks, object classification and furthermore QTO. Object classification is needed for a lot of common tasks in BIM. Classification accuracy is the premise for any subsequence applications. For example, for QTO of wall volume, it can generate costly error if any wall object is mistakenly classified as a slab. Based on accurate QTO results, in turn, cost estimation is another important task in BIM, which therefore also relies on the accurate classification result.

To achieve a high accuracy in AEC object classification, the authors propose to use invariant signatures as features and feed them into machine learning algorithms to classify the AEC object. For algorithm selection, the authors propose to use five promising categories of machine learning algorithms including classic and deep learning algorithms and select the best-performing algorithm to test on the testing dataset.

Apply invariant signature-based object classification to QTO

For evaluating the developed object classification in generating QTO, the authors utilized the developed object classification algorithm on two types of units from a student apartment complex. The models were created in Autodesk Revit. The results generated using the authors' developed object classification algorithm are compared against results generated using a commercial software - Autodesk Navisworks and manual QTO by industry experts. Our expectation is that the algorithm should generate results within a 1% error margin of the results generated by the industry expert using commercial software. Two types of errors may occur, one is from the equations and processes in calculating needed QTO results, with correct invariant signature-based features. The other type are errors in the feature extraction process, where the resulting invariant signature-based features would not be error free. To resolve this, in the algorithm development for both tasks, if the existing properties in under-development invariant signatures do not provide enough information, iterative development of the invariant signatures will be performed, i.e., adding more needed properties to the invariant signatures following a data-driven approach.

Data Collection

To collect needed data, the authors explored existing open BIM repositories, including the "Open IFC Model Repository" (Dimyadi and Henderson 2012) and the NBS National BIM Library (2018), which contain 105 and 6,660 IFC data instances, respectively. They provided good quality models for visualization. However, these data were not tailored for object classification, and therefore they did not have verified object class labels. In contrast, the authors developed a new dataset tailored for BIM object classification.

Introducing a new dataset

The authors selected five models with 1,900 instances of beams, columns, footings, slabs, and walls. The selected models included a duplex apartment model (Deplex_A), a Revit architectural sample model (Rac_basic), a Revit advanced structural sample model (Rst_advanced), a Revit basic structural sample model (Rst_basic), and the Revit technical school sample structural model (Tech_school). The duplex apartment model was already in IFC, whereas the Revit models were transformed into IFC data format. To facilitate data sharing, the authors hosted this open dataset at Purdue University Research Repository (Wu and Zhang 2019c), an initial version of which was described in (Wu and Zhang 2019b). The new dataset (Wu and Zhang 2021) can be used directly by researchers and developers to develop and test object classification algorithms. The dataset contains not only the 1,900 IFC instances of beams, columns, footings, slabs, and walls but also the object class type labels as described below in the Section of Object Class Labeling, and selected object features as described in the Section of Experiment - Construct Invariant Signatures.

Object class labeling

The authors invited three independent annotators with AEC and BIM background to label the data. The three annotators labeled the data independently leading to an average inter-annotator agreement of 87.21%. Then the authors led a discussion of the disagreed instances aiming to reach consensus. After that, the average inter-annotator agreement was improved to 99.05% and no objects had more than two different labels. For the rest of the 18 instances (0.95%) where disagreement remained, the authors used the majority vote to decide object type labels to use. Table 1 shows the numbers of AEC object instances in each type of object labels.

Experiment

In the experiment, the authors randomly split the 1,900 objects into training dataset and testing dataset following a 7:3 ratio. As a result, 1,330 objects were used as training/development data and 570 objects were used as testing data. During the development phase, only the training data set was used.

Construct invariant signatures

The authors followed the proposed method to construct invariant signatures as the interoperable representations of AEC objects, which can later be used as machine learning features. Based on a preliminary analysis of the data, the authors established invariant signatures with three main sub-types: geometric signatures, locational signatures, and metadata signatures. Geometric signatures contain size (e.g., length) and shape (e.g., I-shape) information of an object. Locational signatures include the positional information of all the objects from the same model. Metadata signatures include the data structure (e.g., IFC entities) and statistical data (e.g., the number of faces) used by each object. The invariant signatures include both categorical features and numeric features. Categorical features can be transformed into numeric values using discrete numbers or binary representations. For features that an object may miss value for (i.e., empty feature value), a default value of "0" or empty was assigned. In addition, metadata signatures may have certain information overlap with geometric and locational signatures. For example, if an object has a nonzero value for the I-shape signatures then the object should also have a nonzero value for the extruded area solid signature. Table 2. Shows all the developed invariant signatures with their value types and descriptions.

The signature set contains built-in feature values, such as *Rec_L*, *Rec_W*, *Rec_H*, which represent the length, width, and height of a 3D rectangular shape, respectively. For regular shapes, rectangular, cylindrical and ring shape features were included. For irregular shapes, the authors defined two sets of signatures, one for extruded area solid and one for Brep. The extruded area solid signature for irregular shapes included a set of 3 decimal numbers to represent the length, width, and thickness dimensions of the bounding box, respectively. For Brep representation, in addition to the dimensions of the bounding box, the signatures that indicate the number of faces were also included. Table 3 shows 16 examples of the invariant signature values of independent objects.

The proposed invariant signatures are expected to uniquely identify AEC objects. To allow this identification, the invariant signatures shall describe all the major information embedded in each object. This can be reflected by the statistical relations of each invariant signature with AEC object types. For one type of AEC object, the invariant signature values fall into certain range. For example, the height of a slab object usually does not exceed 0.3 meters. However, the height of a wall object usually does not fall below 0.3 meters. This is reflected in the distribution of each invariant signature values. Figs. 1 to 3 show three plots of object instances' distributions across three different invariant signature features, respectively. Fig. 1 shows the distribution of the locational signature *O3* (elevation of an object) on different object types. It shows that footings have smaller values of elevation. Fig. 2 shows the distribution of the geometric signature *Length* (horizontal dimension) on different object types. It shows that columns have smaller values of length, comparing to other types. Fig. 3 shows the distribution of the geometric signature *Cir_R* on different object types. It shows that circle is not used in walls or slabs.

Apply invariant signatures to AEC object classification

With the proposed invariant signatures, five types of machine learning algorithms were tested using Waikato Environment for Knowledge Analysis (Weka), which is an open machine learning platform developed by the University of Waikato (Witten et al. 2016). Ten-fold cross-validation was used to avoid overfitting.

<u>Perceptron-based techniques: neural networks</u>

A single layer artificial neural network would have the same structure with the linear regression model, so the authors chose to use deep learning (neural networks with multiple layers) for perceptron-based techniques. After parameter tuning, the authors selected the best performing configuration (two-layer with 45 nodes for each layer) to compare with other machine learning algorithms. Fig. 4 shows a visualization of the accuracies of these results. Table 4 shows the detailed classification results of the best configuration. The accuracy started to drop after 3 layers because of overfitting, so the deep learning approach with more hidden layers will not generate better results unless more training data is used.

<u>Logic-based: decision table</u>

A decision table is a graphical implementation of decision trees. The authors used best-first search (BFS) and greedy stepwise search. For BFS, the authors implemented forward, backward, and bidirectional search methods. Fig. 5 shows a visualization of the detailed training results. Table 5 shows the details of the classification results of the best configuration.

Statistical machine learning: Bayesian network

Naïve Bayes assumes that the features are independent from each other. However, some of the features may be highly correlated. To address that, the authors used a Bayesian network instead. Bayesian network used probability graphical model, which was based on the conditional dependencies of the parent nodes on each node in the network. The classification results depend on the number of parent nodes implemented. As a result, the authors used different numbers of parent nodes to train the best model. The visualization of the training results is shown in Fig. 6. Table 6 shows the details of the classification results of the best configuration.

Support vector machines (SVM)

SVM uses hyperplanes to separate data into different classes. The classification results depend on a regularization term, which is defined by a soft margin constant. The authors experimented with different soft margin constants to find the best performance. A visualization of the accuracy using different soft margin constants at different range scales is shown in Fig. 7. Table 7 shows the details of the classification results of the best configuration.

Random forest

Random forest uses a collection of decision trees and takes the statistical majority of the results of each tree. Search depth will determine the classification results. A low depth value may lead to underfitting, whereas a high depth value may lead to overfitting. The visualization of accuracy on different configurations is shown in Fig. 8. Table 8 shows the details of the classification results of the best configuration.

Apply invariant signature-based object classification to QTO

Based on the automation enabled by object classification, QTO jobs can be implemented with high precision and efficiency. To demonstrate the QTO potential and verify the robustness of automated object classification based on invariant signatures and machine learning, the authors tested the invariant signature-based objection classification on QTO of 2 types of units from a student apartment (Fig. 9). The selected models in Fig. 9a show these two types of student apartment units (2-bedroom unit and 4-bedroom unit) extracted from the student apartment complex (Fig. 9b). As shown in Fig. 9a, the units comprise of wall components and floor components. For this experiment, the authors randomly selected 30 wall objects from the 4-bedroom unit and 30 wall objects from the 2-bedroom unit. The results obtained using our algorithms were compared with results of the state-of-the-art method using commercial software. The comparison of the results showed consistent values for the length, width, height, and volume of the selected objects.

Results & Analysis

Random forest achieved the highest F1-measure among all the algorithms in the training phase, so the authors selected random forest as the best-performing algorithm to test on the testing dataset. The overall F1-measure was 99.6% as a result. This was 0.20% higher than the training accuracy. This shows the algorithm did not overfit and perform well in the testing dataset. Table 9 shows the detailed testing results.

Error analysis

For the best performing machine learning algorithm, the random forest, the errors were mainly due to the cutoff values of decision trees. The invariant signatures were verified to be correct, so the errors occurred because of the limitations in selected machine learning algorithm. A level of depth 7 achieved the best performance in the cross-validated data. More levels might lead to overfitting, which would reduce the training performance. The errors included 2 slabs misclassified as 1 footing and 1 wall, respectively. Fig. 10 shows the trained classifier. The tree on the left shows a branch that successfully classified all instances, which are shown in the bracket in the leaf nodes (5 columns and 15 beams were classified). The tree on the right shows a branch that only successfully classified a part of them: among the 10 objects classified as columns, 8 of them were correct and 2 of them were incorrect, while among the 18 objects classified as beams, 15 of them were correct and 3 of them were incorrect. Although each tree may make wrong predictions, e.g., in one of the branches, among 157 classified columns, 47 of them were wrong, after the voting process, the accuracy increased significantly. For example, Table 10 shows the two incorrectly classified instances. The depth of the voting tree ranged from 4 (*mRatio < 0.39; mLow < -3.901; mRatio < 0.37; Z3 >= 0.5: Footing (238/0)*) to 7 (*mLow < -7.151; ExtrudZ >= 0.5; Width >= 0.2375; X1 >= 0.9; O1 < -0.21903; Width >= 0.3178; I_R < 0.0076: Column (3/1)*).

Feature set (invariant signatures) analysis

Our invariant signature-based feature set led to a good performance (i.e., 99.6% F1 score in testing data) but may contain extra information that would be needed only for subcategories. After feature selection, the authors proposed a 6-feature set which achieved 98.65% F1 score. The authors analyzed the features as follows. The *Cir_H* feature describes the height of a cylinder shape. A zero value for this feature means a non-cylinder shape. It mainly helps distinguishing footings and columns from other object types. The *Width* feature describes the width of any shape. It is one of the general features that help distinguish many different objects. The *O3* feature describes the elevation of any object. The elevation is one of the most important pieces of information based on locational features. The *Zmin* feature describes the lowest point of Brep representation, which not only tells whether the object used that representation or not, but also provides information about the size. The *ExtrudZ* feature describes the extruded direction of a swept solid representation, which not only tells whether the object used that representation or not, but also provides information about the orientation direction of the object, which is important to differentiate beams from other types.

Comparison with the state-of-the-art algorithm

The BIM object classification algorithm developed by Ma et al. (2018) encoded experts' insights as rules. A direct comparison would be difficult as the objects used by Ma et al. (2018) and the authors were of different types. To compare with Koo et al. (2019), the authors reproduced the method by Koo et al. (2019) based on the authors' understanding of the feature values used by Koo et al. (2019). Because the authors could not get access to the original data used by Koo et al. (2019), the authors used their own dataset while extracting the features that Koo et al. (2019) proposed. After model training and parameter tuning, the authors obtained 94.86% (94.12% on testing) accuracy using SVM with C=200000, which is in the same range as the experiment of Koo et al. (2019). The authors also tested random forest machine learning algorithm using the same set of features and obtained 98.87% (99.12% on testing) accuracy (Table 11). The random forest machine learning algorithm showed high performance and is expected to be robust in similar types of BIM objects.

QTO results

Table 12 shows a few sample QTO results comparatively using the manual approach and using our method. On average, the difference of the QTO results between using the proposed method (random forest-based object classification) and using traditional approach (manual approach as gold standard) is 0.3%, within the desirable threshold 1%. More importantly, the time difference for QTO tasks was significant. On average, each object took 4.23 seconds in manual QTO, whereas it only took 0.079 seconds using our proposed method. It saved 98.1% of time using our proposed method comparing to the manual approach. The time for QTO using the traditional approach was mainly consumed in the loading, project set-up, and element selection. This shows that the proposed invariant signature-based object classifications can produce correct QTO results that are comparable with traditional approach but with much less time, which illustrates the robustness of the invariant signature-based object classification in QTO application.

Contributions to the Body of Knowledge

There are five main contributions in this paper. First, the authors proposed to use invariant signatures of AEC objects as features/input for BIM object classification. Invariant signatures provided a theoretical foundation for universal and seamless BIM interoperability and practical solutions to push it forward. Second, the authors established an open dataset for BIM object classification, which can be used to reproduce the results and conduct further research in BIM-based automation. This open and consistent data can promote collaboration and comparison between different methods in this task and further BIM-based automation tasks in the AEC domain. Third, the authors improved the state-of-the-art F1-measure (accuracy) for machine learning-based BIM object classification, from 94.9% to 99.6%. This helps push BIM-based full automation of AEC tasks one step closer to reality, which would need 100% accuracy. Fourth, a systematic approach was conducted to test and compare different machine learning algorithms. The best performing algorithm, random forest, was shown to outperform other machine learning algorithms in both the authors' invariant signatures-based feature set and the state-of-the-art features. Last but not least, object classification is the premise of the full automation of many

BIM applications. In this experiment, the invariant signatures-based objection classification showed that it led to comparable QTO results with the traditional approach whereas saving time and manual effort significantly. It can be expected to have broader use in other BIM applications such as building code compliance checking and building energy simulation/analysis to further pursue seamless BIM interoperability and full automation of AEC tasks.

Conclusions

A high classification accuracy and BIM interoperability are two pillars to support BIM-based automation in AEC tasks such as cost estimation, structural analysis, code compliance checking, energy modeling and simulation. Advancing either pillar alone without the other is not sufficient for practical consideration. To synergistically advance both, in the paper, the authors investigated the use of invariant signatures of AEC object as features/input for BIM object classification. Five types of machine learning algorithms were systematically tested. The features using invariant signatures were shown to deliver better BIM object classification results comparing to the state-of-the-art features. In addition, the authors showed that the random forest machine learning algorithm outperformed SVM (the state of the art) and other machine learning algorithms. The overall performance in object classification was improved to 99.6% F1-measure. This shows that the invariant signatures and the random forest machine learning algorithm are promising in BIM object classification. In addition. The random forest-based object classification was tested to achieve less than 1% error in QTO, compared with a gold standard developed using commercial software and traditional manual approach, while achieved significant time saving (98.13%) in conducting the QTO tasks. The invariant signatures in combination with machine learning algorithms are expected to be applicable in a variety of other different BIM applications such as cost estimation, automated code compliance checking, and building energy simulation and analysis, with high accuracy and efficiency.

Limitations and Future Work

As a first step towards building a comprehensive BIM object classifier, the authors established a new dataset tailored for object classification. The dataset contains five BIM models with 1,900 object instances in IFC format. The dataset provides verified labels and processed features for the objects in a systematic way. The dataset was still small compared to the dataset in other domains, such as the ADE20K (Zhou et al. 2018) with 22,210 images for image processing, so the authors are planning to continuously grow the dataset to contain more verified models and objects. The authors also plan to test the potential of invariant signature-based object classification in other BIM applications such as automated building code compliance checking, and investigate more advanced deep learning algorithms to further reduce the needed feature engineering efforts.

Data Availability Statement

Some data and models used during the study are available in a repository online in accordance with funder data retention policies.

- 1,900 IFC instances of beams, columns, footings, slabs, and walls labeled with invariant signatures (https://purr.purdue.edu/publications/3832/1)

Acknowledgments

The authors would like to thank the National Science Foundation (NSF). This material is based on work supported by the NSF under Grant No. 1745374. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

Afsari, K., and Eastman, C. (2016). "A Comparison of Construction Classification Systems Used for Classifying Building Product Models." Proc., 2016 ASC Annual International Conference, Denver, CO., 101-108.

Alamoudi, E.S., and Alghamdi, N.S. (2021). "Sentiment classification and aspect-based sentiment analysis on yelp reviews using deep learning and word embeddings." *Journal of decision systems,* 1-23.

Alshabab, M., Vysotskiy, A., Khali, T., and Petrochenko, M. (2017). "BIM-Based Quantity Takeoff." *Construction of Unique Buildings and Structures*, 4(55), 124-134. EISSN: 2304-6295.

Angermueller, C., Pärnamaa, T., Parts, L., and Stegle, O. (2016). "Deep learning for computational biology." *Molecular Systems Biology*, 12(7).

Bethel, E.W. (2009). "Occam's razor and petascale visual data analysis." *Journal of Physics: Conference Series,* 2009, 180(1), 012084(18).

Bloch, T., and Sacks, R. (2018). "Comparing machine learning and rule-based inferencing for semantic enrichment of BIM models." *Automation in Construction*, 91(2018), 256-272.

BuildingSMART. (2018). "IFC overview summary." <http://www.buildingsmart-tech.org/specifications/ifc-overview> (Feb 14, 2019).

Cartwright, H. (2015). "Artificial neural networks." Springer, New York. ISBN: 1-4939-2239-4.

Choi, J., Kim, H., and Kim, I. (2015). "Open BIM-based quantity takeoff system for schematic estimation of building frame in early design stage." *Journal of Computational Design and Engineering,* 2(2015), 16-25.

David. S., Aja, H., Chris, J.M., Arthur, G., Laurent, S., George, V.D.D., Julian, S., Ioannis, A., Veda, P., Marc, L., Sander, D., Dominik, G., John, N., Nal K., Ilya, S., Timothy, L., Madeleine, L., Koray, K., Thore, G., and Demis, H. (2016). "Mastering the game of Go with deep neural networks and tree search." *Nature*, 529(7587), 484.

Dimyadi, J., and Henderson, S. (2012). "Open IFC Model Repository." < http://openifcmodel.cs.auckland.ac.nz/> (Feb. 13, 2019)

Eastman, C., Teicholz, P., Sacks, R. and Liston, K. (2011). "BIM handbook: a guide to building information modeling for owners, managers, designers, engineers and contractors." John Wiley &Sons, Inc., 2nd edition, New Jersey.

Fraix-Burnet, D., Valls-Gabaud, D., and Grégoire, G. (2014). “Logistic regression.” *EAS Publication Series*, 66, 89-12.

General Service Administration (GSA). (2007). “BIM guide overview.” *GSA BIM Guide Series 01.*, U.S. General Services Administration, Washington, DC. <http://www.gsa.gov/portal/getMediaData?mediaId=226771> (Feb. 13, 2019)

Han, K., and Golparvar-Fard, M. (2015). “Appearance-based material classification for monitoring of operation-level construction progress using 4D BIM and site photologs.” Automation in Construction, 53(May 2015), 44-57.

Han, P., Siu, M., AbouRizk, S., Hu, D., and Hermann, U. (2017). “3D model-based quantity takeoff for construction estimates.” *Proc., ASCE International Conference on Computing in Civil Engineering,* ASCE, Reston, VA, 118-124.

Hugh M. C., and Nawwaf, K. (2018). “Using artificial intelligence in chemistry and biology a practical guide.” CRC Press, Boca Raton, FL. ISBN: 0-429-12759-6.

International Organization for Standardization (ISO). (2013). “ISO 16739:2013 Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries.” International Organization for Standardization, Geneva, Switzerland. <https://www.iso.org/standard/51622.html.> (Feb. 13, 2019)

Jensen, F. (1996). “An Introduction to Bayesian Networks”. Springer. Germany.

Koo, B., La, S., Cho, N.W., and Yu, Y. (2019). “Using support vector machines to classify building elements for checking the semantic integrity of building information models.” *Automation in Construction*, 98, 183-194.

Kotsiantis, S. B. (2007). “Supervised machine learning: a review of classification techniques.” *Informatica*, 31(3), 249(20).

Kuang, D., and Xu, B. (2018) “Predicting kinetic triplets using a 1d convolutional neural network.” *Thermochimica Acta*, 669, 8-15.

Langley, P. (1994). “Selection of relevant features in machine learning.” Institute for the study of learning and expertise Palo Alto, California. <https://apps.dtic.mil/dtic/tr/fulltext/u2/a292575.pdf > (Feb. 13, 2019)

Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C.L. (2014). “Microsoft COCO: common objects in context.” *Lecture Notes in Computer Science*, 8693(5), 740-755.

Liu, H., Lu, M., and Al-Hussein, M. (2016). “Ontology-based semantic approach for construction-oriented quantity take-off from BIM models in the light-frame building industry.” *Advanced Engineering Informatics*, 30 (2016), pp 190-207.

Ma, L., Sacks, R., Kattel, U., and Bloch, T. (2018). “3D object classification using geometric features and pairwise relationships.” *Computer-Aided Civil and Infrastructure. Engineering.*, 33(2), 152-164.

Mahanta, J. (2017). “Introduction to Neural Networks, Advantages and Applications.” <https://towardsdatascience.com/introduction-to-neural-networks-advantages-and-applications-96851bd1a207> (Nov. 18, 2018).

Mandava, B., and Zhang, J. (2016). “A new automated quantity takeoff method for BIM-based bridge designs.” *Proc., CIB W78*, Conseil International du Bâtiment (CIB), Rotterdam, The Netherlands.

NBS of UK. (2014). “NBS National BIM Library.” <http://www.nationalbimlibrary.com/> (Nov. 18, 2018).

Quinlan, J.R. (1986). “Induction of Decision Trees.” *Machine Learning*, 1(1), 81-106. DOI: 10.1023/A:1022643204877.

Ryan G. R. (1997). “Intelligent Visual Inspection Using artificial neural networks.” Springer, New York, U.S.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A., and Li, F. (2015). “ImageNet large scale visual recognition challenge.” *International Journal of Computer Vision,* 115(3), 211-252.

Santos, A., Costa, A., and Grillo, A. (2017). “Bibliometric analysis and review of Building Information Modeling literature published between 2005 and 2015.” *Automation in Construction,* 80(2017), 118-136.

Seeliger, K., Güçlü, U., Ambrogioni, L., Güçlütürk, Y., and Gerven, M.A.J. (2018). “Generative adversarial networks for reconstructing natural images from brain activity.” *NeuroImage*, 181, 775-785.

Son, H., Kim, C., Kim, C. (2012). “Automated color model–based concrete detection in construction-site images by using machine learning algorithms.” *Journal of Computing in Civil Engineering*, 26(3), 421-433.

Trevor, H., Robert, T., and Jerome, H. (2009). “The elements of statistical learning: data mining, inference, and prediction.” Springer, New York, US.

Veropoulos, K., Campbell, C., and Cristianini, N. (1999). “Controlling the sensitivity of support vector machines.” *Proceedings of the International Joint Conference on AI,* 55-60.

Witten, I.H., Frank, Eibe., Hall, M.A., and Paul, C.J. (2016). “Data Mining, 4th Edition.” Morgan Kaufmann, Burlington, MA.

Wu, J., and Zhang, J. (2018). “Automated BIM object classification to support BIM interoperability.” *Proceedings of the Construction Research Congress*, ASCE, Reston, VA, 706-715.

Wu, J., and Zhang, J. (2019a). “New automated BIM object classification method to support BIM interoperability.” *Journal of Computing in Civil Engineering*, 33(5), 04019033.

Wu, J., and Zhang, J. (2019b). “Introducing geometric signatures of architecture, engineering, and construction objects and a new BIM dataset.” *Proceedings of the 2019 ASCE International Conference on Computing in Civil Engineering*, ASCE, Reston, VA, 264-271.

Wu, J. and Zhang, J. (2019c). “Building Information Modelling (BIM) data repository with labels.” Purdue University Research Repository. doi:10.4231/60V2-PJ72.

Wu, J. and Zhang, J. (2021). “Invariant Signatures of Architecture, Engineering, and Construction (AEC) Objects in Industry Foundation Classes (IFC)-based Building Information Modeling.” Purdue University Research Repository. doi: 10.4231/7VW7-0129.

Wu, J., Sadraddin, H.L., Ren, R., Zhang, J., and Shao, X. (2021). “Invariant signatures of architecture, engineering, and construction objects to support BIM interoperability between architectural design and structural analysis.” *Journal of Construction Engineering and Management*, 147(1), 04020148.

Xue, X., and Zhang, J. (2020a). “Building codes Part-of-Speech tagging performance improvement by error-driven transformational rules.” *Journal of Computing in Civil Engineering*, 34(5), 04020035.

Xue, X., and Zhang, J. (2020b). “Erratum for 'Building codes Part-of-Speech tagging performance improvement by error-driven transformational rules.' ” *Journal of Computing in Civil Engineering*, 35(1), 08220002.

Yogesh, G.K.V., and Vanajakshi, L. (2018). "Automated Tolling Solution with Novel Inductive Loop Detectors Using Machine Learning Techniques." *Journal of Computing in Civil Engineering*, 32(6), 4018046.

Zeng, G., Song, R., Hu, X., Chen, Y., and Zhou, X. (2019). "Applying convolutional neural network for military object detection on embedded platform." *Communications in Computer and Information Science*, 994, 131-141.

Zhang, J., Fogelman-Soulié, F., Largeron, C. (2018). "Towards Automatic Complex Feature Engineering." *Lecture Notes in Computer Science*, 11234, 312-322.

Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., and Torralba, A. (2018). "Semantic understanding of scenes through ADE20k dataset." *International Jorunal of Computer Vision*, 1-20.

Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., and Torralba, A. (2017). "Scene parsing through ADE20K dataset." *Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition*, 2017, 5122-5130.

Tables

Table 1. Numbers of instances in each object types.

Object Type	Number of Instances
Beam	790
Column	412
Footing	354
Slab	79
Wall	265
Total	*1,900*

Table 2. Developed object invariant signatures.

Invariant Signature Name	Value Type	Signatures Type	Description
Rec_L	Numerical	Geometric	Length of a rectangular shape
Rec_W	Numerical	Geometric	Width of a rectangular shape
Rec_H	Numerical	Geometric	Height of a rectangular shape
Cir_R	Numerical	Geometric	Radius of a cylinder shape
Cir_H	Numerical	Geometric	Height of a cylinder shape
R_R	Numerical	Geometric	Radius of a ring shape
R_H	Numerical	Geometric	Height of a ring shape
R_T	Numerical	Geometric	Thickness of a ring shape
I_W	Numerical	Geometric	Width of an I-shape
I_H	Numerical	Geometric	Height of an I-shape (extruded depth)
I_D	Numerical	Geometric	Depth of an I-shape
I_R	Numerical	Geometric	Radius of an I-shape
I_WT	Numerical	Geometric	Wide thickness of an I-shape
I_FT	Numerical	Geometric	Flange thickness of an I-shape
X1	Numerical	Locational	
X2	Numerical	Locational	Vector x for placement
X3	Numerical	Locational	
Z1	Numerical	Locational	
Z2	Numerical	Locational	Vector z for placement
Z3	Numerical	Locational	
O1	Numerical	Locational	
O2	Numerical	Locational	Center Cartesian point of the object
O3	Numerical	Locational	
mHigh	Numerical	Locational	Highest elevation of the original model
mLow	Numerical	Locational	Lowest elevation of the original model
mRatio	Numerical	Locational	mRatio = (mHigh – (mHigh + mLow) / 2) / (mHigh - mLow)
Length	Numerical	Geometric	Length of the bounding box
Width	Numerical	Geometric	Width of the bounding box

Height	Numerical	Geometric	Height of the bounding box
Volume	Numerical	Geometric	Volume of the bounding box
Items	Integer	Metadata	Number of items
Faces	Integer	Metadata	Number of faces of Brep representation
F3	Integer	Metadata	Number of faces with 3 edges of Brep representation
F4	Integer	Metadata	Number of faces with 4 edges of Brep representation
F5	Integer	Metadata	Number of faces with 5 edges of Brep representation
F6	Integer	Metadata	Number of faces with 6 edges of Brep representation
F7	Integer	Metadata	Number of faces with 7 edges of Brep representation
AveVerti	Numerical	Metadata	Average number of edges of all faces of Brep representations
Xmax	Numerical	Metadata	Max value in x direction
Xmin	Numerical	Metadata	Min value in x direction
Ymax	Numerical	Metadata	Max value in y direction
Ymin	Numerical	Metadata	Min value in y direction
Zmax	Numerical	Metadata	Max value in z direction
Zmin	Numerical	Metadata	Min value in z direction
Brep	Nominal (Binary)	Metadata	If Brep representation is used
Extruded	Nominal (Binary)	Metadata	If swept solid representation is used
Clipping	Nominal (Binary)	Metadata	If clipping representation is used
CSG	Nominal (Binary)	Metadata	If CSG representation is used
SurfaceModel	Nominal (Binary)	Metadata	If SurfaceModel representation is used
ExtrudX	Numerical	Geometric	
ExtrudY	Numerical	Geometric	Extruded direction
ExtrudZ	Numerical	Geometric	
Rec	Nominal (Binary)	Metadata	If a rectangular shape is used
Cir	Nominal (Binary)	Metadata	If a cylinder shape is used
Ring	Nominal (Binary)	Metadata	If a ring-shape is used
I	Nominal (Binary)	Metadata	If an I-shape is used
Type	Nominal (Quinary)	Ground Truth	Labeled type

Table 3. Sample invariant signature values by instances.

Instance	Model	***Rec_L***	***Rec_W***	***Height***	***Cir_R***	***AveVerti***	***Type***
IfcFooting1	Duplex_A	18.28	0.9	0.3	0	0	Footing
IfcFooting2	Duplex_A	8.38	0.9	0.3	0	0	Footing
IfcFooting3	Duplex_A	17.38	0.9	0.3	0	0	Footing
IfcColumn1	Rst_advanced	0	0	3.5	0.225	0	Column

IfcColumn2	Rst_advanced	0	0	3.5	0.225	0	Column
IfcColumn3	Rst_advanced	0	0	3.5	0.225	0	Column
IfcBeam125	Rst_Basic	0	0	1.44	0.015	0	Beam
IfcBeam126	Rst_Basic	0	0	2.88	0	3.52	Beam
IfcBeam127	Rst_Basic	0	0	3.13	0	3.52	Beam

Table 4. Classification results of best configuration of neural networks.

Category	**Number of objects**	**Number of correctly classified objects**	**Number of objects classified into**	**Recall**	**Precision**	**F1**	**ROC**
Beam	549	548	551	99.8%	99.5%	99.6%	100.0%
Column	286	284	286	99.3%	99.3%	99.3%	100.0%
Footing	250	249	255	99.6%	97.6%	98.3%	99.8%
Slab	53	45	49	84.9%	91.8%	88.2%	99.0%
Wall	192	185	189	96.4%	97.9%	97.1%	99.6%
Total	*1,330*	*1,310*	*1,330*	*98.6%*	*98.6%*	*98.6%*	*99.9%*

Table 5. Classification results of the best configuration of decision table.

Category	**Number of objects**	**Number of correctly classified objects**	**Number of correctly classified objects**	**Recall**	**Precision**	**F1**	**ROC**
Beam	549	544	550	99.1%	98.9%	99.0%	100.0%
Column	286	277	280	96.9%	98.9%	98.2%	99.3%
Footing	250	248	260	99.2%	95.4%	97.3%	99.7%
Slab	53	47	48	88.7%	97.9%	93.1%	98.4%
Wall	192	184	192	95.8%	95.8%	95.8%	99.3%
Total	*1,330*	*1,300*	*1,330*	*97.7%*	*97.7%*	*97.7%*	*99.6%*

Table 6. Classification results of the best configuration of Bayesian network.

Category	**Number of objects**	**Number of correctly classified objects**	**Number of objects classified into**	**Recall**	**Precision**	**F1**	**ROC**
Beam	549	547	549	99.6%	99.6%	99.6%	100.0%
Column	286	285	285	99.7%	100.0%	99.8%	100.0%
Footing	250	247	241	98.8%	98.4%	98.6%	100.0%
Slab	53	44	49	83.0%	89.8%	87.3%	99.8%
Wall	192	181	186	99.5%	97.4%	98.5%	100.0%
Total	*1,330*	*1,300*	*1,330*	*98.8%*	*98.8%*	*98.8%*	*100.0%*

Table 7. Classification results of the best configuration of SVM.

Category	Number of objects	Number of correctly classified objects	Number of objects classified into	Recall	Precision	F1	ROC
Beam	549	547	551	99.6%	99.3%	99.5%	99.6%
Column	286	284	286	99.3%	99.3%	99.3%	99.8%
Footing	250	250	256	100.0%	97.7%	98.8%	99.7%
Slab	53	44	46	83.0%	95.7%	88.9%	97.5%
Wall	192	186	191	96.9%	97.4%	97.1%	99.0%
Total	*1,330*	*1,300*	*1,330*	*98.6%*	*98.6%*	*98.6%*	*99.5%*

Table 8. Classification results of the best configuration of random forest.

Category	Number of objects	Number of correctly classified objects	Number of objects classified into	Recall	Precision	F1	ROC
Beam	549	549	551	100%	99.6%	99.8 %	100%
Column	286	285	285	99.7%	100.0%	99.8%	100.0%
Footing	250	250	254	100%	98.4%	99.2%	100.0%
Slab	53	47	48	88.7%	97.9%	93.1%	99.8%
Wall	192	191	191	99.5%	99.5%	99.5%	100.0%
Total	*1,330*	*1,300*	*1,330*	*99.4%*	*99.4%*	*99.4%*	*99.9%*

Table 9. Testing performance of the selected machine learning algorithm - the random forest.

Category	Number of objects	Number of correctly classified objects	Number of objects classified into	Recall	Precision	F1	ROC
Beam	241	241	241	100.0%	100.0%	100.0%	100.0%
Column	126	126	126	100.0%	100.0%	100.0%	100.0%
Footing	104	104	105	100.0%	99.0%	99.5%	100.0%
Slab	26	24	24	92.3%	100.0%	96.0%	99.1%
Wall	73	73	74	100.0%	98.6%	99.3%	100.0%
Total	*570*	*568*	*570*	*99.6%*	*99.7%*	*99.6%*	*99.9%*

Table 10. Error analysis of the best machine learning algorithm - the random forest.

Instance	Real type	Classified type	Comment/ analysis
IfcSlab1	Slab	Footing	Voting trees: 5 footings, 3 slabs, 1 column, 1 wall
IfcSlab3	Slab	Wall	Voting trees: 5 walls, 2 footings, 2 slabs, 1 column

Table 11. Performance comparison with the state of the art

	Koo et al.'s Features	Authors' Features (Invariant Signatures)
SVM	94.86%	98.65%
Random Forest	98.87%	99.40%

Table 12. Example quantity takeoff results

Model		Volume Using Invariant Signatures (m^3)	Volume Using Commercial Software (Gold Standard) (m^3)	Difference
Wall 1	Two-bedroom unit	4.221	4.221	0.0%
Wall 2	Two-bedroom unit	0.409	0.410	0.3%
Wall 1	Four-bedroom unit	9.855	9.855	0.0%
Wall 2	Four-bedroom unit	0.212	0.212	0.0%

Figure Captions

Fig. 1. Distribution of the locational signature *O3* (elevation) on different object types.

Fig. 2. Distribution of the geometric signature *Length* on different object types.

Fig. 3. Distribution of the geometric signature *Cir_R* on different object types.

Fig. 4. Visualization of training results of deep learning on different configurations.

Fig. 5. Visualization of training results of decision table on different configurations.

Fig. 6. Visualization of training results of Bayesian network on different configurations.

Fig. 7. Visualization of training results of SVM using different configurations.

Fig. 8. Visualization of training results of random forest in different configurations.

Fig. 9a. Visualization of the 2-bedroom and 4-bedroom units used for quantity takeoff.

Fig. 9b. Visualization of the student apartment complex used for quantity takeoff.

Fig. 10. Visualization of the best-performing machine learning model (random forest).

Appendices

Appendix 1. Neural network accuracy using different layers and different number of nodes.

Layers\Nodes per Layer	10	20	30	35	40	45	50
1	98.05%	98.20%	98.12%	97.89%	98.20%	98.02%	97.97%
2	97.74%	98.05%	98.12%	98.05%	98.35%	98.57%	98.20%
3	96.91%	97.97%	97.67%	97.52%	97.97%	97.29%	97.00%

Appendix 2. Decision table accuracy using different search direction and depth.

Forward Depth	Accuracy	Backward Depth	Accuracy	Bidirectional Depth	Accuracy
1	97.44%	1	97.74%	1	97.44%
2	97.44%	2	97.74%	2	97.44%
3	97.44%	3	97.74%	3	97.59%
4	97.44%	4	97.74%	4	97.59%
5	97.44%	5	97.74%	5	97.59%
10	97.44%	10	97.74%	10	97.59%

Appendix 3. Accuracy vs. number of parents of Bayesian network.

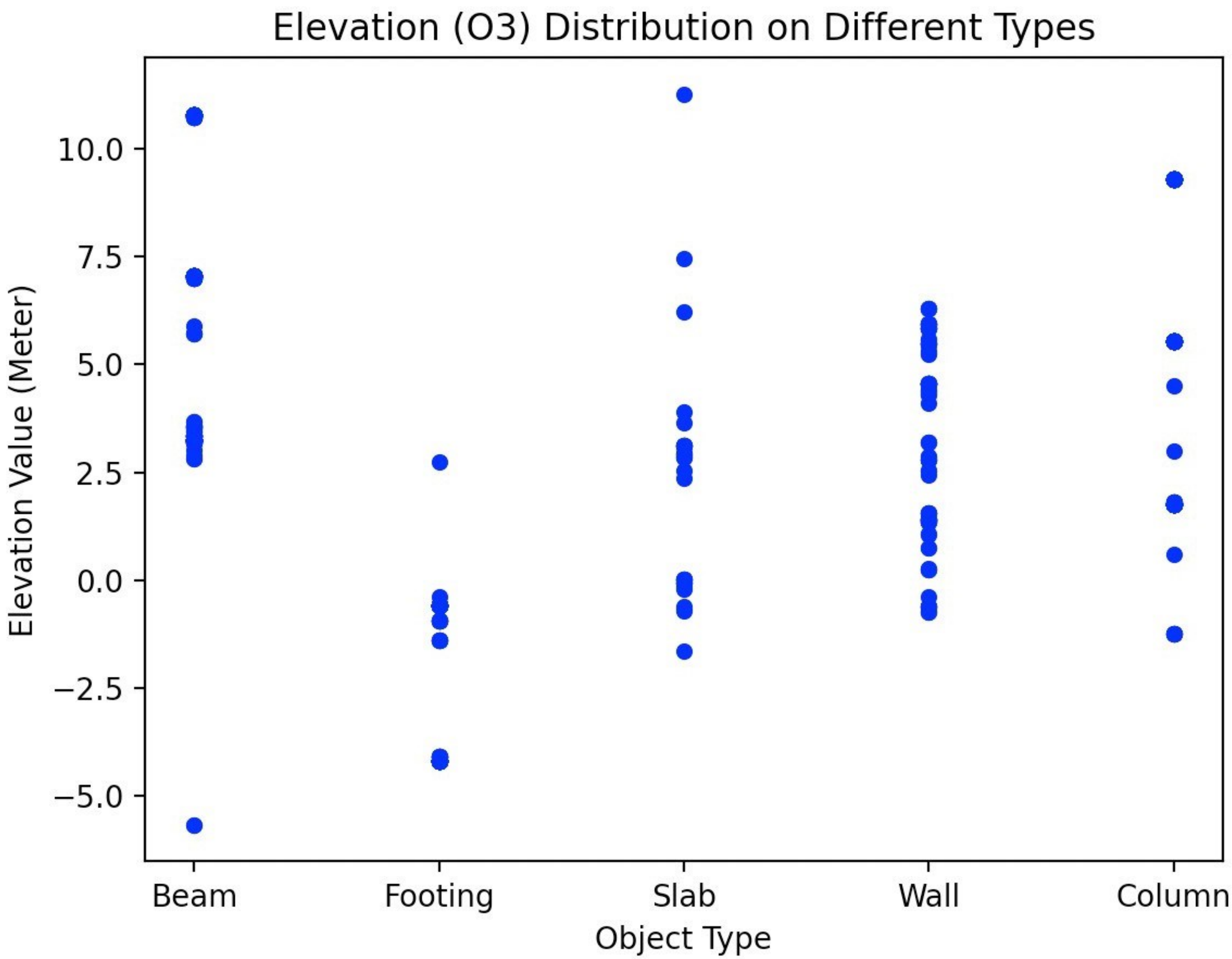
No. of Parents	Accuracy
1	96.62%
2	98.80%
3	98.80%
4	98.80%
5	98.80%
6 - 10	98.80%

Appendix 4. Accuracy on different regulation terms C for SVM.

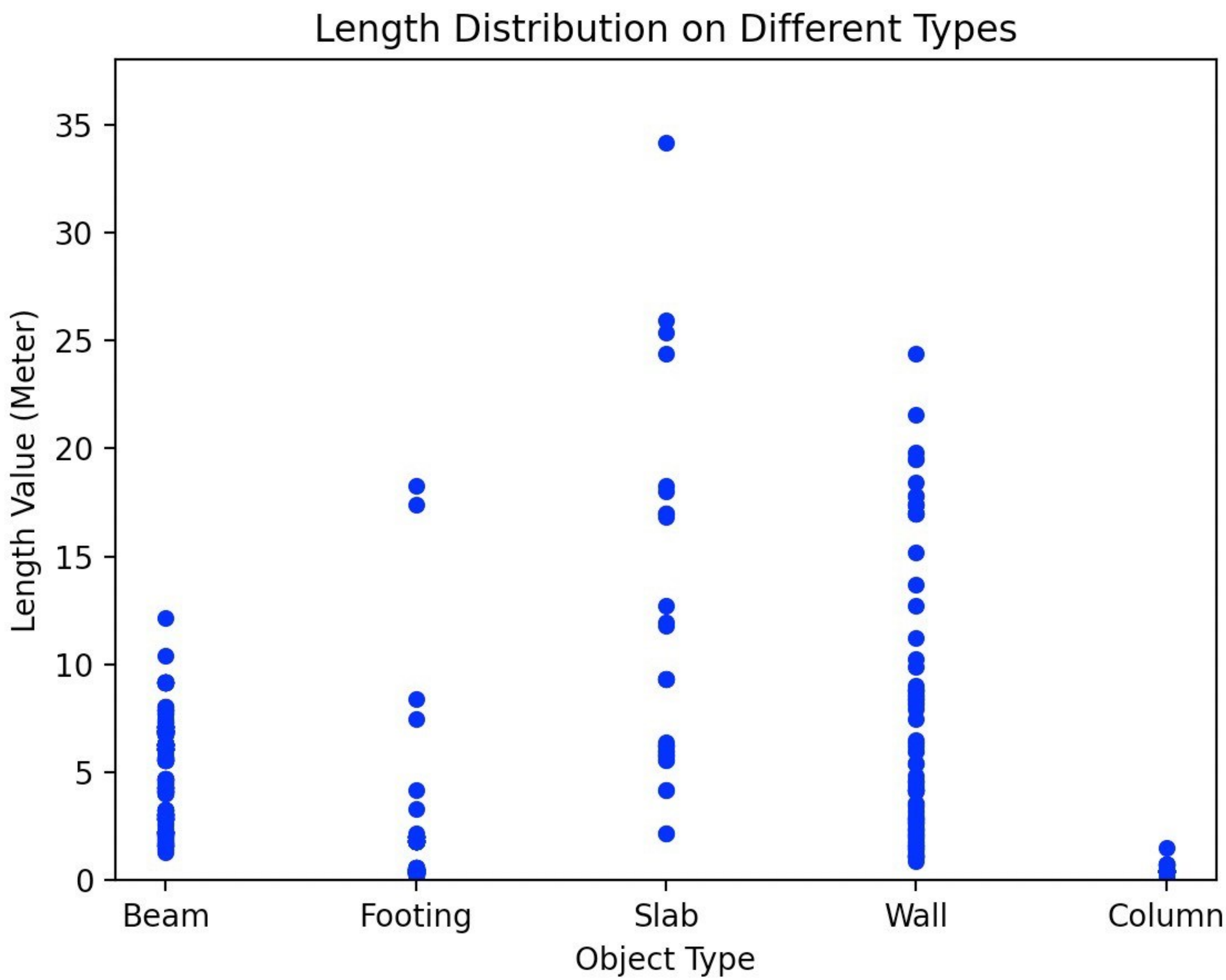
C	Accuracy	C	Accuracy	C	Accuracy	C	Accuracy
1	97.44%	9	98.50%	10	98.57%	14.2	98.65%
5	97.97%	10	98.57%	14	98.65%	14.25	98.72%
10	98.57%	20	98.72%	15	98.72%	14.5	98.72%
50	98.57%	30	98.57%	16	98.72%	14.75	98.72%
100	98.57%	40	98.57%	17	98.65%	15	98.72%
1000	98.20%	50	98.57%	18	98.65%	15.25	98.72%
10000	98.05%	60	98.50%	19	98.65%	15.5	98.72%
		80	98.50%	20	98.72%	15.75	98.72%
		100	98.57%	25	98.65%	16	98.72%
		105	98.50			16.25	98.65%

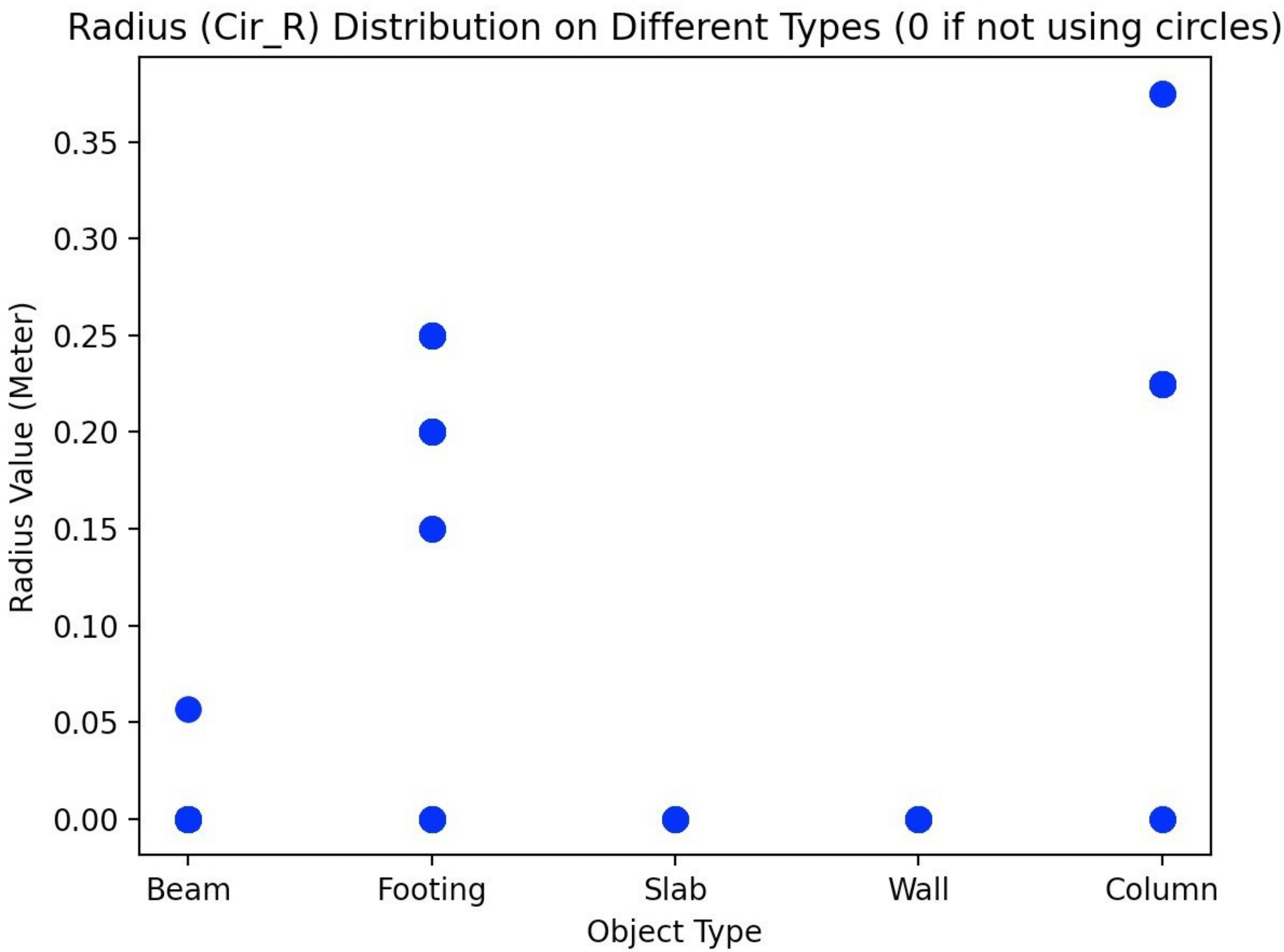
Appendix 5. Accuracy of random forest on different number of parents.

Parents	**Accuracy**
1	60.08%
2	88.72%
3	94.59%
4	97.97%
5	99.17%
6	99.25%
7	**99.40%**
8	99.24%
9	99.17%
10	99.17%
11	99.17%
12	99.25%
13	99.25%
14	99.25%
15	99.25%
20	99.25%
50	99.25%
Unlimited	99.25%



Elevation (O3) Distribution on Different Types
Elevation Value (Meter)
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
Beam
Footing
Slab
Wall
Column
Object Type


Length Distribution on Different Types
Length Value (Meter)
35
30
25
20
15
10
5
0
Beam
Footing
Slab
Wall
Column
Object Type


Radius (Cir_R) Distribution on Different Types (0 if not using circles)
Radius Value (Meter)
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
Beam
Footing
Slab
Wall
Column
Object Type

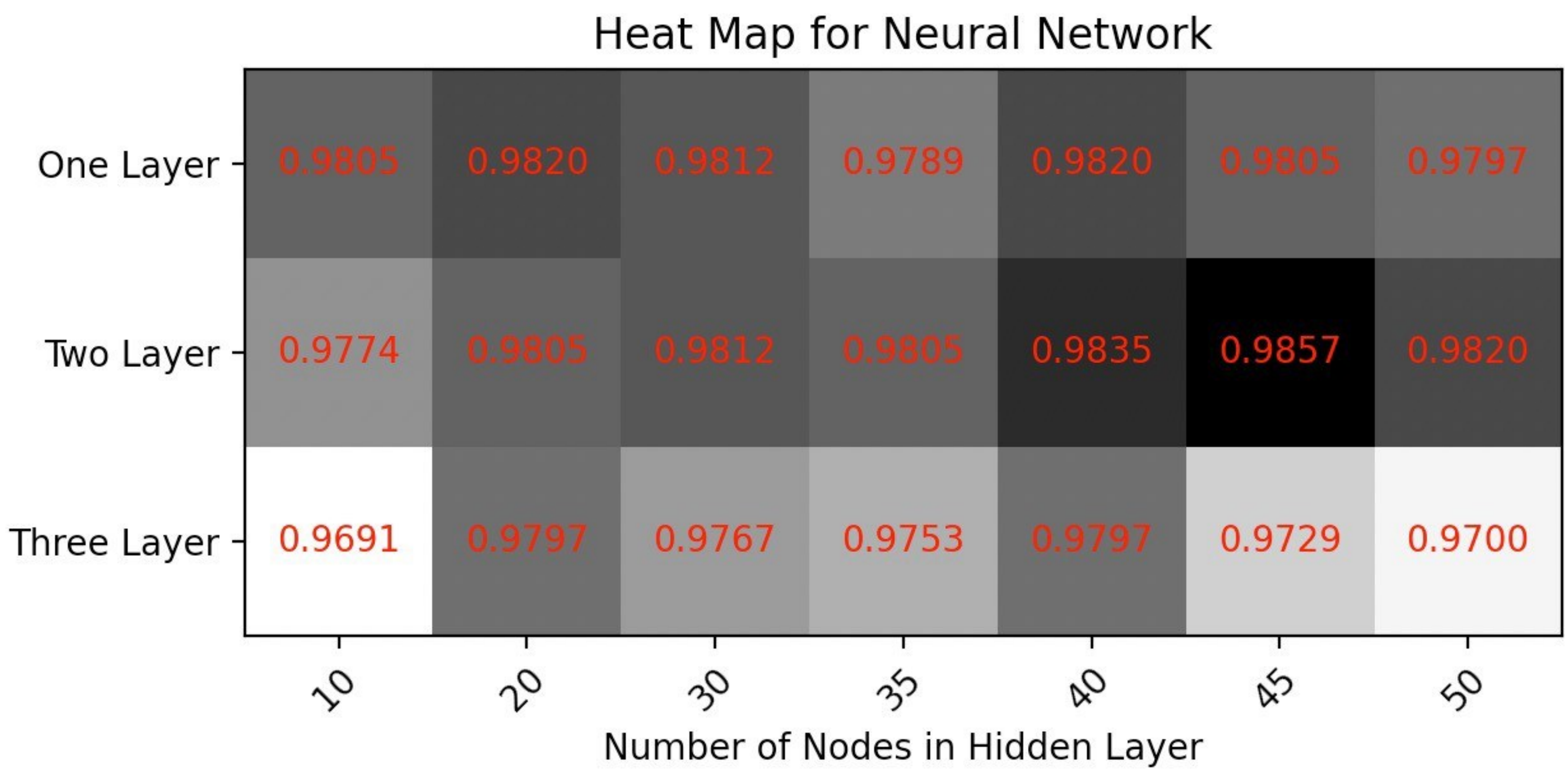

Heat Map for Neural Network
One Layer
Two Layer
Three Layer
0.9805
0.9820
0.9812
0.9789
0.9820
0.9805
0.9797
0.9774
0.9805
0.9812
0.9805
0.9835
0.9857
0.9820
0.9691
0.9797
0.9767
0.9753
0.9797
0.9729
0.9700
10
20
30
35
40
45
50
Number of Nodes in Hidden Layer

Figure 5

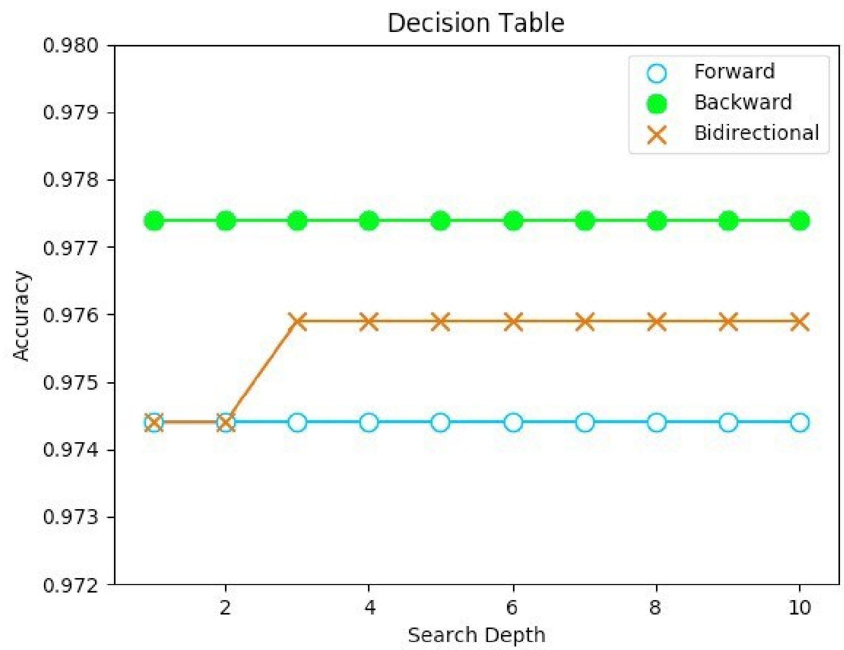

Decision Table
0.980
0.979
0.978
0.977
0.976
0.975
0.974
0.973
0.972
Accuracy
2
4
6
8
10
Search Depth
Forward
Backward
Bidirectional

Figure 6

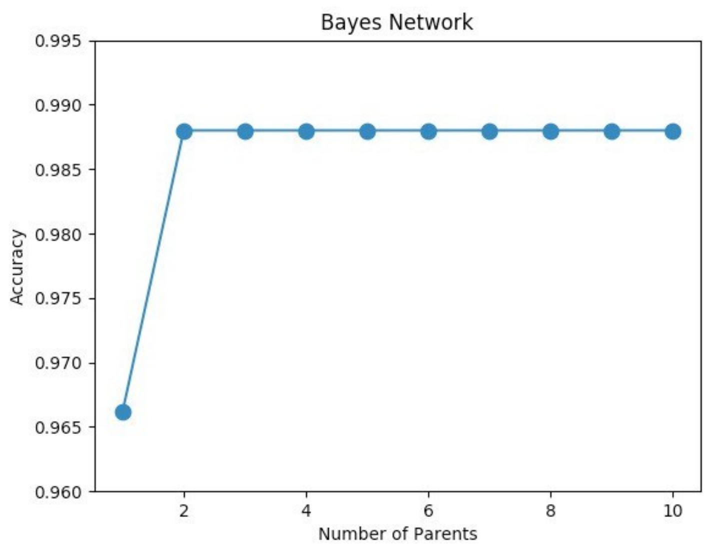
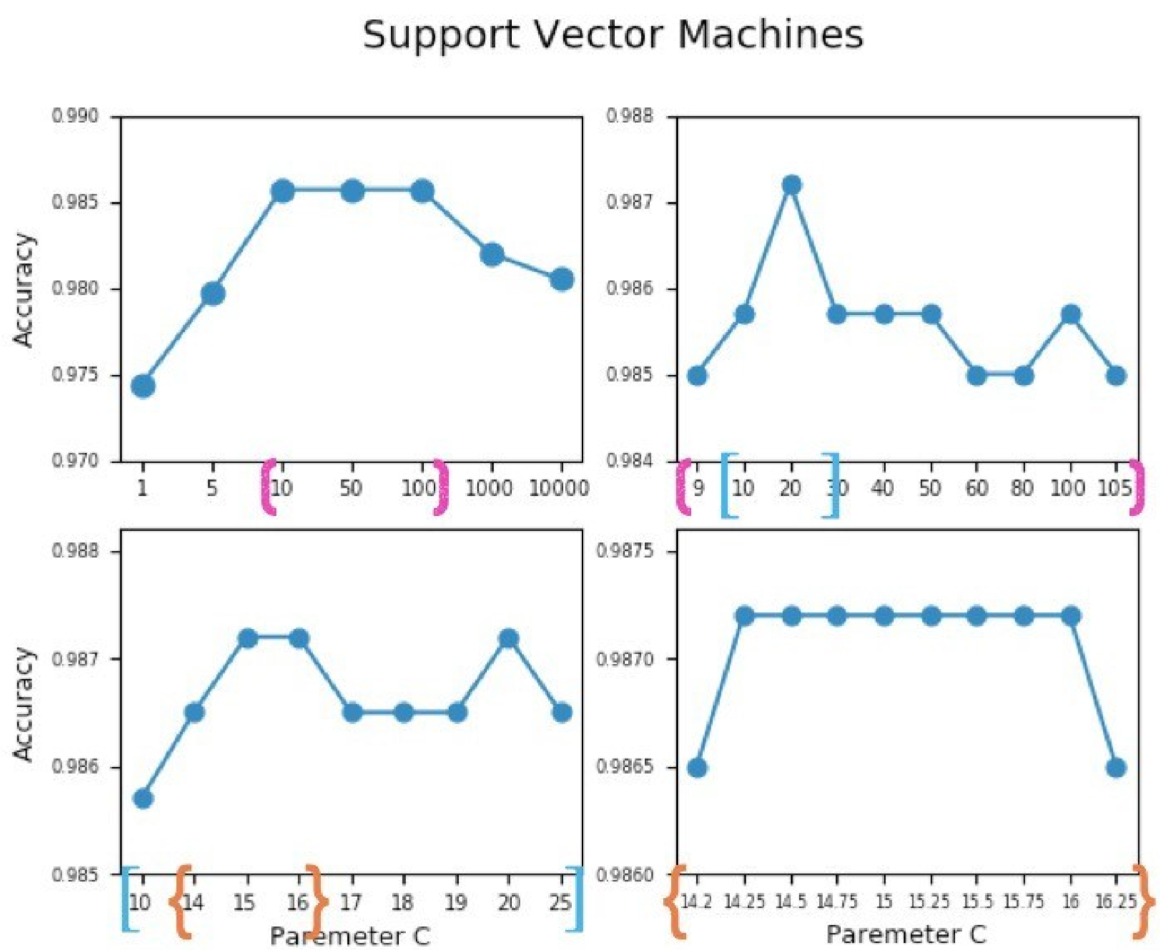
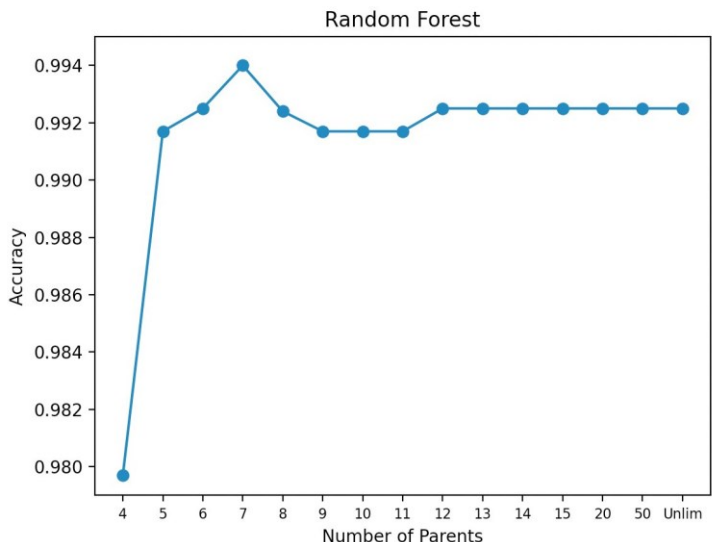

Bayes Network
0.995
0.990
0.985
0.980
0.975
0.970
0.965
0.960
Accuracy
2
4
6
8
10
Number of Parents

Figure 7


Support Vector Machines
Accuracy
Paremeter C
0.990
0.985
0.980
0.975
0.970
1
5
10
50
100
1000
10000
0.988
0.987
0.986
0.985
0.984
9
10
20
30
40
50
60
80
100
105
0.988
0.987
0.986
0.985
10
14
15
16
17
18
19
20
25
0.9875
0.9870
0.9865
0.9860
14.2
14.25
14.5
14.75
15
15.25
15.5
15.75
16
16.25


Random Forest
Accuracy
0.994
0.992
0.990
0.988
0.986
0.984
0.982
0.980
4
5
6
7
8
9
10
11
12
13
14
15
20
50
Unlim
Number of Parents

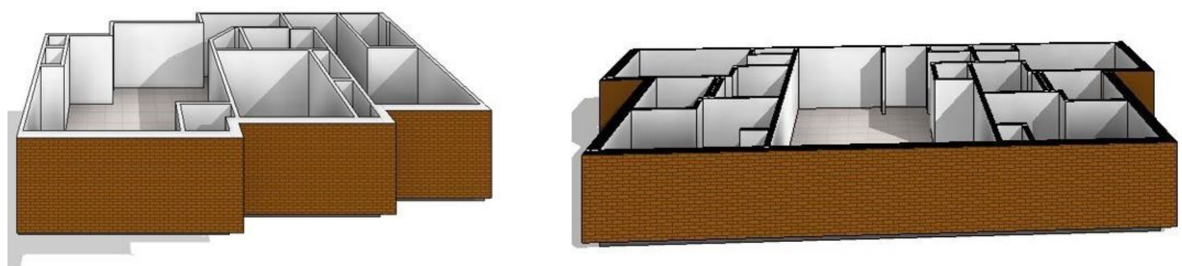


Figure 9b

Figure 10


F5 < 1.5
Yes
No
mRatio < 3.9
• • •
Yes
No
Rec_L < 0.73
• • •
Yes
No
• • •
(4 more layers)
• • •
Yes
No
Rec_H < 5.014
• • •
Yes
No
Column (5/0)
Beam (15/0)
8
More
Trees
• • •
Cir_R < 0.1035
Yes
No
mRatio < 0.37
• • •
Yes
No
Rec_W < 0.3515
• • •
Yes
No
• • •
(4 more layers)
• • •
Yes
No
mRatio < 0.18
• • •
Yes
No
Column (8/2)
Beam (15/3)