Quality check of BIM models using machine learning

https://doi.org/10.21814/uminho.ed.164.2

Iraj Esmaeili[1], **João Poças Martins**[2], **José Miguel Castro**[3]

[1] *PhD Student, Department of Civil Engineering, Faculty of Engineering of the University of Porto, Porto, 0000-0002-4819-0312*

[2] *CONSTRUCT-FEUP, BUILT CoLAB, Porto, 0000-0001-9878-3792*

[3] *Associate Professor, Department of Civil Engineering, Faculty of Engineering of the University of Porto, Porto, 0000-0001-9732-9969*

Abstract

The complexity of BIM models challenges the engaged parties to deliver an accurate model suitable for various purposes. This is especially important during the construction stage, where errors in construction drawings entail considerable cost and time burdens. As a possible solution, artificial intelligence and machine learning (ML) techniques can be deployed to assist BIM parties with the time and resource-consuming task of checking the quality of BIM models. This study aims to use machine learning techniques to check the quality of BIM models, especially in precast structural wall openings. A machine learning model was used in a BIM model of a project to detect anomalies in openings of precast structural walls, and it was able to detect all the openings with wrong information, which, consequently, would negatively impact the final delivery of the walls. Considering the applicability of using such an ML model in other projects, the contribution of this study is to reduce the errors in the construction drawings and consequently secure the projects in terms of time and cost burdens due to these errors.

1. Introduction

As the construction industry moves towards digitalization and adoption of Building Information Modelling (BIM), ensuring the quality of BIM models becomes relevant. Delivering the project within the planned time, budget, and quality is tightly connected to the drawings issued for construction. Since construction drawings are produced from BIM models, missing and incorrect information in BIM models leads to errors in later phases. Therefore, a sound BIM model will produce constructible drawings with fewer errors.

Missing and incorrect information can hinder the automation of tasks and jeopardize the quality of construction output. Due to the large variability of geometries and objects in BIM models, the data embedded in the models cannot be automatically verified by setting explicit rules [1]; therefore, artificial intelligence (AI) and specifically machine learning techniques can replace the need for hardcoding the rules. Moreover, rule inference is itself a specific and constrained instantiation of AI [2].

The field of artificial intelligence is a thriving field that has numerous practical applications. The ability of AI systems to learn from data alleviated the difficulties encountered by systems that rely on hard-coded knowledge [3]. Machine learning algorithms offer solutions in several areas that need prediction, classification, clustering, and anomaly detection. Therefore, manual or rule-based data verification for anomaly detection can be replaced by an automated machine learning process.

As BIM models are growing in size and complexity, a human-performed quality check, even on a specific object class, might be impossible within the strict deadlines of projects. Hence, this study proposes a method for BIM model quality checking for openings where mechanical, electrical, and plumbing (MEP) services pass through them in the walls, floors, and ceilings of buildings. A machine learning model was applied to identify errors and omissions in the data embedded in opening elements.

The structure of this study is organized as follows: first, a background on BIM and machine learning studies is provided in Section 2. Next, in Section 3, the research method is discussed. Then, the experiments and the results are reported in Section 4. Finally, the conclusions and future work are presented in Section 5.

2. Background

Various studies peeked into the evaluation of BIM models and the quality of data in the models. Detecting abnormal data in BIM models [1], classification of room types and semantic enrichment [2], detection of anomalies in mapping BIM to IFC (industry foundation classes) [4], and code compliance checking and semantic enrichment [5] are among the BIM quality checking studies conducted so far.

On the other hand, using machine learning with BIM is increasingly common. In a study by [6], a BIM and machine learning integration framework was developed to

automate real-state property valuation. A Support Vector Machines (SVM) model was proposed in [7] to classify heritage building objects such as floors, ceilings, roofs, beams, and walls given a point cloud. In facility maintenance management (FFM), machine learning algorithms were used in conjunction with BIM and Internet of Things (IoT) devices to predict the future condition of MEP elements [8]. Another study [9] focused on integrating BIM and machine learning in railway systems to localize defects in railway infrastructure. Using new technologies such as image processing, machine learning, and virtual reality (VR) along with BIM to automate construction project simulation [10] is another example of how machine learning is applied with BIM.

The use of machine learning in the quality assessment of BIM models and the embedded information is not rare either. Several studies ([1], [2], [4], [5]) benefited from machine learning in their BIM quality checking research. Therefore, this study is also motivated to explore a new aspect of BIM model quality checking through the application of machine learning models. Machine learning models were deployed in the BIM openings model where each opening in the model represents a void space in architectural or structural elements of the building. Elements in the BIM opening model have specific parameters whose values are related to the wall's type (precast, gypsum, masonry, among others) in which the opening resides. The goal of this study is to verify these data values for openings in the BIM model.

3. Research method

3.1. Problem statement

Building upon the existing gaps, this study started with a literature review. Due to the complexity of the BIM models and limitations of the rule-based methods for BIM quality checking in terms of scalability and interoperability, a method to use machine learning techniques was adopted. The federated BIM model consists of several separate models from different disciplines, such as architectural, structural, and mechanical. Since each of these models might have been developed in a different BIM authoring software, interoperability issues may arise. This is where the main authoring software might fail to recognize the correct categories of elements from other BIM authoring software or detect the clashes between elements of models from different authoring software. Therefore, machine learning models were used instead of rule-based scenarios to avoid such issues. Two machine learning models were developed and implemented in the context of a BIM project, and finally, the results of the experiments and the performance of each machine learning model were reported. In the following, the pipeline for gathering data, preprocessing data, developing the machine learning models, and finding the best hyperparameters for the models are discussed in detail.

3.2. Data gathering from BIM model

A sample federated BIM model consisting of architecture, structures, and MEP disciplines with a model for openings was adopted. The BIM model for openings is a set of rectangular or circular elements representing the void spaces where the MEP services pass through the architectural or structural elements, i.e., walls, slabs, and floors. Having the BIM model, a Python script was developed using Revit's API to retrieve all the openings in the model and their location and geometry features. These features consist of the location of the opening in space, its rotation, facing orientation, hand orientation, and transformation. Therefore, each sample will have three features for location, one rotation, three facing orientations, three hand orientations, and nine transformations (three for each basis).

Since these features are location-based, the openings' data are independent of the relationship between the BIM opening model and BIM models of other disciplines. This means that, for example, there is no need to see if the opening is clashing with a wall and then check the type of the wall to associate the correct parameter value for the opening based on the wall type; hence, a rule-based quality check. This rule-based approach might be the case when all BIM models constituting the federated BIM model are developed in the same authoring software. However, this approach fails when the BIM models of different disciplines are developed in different authoring software, which is the case in many situations. In addition, the rules need to be tailored for every authoring software due to their intrinsic differences. With the method described in this paper, the dependency on the authoring software, and consequently interoperability issues, is eliminated.

Finally, each opening has a specific parameter, such as the "Workset" parameter used in this study, that indicates the type of wall in which the opening resides. This feature needs to be verified in terms of the correct value. In case of failure to meet this verification, errors will occur in construction drawings.

3.3. Data preprocessing and machine learning model development

Processing data before feeding it to the machine learning model is beneficial and sometimes necessary. The data contains both numerical and categorical values. In addition, the numerical values have different ranges; hence, they need to be scaled. This task was done by creating a Standard Scaler as a numerical preprocessor to transform numerical data by removing the mean and scaling to unit variance. For categorical features, such as Workset, One-Hot encoding was implemented to create binary columns for each category to represent the presence or absence of that category with a value of 1 or 0, respectively. For example, if Workset has two possible values A and B, One-Hot encoding will create two separate columns for Workset A and Workset B. Whenever the element's Workset is A, it will assign a value of 1 for Workset A and a value of 0 for Workset B and vice versa.

After preprocessing the data, a K-Means model was adopted to cluster the openings with similar locations and geometric features. The K-Means algorithm is an unsupervised machine learning algorithm used for clustering. The K-Means algorithm partitions the data into *k* clusters, and each data point is assigned to the nearest cluster based on its distance to the mean of clusters. The parameter *k* needs to be defined by the user. Generally, determining an appropriate value for *k* might be challenging, and selecting an incorrect value could lead to unfavourable results. A silhouette score, which is the mean silhouette coefficient over all instances, can be used as a more precise alternative. The Silhouette coefficient for an instance is calculated by Equation (1).

$$silhouette\ coefficient = \frac{b - a}{\max(a, b)} \tag{1}$$

In Equation (1), *a* is the mean distance to the other instances in the same cluster, and *b* is the mean distance to the instances of the next closest cluster. The silhouette coefficient can vary between -1 and +1, where a value close to +1 means that the instance is in the right cluster. A value close to -1 means that the instance is in the wrong cluster, and a value close to 0 means that the instance is close to a cluster boundary [11].

Following the determination of the optimal *k*, the K-Means algorithm was developed considering the optimal number of clusters. Next, the opening elements that were collected and preprocessed in previous steps were clustered using the K-Means algorithm. As mentioned earlier, the purpose is to detect the openings that do not have the correct value for the Workset parameter. Then, the mode with respect to the Workset for each cluster was calculated. Finally, anomalies were detected where the Workset value for each element was different from the cluster's mode. As shown in Figure 1 there are openings on structural walls and architectural walls. The openings residing on each wall type must have the proper Workset value, i.e., "Precast-Wall" for the openings on the structural walls, "Gypsum-Wall" for the openings on the architectural walls, and so on. Failing to meet this criterion means that the opening with incorrect information will not appear on the respective drawings and consequently, a mistake in construction will happen that will have negative cost and time impacts.

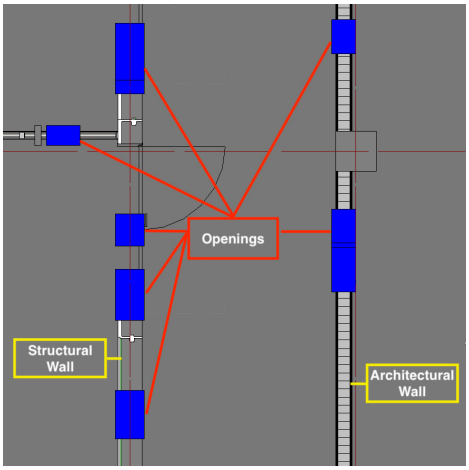

Figure 1
An example of wall types and openings.

4. Results

Following the collection of opening elements in the project, the aforementioned features (location, orientation, Workset, and so on) were retrieved for each opening. Next, the data was preprocessed to transform numerical data and categorical data. The output of data preprocessing served as the main dataset to feed the machine learning model. Next, the K-Means model was implemented to cluster the openings and detect anomalies. In this scenario, an opening was modelled on the structural wall with an incorrect Workset as a "Gypsum-Wall" (see Figure 2).

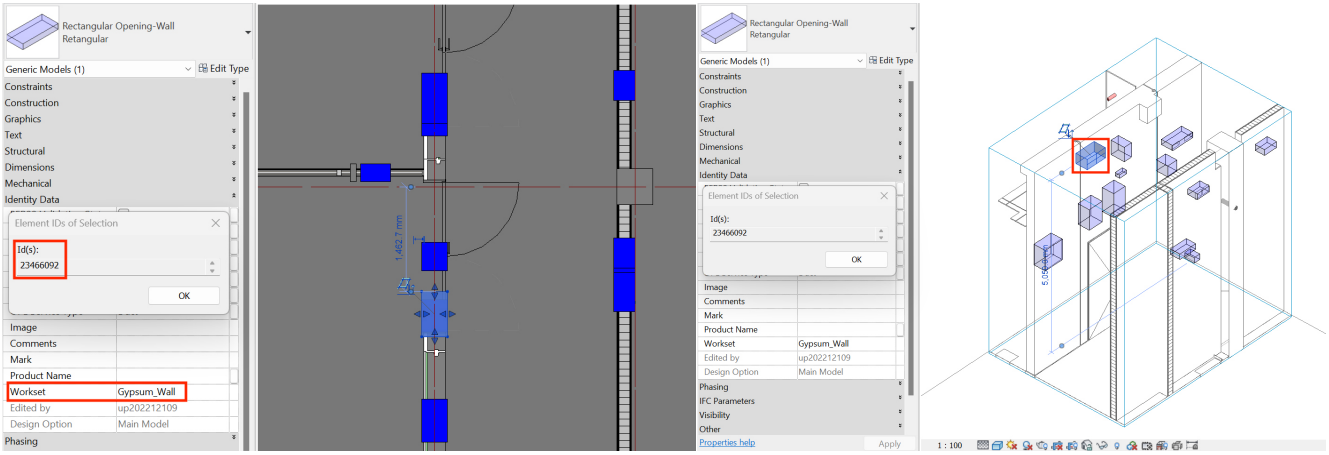

Figure 2
The anomaly opening in 2D and 3D views with its ID.

The model was able to cluster the openings and correctly detect the opening with the incorrect information. As shown in Figure 3, the model was able to cluster the openings properly. In Figure 3, openings with the same colour belong to the same cluster.

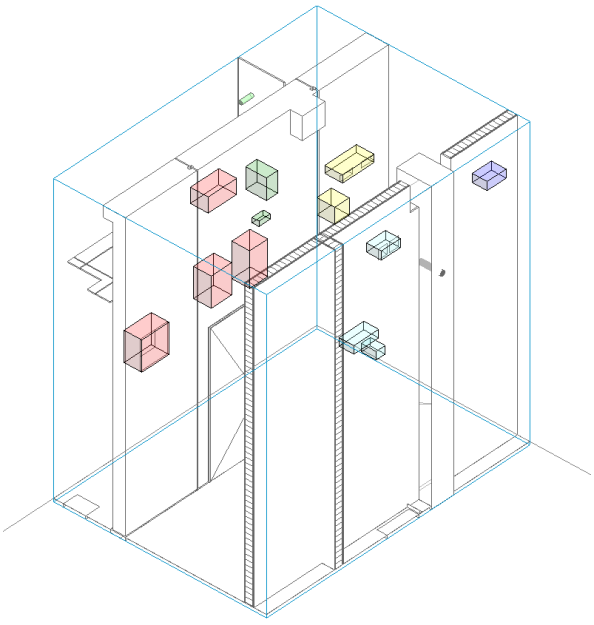

Figure 3
Clustering results in a 3D view

A summary of the results is shown in Figure 4. As reported, the model was able to detect the opening with incorrect information and report its ID. This scenario had 13 openings, and the model grouped them into 5 clusters. The processes of finding the optimal number of clusters and clustering the openings took less than 1 second on a MacBook Pro with an Apple M1 Pro chip and 16 GB memory.

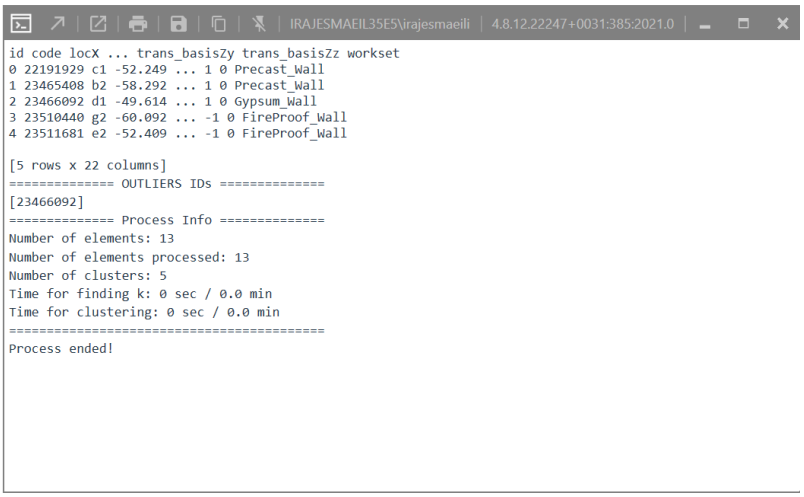

Figure 4
A summary of the process with the anomalies detected by the K-Means model.

Following the successful experiment with a small-scale project, the K-Means model was implemented in a project with 170 openings. This project contained three openings with incorrect information. A total number of 15 random initializations was used to find the best *k*, and 10 random initializations were used for the K-Means model at the clustering step. Using the silhouette method for finding the optimal *k*, a total number of 54 clusters was suggested. The K-Means model was implemented with these parameters as the model's hyperparameters. Although the model was able to detect the three anomalies, it incorrectly included nine more openings as anomalies. As depicted in Figure 5, this incorrect detection is because the K-Means model puts the openings on the two facing walls in the same cluster, which is incorrect. The opening tags in Figure 5 mean that, in total, four openings belong to cluster number 25.

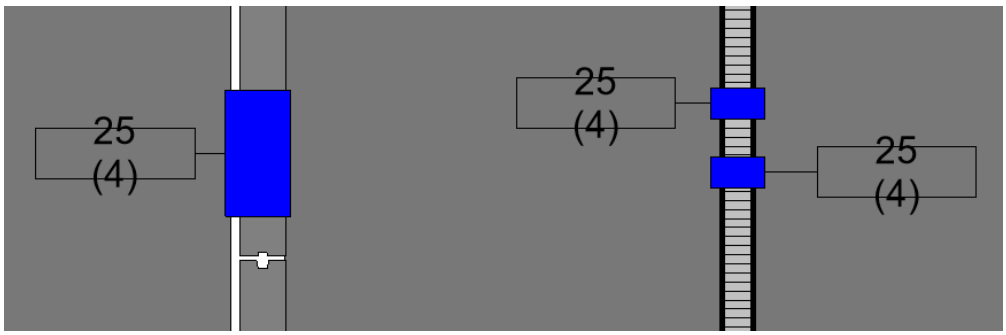

Figure 5
Example of undesirable clustering by the K-Means model.

A possible reason for this behaviour of the K-Means model could be that the data is not isotropic, i.e., the data is not evenly distributed in all directions. The openings reside along the walls in *x* or *y* directions, so from a location point of view, the data has elongated shapes. The performance of the K-Means models can diminish when working with elongated data; therefore, this incorrect clustering could be due to non-isotropic data.

As an alternative, a Gaussian Mixture Model (GMM) was deployed. A GMM is a probabilistic model, and the underlying assumption is that the instances were generated from a mixture of several Gaussian distributions [11]. In its simplest form, the number of *k* of Gaussian distributions must be known in advance. Using the same process of finding the optimal *k* as the K-Means model and the same number of random initializations, a GMM model was developed. Following the implementation of the GMM model, the results showed that the model could properly cluster the openings and detect all three anomalies (see Figure 6).

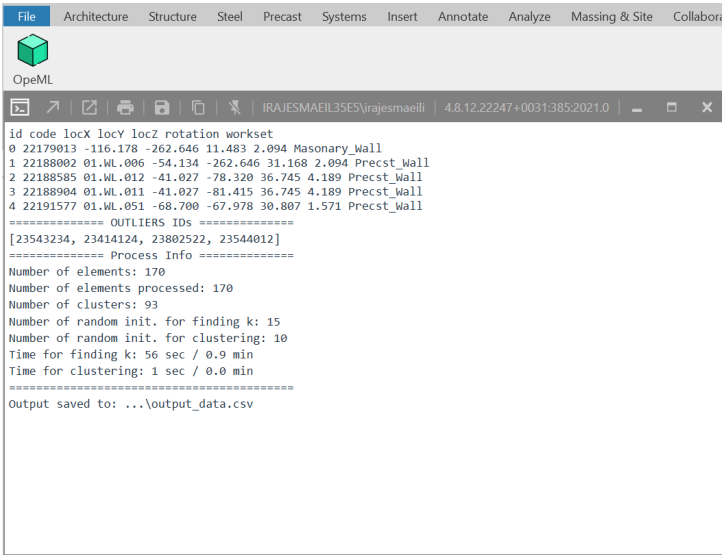

Figure 6
A summary of the process with the anomalies detected by the GMM model.

The model only returned one incorrect opening as an anomaly, which is much less than the K-Means model. A probable reason for this kind of misdetection could be the existence of clusters with only one or two elements. When the clusters have one or two elements, the mode of the cluster is the element itself in case of one element, or the mode can be both elements in case of two elements with different Worksets. This can be avoided by assigning these elements to the next nearest clusters.

As depicted in Figure 6, with the hardware configuration used for running the machine learning models, the time for clustering is almost negligible, even for a higher number of random initializations. However, finding the optimal *k* using the silhouette score took almost 1 minute for the model with 170 elements. For projects with hundreds or a few thousands of openings, it will take longer to come up with a good number for *k* since this approach is computationally expensive.

5. Conclusions

As the construction sector progresses towards digital transformation and the integration of BIM, ensuring the quality of BIM models becomes relevant. Missing and incorrect information integrated with the BIM models in the design phase can lead to errors in subsequent phases. Several studies in the past have peeked into the quality assurance of BIM models, and a number of them have tried to use machine learning to serve their purpose. This study considers a new aspect of BIM model quality checking in the area of data verification for MEP services' openings. As a new approach, we deployed two machine learning models, K-Means and Gaussian Mixture Models, and the results showed promising results in the proper detection of incorrect information embedded in the BIM models.

This study contributes to the body of knowledge by putting forward a new aspect of BIM data verification and exploring the application of machine learning techniques in this area. Equally important, this study contributes to the practice by reducing the errors in the BIM models and consequently securing the projects from time and cost burdens due to these errors.

This study tried to break free from the dependency on specific BIM authoring software by using location data of the BIM elements and machine learning models. This method can be insightful for future studies in the field of BIM quality checking and encourage researchers to consider the benefits of using unsupervised machine learning techniques in their research. In addition, since finding the optimal number of clusters might be computationally expensive in large models, future work can elaborate on methods that are less time-consuming yet efficient for finding a good number of clusters for unsupervised machine learning models.

Acknowledgments

This work has been developed within the scope of "R2U Technologies - modular systems" project, contract C644876810-00000019, funded by the Recovery and Resilience Plan (PRR) and by the European Union - NextGeneration EU.

This work is also integrated in the R&D activities of the CONSTRUCT Institute on Structures and Constructions, financially supported by Base Funding UIDB/04708/2020 through national funds of FCT/MCTES (PIDDAC).

References

[1] M. Xiao, Z. Chao, R. F. Coelho, and S. Tian, "Investigation of Classification and Anomalies Based on Machine Learning Methods Applied to Large Scale Building Information Modeling," *Applied Sciences (Switzerland)*, vol. 12, no. 13, Jul. 2022, doi: 10.3390/APP12136382.

[2] T. Bloch and R. Sacks, "Comparing machine learning and rule-based inferencing for semantic enrichment of BIM models," *Autom Constr*, vol. 91, pp. 256-272, Jul. 2018, doi: 10.1016/J.AUTCON.2018.03.018.

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[4] B. Koo, B. Shin, and T. F. Krijnen, "Employing Outlier and Novelty Detection for Checking the Integrity of BIM to IFC Entity Associations," vol. 2017, p. 1, 2017, doi: 10.22260/ISARC2017/0002.

[5] T. Bloch and R. Sacks, "Clustering Information Types for Semantic Enrichment of Building Information Models to Support Automated Code Compliance Checking," *Journal of Computing in Civil Engineering*, vol. 34, no. 6, p. 04020040, Jul. 2020, doi: 10.1061/(ASCE)CP.1943-5487.0000922.

[6] T. Su, H. Li, and Y. An, "A BIM and machine learning integration framework for automated property valuation," *Journal of Building Engineering*, vol. 44, p. 102636, Dec. 2021, doi: 10.1016/J.JOBE.2021.102636.

[7] M. Bassier, M. Vergauwen, and B. Van Genechten, "AUTOMATED CLASSIFICATION OF HERITAGE BUILDINGS FOR AS-BUILT BIM USING MACHINE LEARNING TECHNIQUES," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. IV-2-W2, no. 2W2, pp. 25-30, Aug. 2017, doi: 10.5194/ISPRS-ANNALS-IV-2-W2-25-2017.

[8] J. C. P. Cheng, W. Chen, K. Chen, and Q. Wang, "Data-driven predictive maintenance planning framework for MEP components based on BIM and IoT using machine learning algorithms," *Autom Constr*, vol. 112, p. 103087, Apr. 2020, doi: 10.1016/J.AUTCON.2020.103087.

[9] J. Sresakoolchai and S. Kaewunruen, "Integration of Building Information Modeling and Machine Learning for Railway Defect Localization," *IEEE Access*, vol. 9, pp. 166039-166047, 2021, doi: 10.1109/ACCESS.2021.3135451.

[10] F. Pour Rahimian, S. Seyedzadeh, S. Oliver, S. Rodriguez, and N. Dawood, "On-demand monitoring of construction projects through a game-like hybrid application of BIM and machine learning," *Autom Constr*, vol. 110, p. 103012, Feb. 2020, doi: 10.1016/J.AUTCON.2019.103012.

[11] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. " O'Reilly Media, Inc.," 2019.