

BES469389

Automating MEP Element Placement Using Machine Learning in Revit

Patrick Eldridge
KLH Engineers

Learning Objectives

- Discover how machine learning can be used as a tool in the AEC industry.
- Learn how KLH trained a model on a small dataset of objects.
- Learn how to integrate a concrete use of machine learning into a company's workflow.
- Learn that you don't need super computers and a PHD to develop and deploy this technology.

Description

Machine learning is a much talked about and relatively new way for creating artificial intelligence systems and applying them to complicated problems. KLH Engineers is applying artificial intelligence to solve challenging problems in the architecture, engineering, and construction (AEC) industry. This class is a case study on applying machine learning to the problem of MEP element placement based on BIM (Building Information Modeling) data provided by other consultants. Attendees will discover how KLH applied a machine-learning algorithm to classify BIM objects in a room.

Speaker(s)

Patrick Eldridge started in the AEC industry working as a co-op for a plumbing contractor during the great recession. After starting at KLH, Patrick was drawn to creating and programming tools for engineers and other members of the AEC value stream. After five years at the company, Patrick has released several tools that drive the company and our clients forward in the industry. Patrick holds a Bachelor of Science in Mechanical Engineering Technology from the University of Cincinnati. Since 2016 KLH has been on a journey to move the industry forward, including being a 100% Revit company, and having a software department to bring innovative ideas to life.

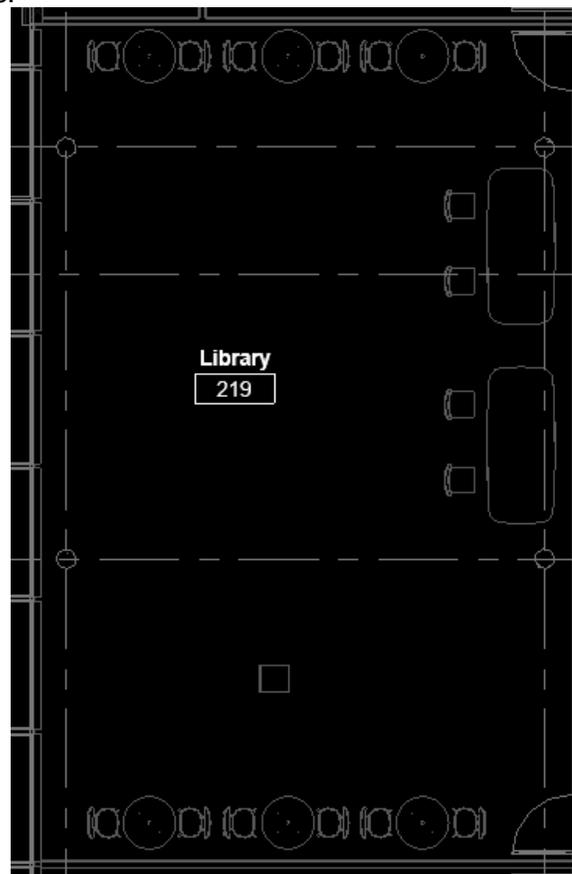
Background for Placement

Design Process Overview from 10,000 ft

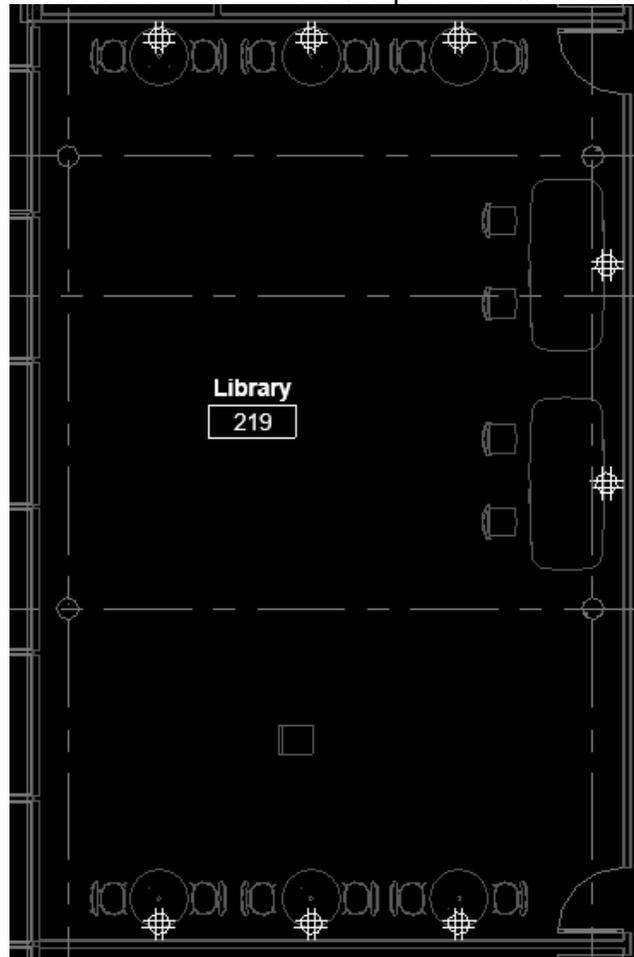
- In general the design process for the AEC industry begins with inputs. Architects, clients, building owners, governing codes, etc all provide inputs that need to be incorporated into the design and satisfied.
- Once those have been gathered, model population can begin in earnest. This can be the most time consuming portion of the project for the design team, and in this break down covers everything from Schematic design all the way through to just before permit/bid/construction sets are issued.
- The next step is final verification that all requirements are met. Signing engineers and architects review the drawing package, calculations are verified, double checks are made that all input requirements are met.
- Finally the drawing package(s) are sent to general contractors and others involved in the construction process.

Traditional Model Population Process

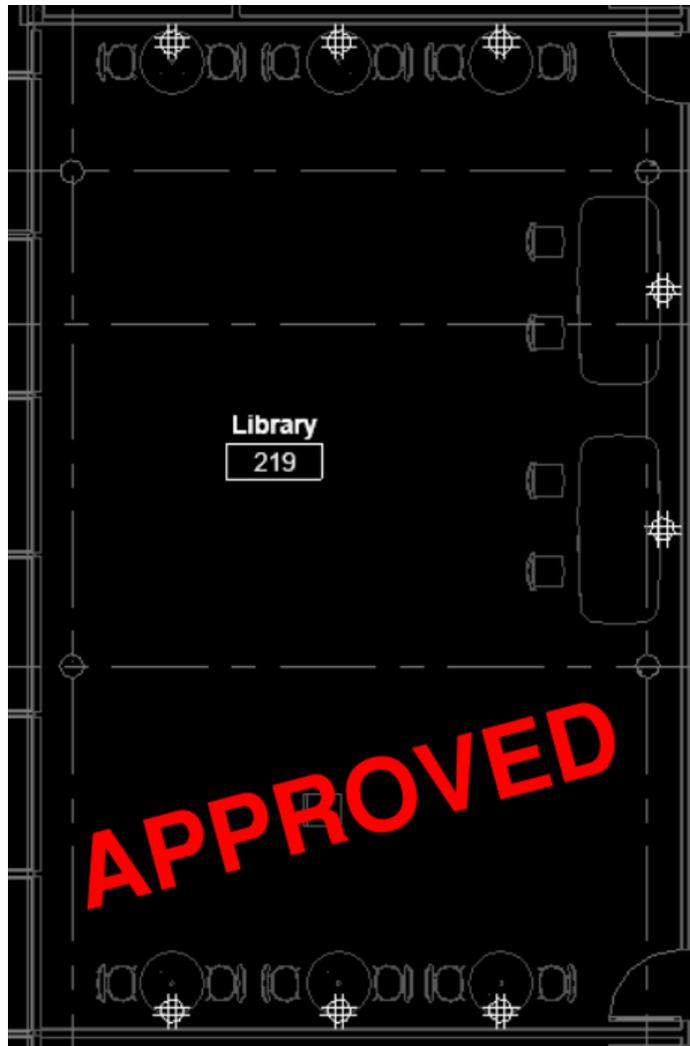
- Scan baseplans looking for elements or areas that require elec/technology/FP/Mech/PL fixtures or devices.



- Place those elements based on known rules/requirements/codes/conventions.



- This design gets reviewed and approved or sent back to designers with items that need to be addressed.



Why Try to Automate Placement?

- Time savings for other areas of project. placement of objects is time consuming, and a large part of the fee for a project goes towards populating the model. For critical pieces of equipment that's not a problem but a lot of devices need to be placed, and the rules governing where they go are fairly simple.
- Better conformance to client / project standards. Devices in a room will be laid out differently for no other reason than 2 different people worked on the project. Which isn't desirable. Manual process for checking client req's means finding every instance where a rule applies and checking, which is slow and error prone.
- Consistency with standards/codes. Manual placement is prone to human errors, missing devices can cause change orders that add previously unknown costs to the project. If

you're going to do something wrong, do it the same wrong way everytime, so you only need to figure out how to fix it once.

Challenges to Automate Placement?

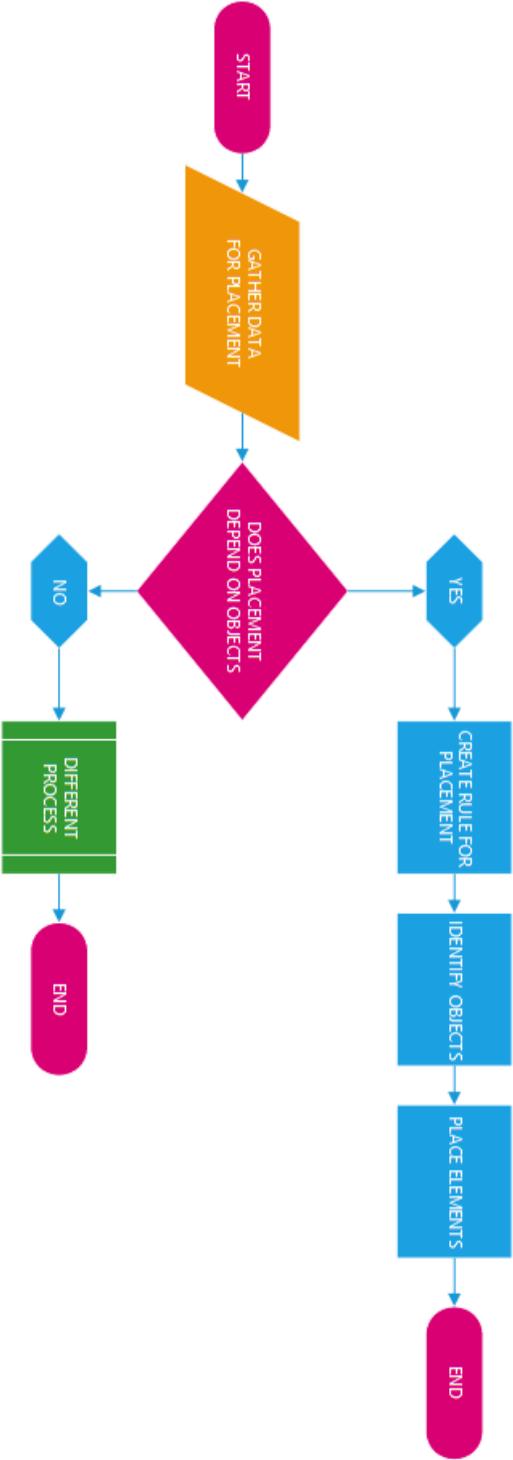
- Architects and clients don't use BIM in the same way. Different clients (mainly architects) use different families, some use generic families more like autocad blocks, others will find the exact product they have scheduled. We can start to database these objects but more likely than not we will just keep coming across new fixtures.
- There is no data standard in the US. If all manufacturer's had to have a specification parameter with the master spec section, that would largely resolve this. Its not likely to happen soon across all 50 states.

Separation of Concerns

When KLH sat down to tackle this, we broke it out into two different problems. Automating device placement across different types of objects, and automatically recognizing what objects are in the building.

Rules for Object Categories

A diagram for rules is on the next page, but quickly let's cover what this process focuses on. This is for MEP elements that have a relationship with an object present on the baseplans. Elements like air diffusers or convenience receptacles aren't placed in relation to architectural objects in the same way a dedicated duplex receptacle is placed for a microwave, or a water connection is placed for a water closet. This process maps an element directly to an architectural object.



Information Gathering

- Access the project needs, making sure to keep a list of architectural items that will need an MEP element everytime. This should be done per spec division, 21,22,23,26,etc.

Creating Rules

- Once the architectural objects have been identified create rules for each MEP device that needs to be placed. KLH uses custom written software to do this, but Dynamo could be utilized as well. The rules should include specific offsets with relation to the architectural object, including quantity.

Identify Objects in the Space.

- Categorize each object in the room or building. This will be needed for the next step.

Place Elements

- With the rules for each category set, and the architectural elements categorized, you're now ready to place the elements.

Integrating Machine Learning into the process

What is machine learning?

For the purposes of this talk, machine learning is a process to **create a model** that can perform predictions on data that you feed it. A diagram for this process is on the next page.

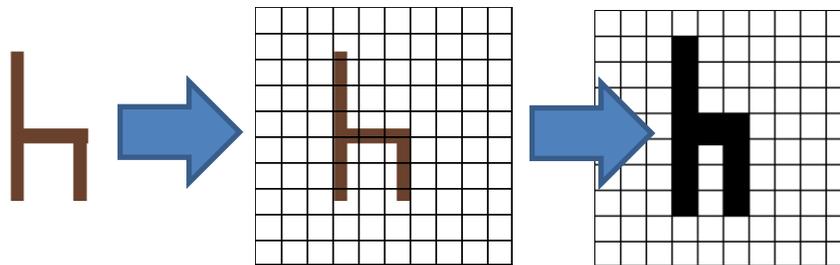


Identify a Task

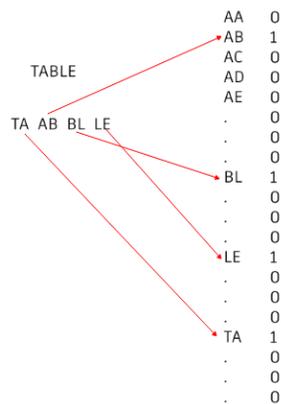
- Starting with the task is essential. Machine learning is a problem looking for a solution, so incorporating it into a process can be tricky.
- After an initial success with machine learning, KLH tried to implement it on several other processes. They were all too big, and we didn't have enough data to get good results.
- Pick a task with clear inputs and outputs.
- We chose to try and automate object detection, because it had clear patterns in the data we could feed to the machine learning model, and clear inputs and outputs.

Prepare Data

- For the data to feed into the model, KLH chose the family geometry, and family name.
- The revit family geometry and name aren't something a model can understand, so they need converted into arrays of numbers.
- KLH used a voxelization technique for the geometry data. This takes a variable input and converts it to a fixed input, which is important for the model. First the family was scaled so it fit inside a grid. Any grid space that had the mesh in it was filled in with a 1. This way an object of any size or detail level can be fed into the model in a uniform way.

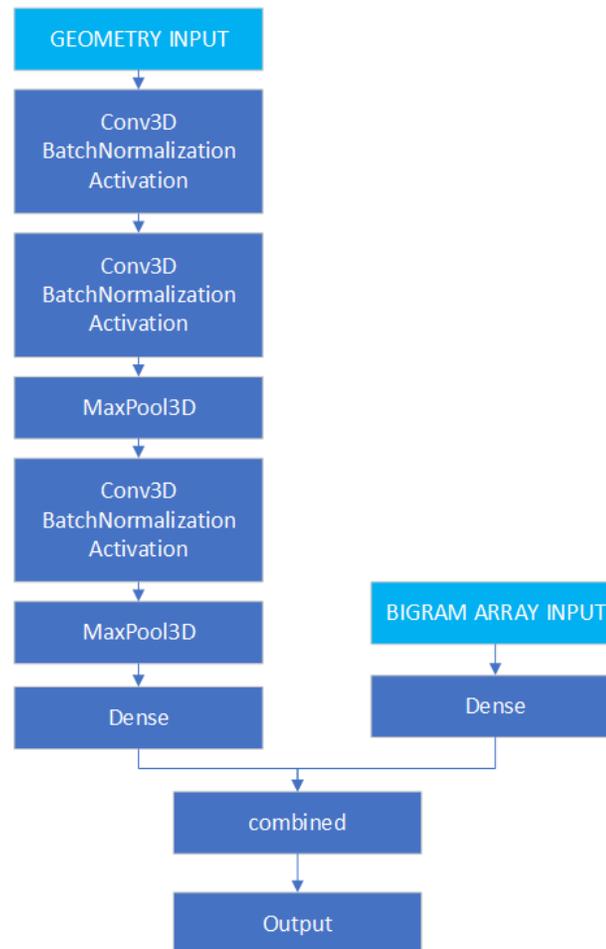


- The model can't just take a name in, so we need to modify it in some way. The way we chose to do that is to break the name up into letter pairings, or bi-grams, and count the total number of bigrams in each word. So we created categories that represent each possible letter pairing, from AA all the way to ZZ, and we did that by using an array all filled with zeros. Then we take the letter pairings or bi grams and for each one we find in the name we add 1 to the category in the array that represents that letter pairing. Again this way we can take a string of any length, count its bigrams, and feed it into the model.



Build and Train the model

- This model was built using Keras with the tensorflow 2.0 back end.
- Below is a basic outline of the model architecture. There's a lot to digest here, but I'm only going to cover the basics. There will be a resources section later on that will point you to all sorts of online resources for learning about AI and ML.
- A model is built up of different layers. Each layer has a job to do in the model. Dense layers and Conv3D layers are there to pick up on patterns in the data that get passed to them. Layers are made up of nodes. Each node keeps track of the strength of its relationship to nodes in the layer in front of it. The relationships of the nodes refine themselves as the model is trained. Data starts in the input layer, and calculations are performed to see which nodes in the next layer are activated. Over time certain nodes activate other nodes in forward layers in patterns. The output layer is the last to be activated. There is a node in the output layer for each possible prediction, and the output node with the highest number is the model's prediction.
- All the layers in this model are "fully connected", which means each node in the layer is connected to all the nodes in the previous layer, and all the nodes in the next layer.
- There is a branch in this model. The voxel data and the bigram data get processed separately from each other, and they are only combined at the end. This was an intentional choice. We wanted the voxel data processing to happen independently of the family name.



Challenges During Implementation

Model Architecture

- The original iterations of this model only used voxel data. Most scholarly papers and studies only use voxel data when performing categorization, so that is the approach we tried.
- In Revit however there is always a family name, and often enough the manufacturer will include something like “table” or “chair” in the name. We believe this helped the model.

Not Enough Data

- The original dataset for this model had 4000+ families across 13 categories. There was a wide variant of sources for the data, and overall the dataset didn’t meet our needs.
- In a revamped dataset, less than 100 families were pulled, and the data was boosted by means of rotating the object around in 3D space. This functioned to give the model several different “views” of the same object, and hopefully generalize the shapes of the object, so it can predict on data its never seen before.

Closing Thoughts

How Much Does Machine Learning Cost?

- Larger models like the one used for Jeopardy can cost millions of dollars just to train on the gigabytes of data that need to be fed to them. I know I associated machine learning to big data, because one preceded the other.
- This doesn't need to be true. Smaller models handling a small task that is otherwise tedious or error prone for whatever reason can have a large impact on your business as they run over time. The model used for this presentation takes less than an hour to train, which equates to less than a dollar in cost.
- I don't want to mislead you though. Coming up with the idea for what to automate, and figuring out how to preprocess the data is about 70%-75% of the work, and can take many hours for a team of people to develop, especially at first.

Resources for Further Learning

Below is a list of resources available on the web. All of these sites and resources are free, as of the time of this handout being made.

- [Kaggle.com](https://www.kaggle.com) – this is a huge repository of datasets, and you also have the ability to publish the models you make for other people to see. You can write the scripts to build and train the models right on the website, and they also have learning courses available for the most prominent machine learning skills.
- [Keras.io](https://keras.io) – If you choose to use the Keras library for building and training your models, this website has a host of examples for you to try out, many of which you can run right on your browser using google colabs, which you get for free with a gmail account(which is also free). This website also has links to the documentation, which is always helpful when you need to look something up.
- [Pytorch.org](https://pytorch.org) – Similar to the Keras website, examples and documentation for the pytorch library. Even though colab is owned by google they don't stop you from using pytorch in colab notebooks, so this documentation also has colabs.
- On youtube the user "Sentdex" has some really great video playlists about building and using neural networks. If you wanted to build one from scratch to really understand whats going on behind the scenes in libraries like Keras and Pytorch, his videos get into those nitty-gritty details.