



EPiC Series in Built Environment

Volume 2, 2021, Pages 228–236

ASC 2021. 57th Annual Associated Schools  
of Construction International Conference



# Investigation of Machine Learning for Clash Resolution Automation

Ashit Harode and Walid Thabet, Ph.D., CM-BIM

Virginia Tech  
Blacksburg, Virginia

Various research work has recently investigated the utilization of Machine Learning for automating the process of clash resolution during design review and coordination of BIM models on construction projects. Literature review shows that current research work focuses on using Supervised Learning for automation of clash resolution. Individual implementation of Supervised Learning has its drawbacks. The automated model trained through Supervised Learning will only be able to resolve clashes similar in nature to those clashes used to train the model. This paper proposes a new approach that integrates Supervised and Reinforcement Learning to overcome these limitations. Reinforcement Learning will assist in overcoming the dependency of Supervised Learning on training data, while Supervised Learning will reduce the time for Reinforcement Learning by eliminating iteration with low rewards or illogical solution. The proposed approach will be able to assist industry practitioners in resolving clashes efficiently and effectively.

**Key Words:** Design Coordination, Clash Resolution, Supervised Learning, Reinforcement Learning, Machine Learning

## Introduction

A major milestone in the design coordination and review process is to generate a clash-free BIM model of the project. Design clashes are the major cause for reduced construction productivity and rework leading to increased project costs (Pärn et al., 2018). The design coordination and review process in itself can be costly with research showing that MEP coordination accounts for 6% of the total MEP cost (Hu et al., 2020). An effective and efficient design review process therefore is required to prevent reduction in productivity and minimize or eliminate budget overruns.

On a construction project, the general process of design review consists of several steps. First, a BIM coordination team will merge multiple trade models into a single file using BIM software such as Autodesk Navisworks. Using built-in clash detection tools, clash tests among different trades are conducted in a prioritized order to identify clashes. Clashes are analyzed, filtered and classified and then discussed during coordination meetings among project team members to find alternative

solutions to resolve the clashes (Hu et al., 2018). This is a slow process and is highly dependent on the BIM coordinator experience and the established process of resolving clashes. Resolving clashes is also an iterative approach, that requires making design changes and running clash tests multiple times until all relevant clashes are eliminated.

One way to improve the current clash resolution process is through automation. Automated methods that can accurately analyze and resolve a large number of clashes need to be developed (Pärn et al., 2018). Application of rule-based automation for clash resolution has been tested. However, developing rule-based automation systems are time-consuming, and their effectiveness depends on the number of rules defined (Huang & Lin, 2019). One alternative solution to rule-based automation is the implementation of Machine Learning.

Machine Learning is a branch of computer science that relates to how machines can learn to self-perform tasks similar to humans without explicit coding (Huang & Lin, 2019). Based on how knowledge is transferred to the machine, machine learning can be divided into supervised learning, unsupervised learning, and reinforcement learning (Theobald, 2017). With multiple machine learning processes available to impart knowledge to the machine, complex, repetitive, and time-consuming construction tasks can be effectively and efficiently automated. Recent research on the automation of clash classification and resolution using machine learning has also been conducted. Hsu, et.al, 2020 used supervised learning to automate the clash resolution of MEP for the basement of student residence (Hsu et al., 2020). Huang and Lin, 2019 worked on comparing the accuracy of rule-based automation and supervised learning in automating the clash classification task (Huang & Lin, 2019).

Though current research shows promising results towards the automation of clash resolution, there are limitations to the methods currently being adopted for the automation of clash resolution. These limitations can be attributed to the selection of machine learning process adopted for automation. As most of the current research on the use of machine learning for automation of clash resolution process focuses only on the use of supervised learning, this paper has two main objectives:

1. Identifying the shortcomings of current clash resolution automation approaches that use supervised machine learning only.
2. Propose a framework for a new approach to automate clash resolution using an integrated Supervised and Reinforcement Learning method for better automation.

The following sections will discuss the clash coordination process, different machine learning methods, the existing work done in the field of automation in construction, and drawbacks of the current methodology adopted for the automation of clash resolution process. The paper concludes with providing a methodology for developing a new machine learning model for automation of clash resolution.

## **Clash Coordination**

Design clashes can often be attributed to factors like incorrect level of detail, design complexity, design errors or lack of accuracy due to deadline constraints (Akponeware & Adamu, 2017). Software platforms such as Autodesk Navisworks enables BIM coordinators to automatically identify clashes in a federated model. Once clashes are identified, they are analyzed and classified as either relevant or irrelevant clashes based on their impact on the project (Tommelein & Gholami, 2012). Classification of clashes is an important step in clash management because individually reviewing all clashes during coordination meetings is time-consuming and will waste meeting time and reduce trust of project

participants in BIM (Hu et al., 2018). Relevant and irrelevant clashes can be further divided into true-positive clashes, false-positive clashes, and false-negative clashes (Leite, 2020). This categorization is based on whether the clash is relevant or not and whether it was identified in the clash identification process.

1. True-Positive clashes are the clashes that are relevant and were identified as clashes in the clash identification process.
2. False-Positive clashes are the clashes that are considered irrelevant as they do not cause loss of productivity or are duplicate clashes and are identified as clashes in the clash identification process.
3. False-Negative clashes are the clashes that are relevant and were not identified as clashes in the clash identification process due to human error in manual clash identification or modeling error/limitation leading to missing elements that would have resulted in a clash in automatic clash identification.

Despite efforts to automate clash identification through software, classifying clashes as relevant or irrelevant and standardizing the priority for clash resolution, the manual process of clash resolution remains slow (Hsu et al., 2020). One factor could be that BIM coordinators need to be meticulous in the clash resolution process to make sure that the process does not generate new clashes or violates any design restrictions. This attention to detail requires time to resolve clashes. Another factor contributing to the slow clash resolution process can be that multiple iterations of the design model need to be generated with every subsequent coordination meeting to capture the design changes made to resolve clashes. Time delays in the clash resolution process often increase with the increase in the number of clashes (Lee & Kim, 2014). As slow decision-making process is a major contributor to project delays (Arantes et al., 2015), there is an immediate need to improve the manual clash resolution process. One way to overcome these challenges is through automation of clash resolution using Machine Learning.

## **Machine Learning**

Machine Learning is a branch of computer science that focuses on imparting knowledge to the machines to self-perform tasks (Huang & Lin, 2019). The process of imparting knowledge or knowledge modeling in Machine Learning can be divided into two main steps. Step one includes training the machine learning model, and step two involves testing the trained machine learning model. Data is collected to train the machine learning model in a specific domain. Once the model is trained the accuracy of model to automate the task is analyzed using the training data. The model is then adjusted until the desired accuracy is obtained.

Based on the method used to impart knowledge to the machine, Machine Learning can be divided into three categories: (1) Supervised Learning; (2) Unsupervised Learning; and (3) Reinforcement Learning.

### *Supervised and Unsupervised Learning*

During the training phase of Supervised Learning, the machine learning model is provided with both descriptions for each data instance in a training dataset, called features and the corresponding values for the data instance that needs to be predicted, called labels. The features and labels act as the prior knowledge provided to the machine learning model. During the training step the machine learning model develops a relationship between the features and labels. During the testing step the machine

learning model is provided only with the features for each of the testing data instances. Using the relationship developed in the training step and the given features from the testing data, the machine tries to predict the labels for the data instances. The accuracy of the model is defined as the percentage of labels that were predicted correctly by the model.

In Unsupervised learning the machine learning model is only provided with features and no labels. As Unsupervised Learning focuses on analyzing relationship between features (Theobald, 2017) it can be useful in cases where labels for the data are unknown and the objective of the machine learning model is to analyze the data distribution.

### *Reinforcement Learning*

Unlike supervised learning, a reinforcement learning machine learning model is not provided with explicit prior knowledge. Rather the machine learning model is allowed to explore its environment freely and gather knowledge based on its success and failure in making predictions. The machine learning model can then utilize the knowledge collected to improve its accuracy and improve its interaction with its environment. For each successful interaction with the environment, the machine is given a numerical reward to encourage similar interaction in the future. For each unsuccessful interaction with the environment, a numerical penalty is assigned to discourage similar interactions in the future.

A reinforcement learning model has five components: (1) Agent: the learner and decision maker, (2) Environment: everything the agent interacts with and is external to the agent, (3) Actions: interactions the agent makes with the environment, (4) State of the Environment: each representation of the environment, and (5) Reward: numerical value given to the agent based on how beneficial the actions taken by the agent is for task completion (Sutton & Barto, 2018).

During the reinforcement learning process the agent analyzes the current state ( $S_t$ ) of the environment at any given time  $t$  and takes an action ( $a_t$ ). Based on the action taken by the agent, the state of the environment changes from  $S_t$  to  $S_{t+1}$ . The agent also receives a numerical reward ( $R_t$ ) based on the success of the action. This iterative loop continues until the agent learns to respond to the states with actions that maximize the total reward received by the agent at the end of each training. In reinforcement learning the action ( $a$ ) taken by the agent for a state ( $s$ ) of the environment is determined by a policy function ( $\pi$ ). Formally, a policy function maps states of an environment to the probabilities of selecting each passible action. Therefore, a policy function  $\pi(a|S)$  is equal to the probability of selecting action ( $a$ ) given the state  $S$  of the environment (Sutton & Barto, 2018).

During reinforcement learning the focus is on optimizing this policy function to improve the agent response to the states of the environment. With each iteration of the reinforcement learning loop, the policy function improves. Let's denote the improved policy function as  $\pi'$ . The loop is iterated until the total reward obtained by the agent at the end of each loop becomes constant and no increase is seen in the total rewards for any subsequent iterations. The constant reward observed signifies that the agent has achieved the most optimal policy function ( $\pi^*$ ).

## **Review of Literature on Automation of Clash Resolution**

Literature review using Elsevier's Scopus database was conducted to identify current research work focused on automation of Clash Resolution using Machine Learning. To ensure that the research is current, only refereed manuscripts published since 2015 were considered for the literature review. Hu,

Y., Castro-Lacouture, D., Eastman, C. M., & Navathe, S. B. (2020) used Supervised Learning, and simulated annealing as the optimization process, to create an artificial intelligence system for automatic design resolution of MEP for the basement of student residence. Huang, Y.-H., & Lin, W. Y. (2019, May 24) conducted research to automate classification of design conflicts using Supervised Learning. The research found that the classifier trained using Supervised Learning is more accurate than using a rule-based system in predicting the type of clash. Liu, J., Liu, P., Feng, L., Wu, W., Li, D., & Chen, F. (2020) utilized Q-Learning, a model free reinforcement learning, to reduce the average rebar design time in reinforced concrete members by 90% by automating the rebar design process.

It can be seen from the current research that the implementation of machine learning to automate construction tasks has proven to generate better results compared to performing the tasks manually. It should also be noted that as more data for training the machine learning model becomes available through the use of laser scanners, machine vision, sensors, and building information modelling (Pärn et al., 2018), machine learning models will become more accurate and efficient.

Though examples of machine learning methods discussed in this section to automate construction tasks have shown great accuracy, more research is needed to continue the optimization of machine learning in construction. This optimization can be focused on overcoming the drawbacks of currently used machine learning models. For example, there are two drawbacks to the machine learning models currently adopted for the automatic clash resolution:

1. Kangin, D., & Pugeault, N. (2018) realized that while using supervised learning, knowledge of the model is restricted to the quantity and quality of data. Therefore, a large quantity of data is required to train an automated clash resolution model using supervised learning.
2. Huang, Y.-H., & Lin, W. Y. (2019, May 24) concluded that a model trained using supervised learning can only be used to resolve clashes similar to the clash data provided to train the machine learning model. The clash data set needs to be extended to be able to resolve a wide variety of clashes.

### **Proposed New Methodology for Using Integrated Supervised and Reinforcement Learning for Clash Resolution**

Inspired by Kangin, D., & Pugeault, N. (2018) combined Supervised–Reinforcement Learning approach, this paper proposes a methodology to develop a new model for clash resolution that aims at using supervised learning to pre-train the reinforcement learning model. This pre-training will help in avoiding exploration of actions that will result in penalties or smaller rewards if Reinforcement Learning is used independently. The pre-training will also prevent reinforcement learning from exploring options that will not result in meaningful results. This will allow reducing the overall steps in reinforcement learning. The use of Reinforcement Learning will also help in expanding the knowledge by allowing the model to continue learning from its correct and wrong actions.

Figure 1 shows a diagrammatic representation of the methodology adopted to develop the proposed model. The methodology is divided into two parts. Each part includes several steps for developing an intergraded supervised reinforcement learning model.

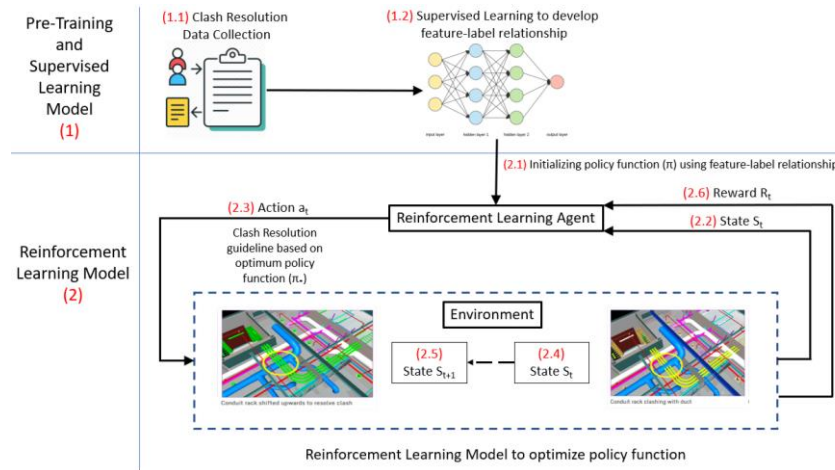


Figure 1. Proposed Integrated Supervised - Reinforcement Learning Model

### Part 1: Pre-Training and Supervised Learning Model

- 1.1 Clash Resolution and Data Collection: Data is collected after interviewing industry professionals to capture their input on possible clash resolution alternative solutions based on examples of clashes provided to them using different case studies.
- 1.2 Supervised Learning to develop feature-label relationship: The collected data is then used to train the supervised learning model. The training of supervised learning model will result in the formulation of a relationship between the features (clash information) and labels (clash resolution guideline).

### Part 2: Reinforcement Learning Model

- 2.1 Initializing policy function ( $\pi$ ) using feature-label relationship: The relationship between features and labels established in part 1 will act as the initial policy function ( $\pi$ ) for the reinforcement learning agent.
- 2.2 State  $S_t$ : In the first iteration of reinforcement learning, reinforcement learning agent will receive state  $S_t$  from the environment which is the 3D BIM model of a building.
- 2.3 Action  $a_t$ : Using the initial policy function generated in step 2.1 and the state  $S_t$  the agent will take the action ( $a_t$ )
- 2.4 Changing from state  $S_t$ : The action  $a_t$  will act on the current state  $S_t$  of the environment.
- 2.5 Changing to state  $S_{t+1}$ : The action  $a_t$  will change the state  $S_t$  of the environment to the state  $S_{t+1}$ .
- 2.6 Reward  $R_t$ : Based on how desirable the new state  $S_{t+1}$  is to the clash resolution; the agent will receive the reward  $R_t$ .

Based on the reward  $R_t$ , the agent will update the policy function ( $\pi$ ). Using new policy function ( $\pi'$ ) and the new state  $S_{t+1}$ , the reinforcement learning agent will again take a new action. This reinforcement learning iterative loop will continue to run until the reinforcement learning agent will learn an optimal policy function ( $\pi^*$ ). This optimal policy function ( $\pi^*$ ) will be used to resolve clashes by the agent.

In the following section the two parts of the methodology will be discussed in detail.

### *Pre-Training and Supervised Learning Model*

The supervised learning model of the proposed framework will be used to develop an initial relationship between clash information (features) and proposed solution guidelines for clash resolution (labels).

To train the supervised learning model, data for clash resolution will be gathered by interviewing industry professionals. The interviews will include examples of clashes along with details for the clashes, such as clashing system type, the distance of the clash from the edge of the elements involved in the clash, and type of clash. The respondents will then be asked to suggest possible actions to resolve the clashes in terms of what system or element (revised system or element) needs to be revised to resolve the clashes and what changes to the system or element will resolve the clash.

The data gathered through the interviews will act as the existing knowledge and will be used to train the supervised learning model. The supervised learning model will then predict the relationship between the features and labels. This relationship will form the bases of pre-training the reinforcement learning model.

### *Reinforcement Learning Model*

The goal of reinforcement learning is to find the optimum policy function that can maximize the total reward an agent can receive while interacting with its environment. To achieve this, the reinforcement learning model starts with a random initial policy function and keeps improving this policy until it achieves the optimum policy. As this will be an iterative process of agent interacting with its environment, the number of iterations will increase with the increase in the complexity of the problem.

For example, a reinforcement learning model to resolve a clash between a beam and a duct in a building will have the 3D model of the building as its environment, the clash and clash information as the state of the environment and will start with a random initial policy function. This random initial policy function ( $\pi$ ) can be, “there is an equal probability that moving the duct up, down, left, or right will resolve the clash”. Based on this policy function the agent thinks that moving the duct in any direction will resolve the clash. Therefore, the agent will randomly pick a direction and move the duct in that direction for the first iteration. Let’s say the duct was moved up. If the clash is resolved the agent receives a reward and the probability of selecting the direction up to resolve similar clashes is increased in the new policy ( $\pi'$ ). If the clash still exists after moving the duct up or if a new clash is created the agent will receive a penalty and the probability of selecting the up direction to resolve similar clashes is therefore decreased in the new policy ( $\pi'$ ). After multiple iterations are made by the agent to resolve the different clashes in the model, the probabilities to move the ducts in different directions based on the clash information will keep improving, hence improving the total rewards the agent can receive at the end of each iteration. The policy function that generates the maximum total reward is the optimal policy function ( $\pi^*$ ) to resolve the clash. This policy would be used by the model agent to resolve subsequent clashes.

For the proposed integrated Supervised and Reinforcement Learning model, the states of the environment in reinforcement learning can be thought of as similar to the clash information used as features in supervised learning and actions in reinforcement learning can be thought of as the guidelines for resolving the clashes. Hence, the reinforcement learning problem for clash resolution automation can be defined as identification of an optimal policy function ( $\pi^*$ ) that maps the given clash information (state) to the probabilities of clash resolution guidelines such that the agent receives the maximum rewards after all the clashes are resolved.

In the proposed model, instead of selecting a random initial policy function to map states to actions, relationship developed between the clash information (features) and clash resolution guideline (labels) in supervised learning training can be used. This relationship will provide a much robust initial policy function for reinforcement learning. As this initial policy will be developed using data collected from the industry this will prevent the agent from selecting low rewards and illogical actions from the first iteration of reinforcement learning. This in turn will also reduce the number of iterations required to achieve the optimal policy function.

The reinforcement learning model will continue optimizing the initial policy function (relationship developed using supervised learning) with each iteration. Therefore, the final policy will be able to perform better than the relationship developed by a supervised learning model alone.

The reinforcement model will be trained using an algorithm that will learn through interacting with the actual environment. In reinforcement learning these algorithms are called model free algorithms as they gain knowledge by interacting with the actual environment rather than a simulation/model of the environment.

For the development and implementation of the reinforcement model, the reinforcement learning algorithm will be allowed to interact iteratively with a Revit model. Using the Revit model, the algorithm will make attempts in an iterative fashion to resolve clashes present in the model. With each iterative interaction, the reinforcement model will develop knowledge on the most effective ways to resolve the clashes. This process will become the part of training the reinforcement learning to resolve clashes. Once the training for the reinforcement learning model is completed, the reinforcement learning model can be used to automate the clash resolution for other Revit models using the knowledge gained during training.

## Conclusion

Existing literature for automating clash resolution using machine learning focuses on using Supervised Learning only. The effectiveness of Supervised Learning depends on the quality of data defined during the training phase. This paper proposes a new approach that integrates Supervised and Reinforcement Learning to develop an automated clash resolution model that can learn faster than traditional stand-alone Reinforcement Learning and create its policies for clash resolution, an advantage over conventional Supervised Learning. In the proposed integrated model, Supervised Learning will be used to pre-train the reinforcement learning model. Because of the pre-training, the reinforcement learning model will start with a random policy based on the relationship between features and labels provided by the supervised model and established through industry practitioners' knowledge. The research anticipates that the number of iterations required to reach an optimal policy function will also be reduced due to pre training.

The proposed model can be further improved with the addition of a rule-based model that can check if the clash resolution proposed by the model satisfies design standards and code requirements. This way the proposed solution for resolving the clash is verified against design and code regulations before implementation. As the use of rule-based model to verify the correctness of clash resolution as per design standards and code is out of the scope of this paper, specification on how the rule-based model would be developed and implemented is not discussed. The authors understand the importance and value of such a rule-based model and plan to discuss it in future publications.



## Reference

- Akponeware, A. O., & Adamu, Z. A. (2017). Clash detection or clash avoidance? An investigation into coordination problems in 3D BIM. *Buildings*, 7(3).  
<https://doi.org/10.3390/buildings7030075>
- Arantes, A., Fernandez da Silva, P., & Miguel F Ferreira, L. D. (2015). *Delays in construction projects-Causes and impacts*.
- Hsu, H. C., Chang, S., Chen, C. C., & Wu, I. C. (2020). Knowledge-based system for resolving design clashes in building information models. *Automation in Construction*, 110.  
<https://doi.org/10.1016/j.autcon.2019.103001>
- Hu, Y., Castro-Lacouture, D., & Asce, A. M. (2018). *Clash Relevance Prediction Based on Machine Learning*. [https://doi.org/10.1061/\(ASCE\)](https://doi.org/10.1061/(ASCE))
- Hu, Y., Castro-Lacouture, D., Eastman, C. M., & Navathe, S. B. (2020). Automatic clash correction sequence optimization using a clash dependency network. *Automation in Construction*, 115.  
<https://doi.org/10.1016/j.autcon.2020.103205>
- Huang, Y.-H., & Lin, W. Y. (2019, May 24). *Automatic Classification of Design Conflicts Using Rule-based Reasoning and Machine Learning—An Example of Structural Clashes Against the MEP Model*. <https://doi.org/10.22260/ISARC2019/0044>
- Kangin, D., & Pugeault, N. (2018). Continuous Control With a Combination of Supervised and Reinforcement Learning. *2018 International Joint Conference on Neural Networks (IJCNN)*, 1–8. <https://doi.org/10.1109/IJCNN.2018.8489702>
- Lee, G., & Kim, J. W. (2014). Parallel vs. Sequential cascading MEP coordination strategies: A pharmaceutical building case study. *Automation in Construction*, 43, 170–179.  
<https://doi.org/10.1016/j.autcon.2014.03.004>
- Leite, F. L. (2020). *BIM FOR DESIGN COORDINATION A Virtual Design and Construction Guide for Designers, General Contractors, and MEP Subcontractors*. Retrieved from [www.wiley.com/go/permissions](http://www.wiley.com/go/permissions).
- Liu, J., Liu, P., Feng, L., Wu, W., Li, D., & Chen, F. (2020). Towards automated clash resolution of reinforcing steel design in reinforced concrete frames via Q-learning and building information modeling. *Automation in Construction*, 112. <https://doi.org/10.1016/j.autcon.2019.103062>
- Pärn, E. A., Edwards, D. J., & Sing, M. C. P. (2018). Origins and probabilities of MEP and structural design clashes within a federated BIM model. *Automation in Construction*, 85, 209–219.  
<https://doi.org/10.1016/j.autcon.2017.09.010>
- Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: an introduction. In *Adaptive computation and machine learning*. Cambridge, Massachusetts: The MIT Press.
- Theobald, O. (2017). *Machine learning for absolute beginners: A plain English introduction*. Scatterplot Press.
- Tommelein, I. D., & Gholami, S. (2012). Root Causes of Clashes in Building Information Models. In I. D. Tommelein & C. L. Pasquire (Eds.), *20th Annual Conference of the International Group for Lean Construction*. San Diego, California, USA. Retrieved from <http://iglc.net/Papers/Details/850/pdf>