Automatic Classification of Design Conflicts Using Rule-based Reasoning and Machine Learning—An Example of Structural Clashes Against the MEP Model

Y. Huang[a] and W.Y. Lin[b]

[a]Department of Civil and Construction Engineering, National Yunlin University of Science and Technology, Taiwan
[b]Department of Civil Engineering, Feng Chia University, Taiwan
E-mail: huangyh@yuntech.edu.tw, weiylin@mail.fcu.edu.tw

Abstract -

With the emergence of 3D technologies in a recent decade, BIM software makes it easy to detect those conflicts in the early stage of a project. Clash detection in BIM software is now a common task. Among those conflicts found by BIM software, however, a relatively high percentage belongs to 'pseudo conflicts'—which are permissible or tolerable, but BIM software does not reveal this information. Thus, currently BIM managers have to manually inspect every detected conflict to classify the type of conflict. Some researchers urged an automated process to facilitate this laborious process. This study implemented both a rule-based reasoning system and machine learning classifiers to help classify those BIM-detected conflicts. Preliminary testing results indicate that machine learning algorithms can achieve comparable results to a traditional rule-based system, but with much less costs and energy in developing.

Keywords –

Clash detection, Machine learning, Rule-based reasoning, BIM

1 Introduction

Generally, engineering designs are constructed and compiled by engineers of different professions, such as those in construction, structural engineering, and mechanical, electrical, and plumbing (MEP). Therefore, conflicts between different design disciplines are inevitable [1, 2]. Design conflicts are errors in which building components overlap in a certain space when various design plans are being compiled. Minor design conflicts often induce rework and cost increases, whereas severe conflicts may cause change orders, resulting in cost overruns, delays, and labour safety issues. Therefore, project performance will be substantially affected if design conflicts are not properly mitigated. [3].

The emergence of building information modelling (BIM) software in recent years has simplified design clash detection, with conflict or interference checking being basic functions of BIM software programs. Because resolving design conflicts is critical to project performance, combined with the popularity of BIM software, building codes in many countries have mandated the enforcement of clash detection in public projects [1]. For example, in the United Kingdom, a design team is mandated to perform clash detection every 1 to 2 weeks to ensure that engineering designs are fully coordinated and free of conflicts, thus reducing the likelihood of design changes occurring.

However, a discussion of effective methods to mitigate design conflicts and the use of BIM to assist in decision-making has been neglected in the literature [3]. Due to the existence of numerous engineering interfaces in a single project, BIM software can often easily detect hundreds of clashes, even for small-scale projects. BIM managers thus must examine and analyse each conflict manually. Some researchers have stated that although BIM software programs have excellent clash detection functions, the classification of detected clashes is still difficult, time-consuming, and costly, and thus an automated classification method is urgently required [1].

Because classifying clash types requires the input of experts with vast knowledge and experience, conventionally a rule-based expert system is often used for automated process development. However, rule acquisition when developing an expert system is time-consuming and laborious, and its prediction performance is usually limited by the number of rules. By contrast, supervised learning (i.e., machine learning) only requires the collection of sufficient cases and appropriate feature manipulation to quickly obtain required results. Furthermore, its accuracy increases continuously as the number of cases increases.

This study used the structural and MEP models of a construction project as an example, and implemented

three automated processes to classify those conflicts detected by BIM software using rule-based expert system, individual classifiers of supervised learning, and multiple classifiers of ensemble learning, respectively. Their performances in terms of predictive accuracy were then compared and evaluated. The results showed that the accuracy of the rule-based expert system was approximately 58% while 92% and 94% were acquired using individual classifiers and ensemble learning, respectively. This indicated that machine learning algorithms can achieve comparable results to a traditional rule-based system regarding conflict classification, but with much less costs and energy in developing.

2 Related Work

2.1 Spatial clashes

Wu [4] divided spatial conflicts into four categories: design conflicts, construction conflicts, damage conflicts, and congestion. Design conflict refers to spatial overlap between building components when different design teams develop their own designs. Wu also found that the existence of numerous interfaces among construction project participants is the main source of design conflicts. Researchers in another participatory action research project in the United Kingdom introduced a collaborative environment in a jointly designed large-scale, multistory construction project, and assisted the designers in avoiding design conflicts. However, more than 400 conflicts were still observed between the structural model and the MEP model [1]. The study revealed that although collaboration can decrease design conflicts, clash detection still remains a necessary process.

In summary, design collaboration does not prevent all conflicts; thus, clash detection and resolution are still necessary. Although most BIM software provide clash detection, classifying and resolving those detected clashes remain manual and difficult. The UK study [1] found that after performing clash detection, BIM managers need to classify types of those detected clashes, namely: (1) errors, which must be resolved; (2) pseudo clashes, which are permissible and does not require to be resolved; 3) deliberate clashes; and (4) duplicate clashes. Due to the considerable number of engineering interfaces, BIM software will produce substantial clash detection results, even for small-scale projects. BIM managers must go through the above-mentioned manual process. Since this process is highly mechanized, time-consuming, and costly, an automated classification method is urgently required; otherwise, the benefits of clash detection provided by BIM software will be reduced due to information overload [1]. Concluding remarks in those studies illustrated the necessity and importance of this study.

2.2 Machine learning in civil engineering

Machine learning is a computer science related to how machines can learn in a similar manner to humans. In machine learning, the process of human learning and adjustment is transferred to machines. According to whether or not the answers are provided during the training, machine learning has been usually identified as two categories: (1) Supervised learning：During training, answers (labels) are supplied to a question to provide the machine with the ability to recognize errors. This study applied this approach to train and test those classifiers; and (2) Unsupervised learning：No answers are provided to a question during training process; thus, the machine must find the answer itself. The expected results from this learning method are usually less accurate than those in supervised learning.

Machine learning has widely been used in numerous studies on civil engineering applications, such as for monitoring construction progress using 4D BIM, performing regulatory inspections using BIM, reconstructing 3D models using computer vision, and using static images to monitor constructability [5].

The development of machine learning algorithms involves five basic steps [6]: (1) Obtaining data required for training, analysing and selecting the features related to problem solving. (2) Choosing a performance metric. (3) Choosing a classifier and optimization algorithm (4) Evaluating the performance of the model; and (5) Tuning the algorithm. Once learning is completed, estimations can be performed by inputting new data into the algorithm. In most problem-solving tasks, the classifiers can be either linear or nonlinear models. The most common linear model is the linear regression model while non-linear models include the decision trees, k nearest neighbors (k-NN), support vector machine (SVM), and deep learning. However, due to problems having different domain-specific characteristics, it is nearly impossible for a single classifier to work well across all scenarios [7]. Besides, the performance may be dependent on the training and testing data, especially when the number of samples is not large enough. Therefore, it is highly suggested to compare the performance of different classifiers and select the best model for the domain problem.

Since the problem of identifying clash types in this study belongs a classification problem in machine learning, this study used non-linear models to implement individual classifiers and ensemble learning models.

3 Methodology

To evaluate the performance of the rule-based reasoning and machine learning algorithms on classification of clash types, this study developed a process of research methodology as shown in Figure 1. Section 3.1 and 3.2 describes the collection of clash cases and how we labelled the clash type by human experts. Section 3.3 introduced the implementation of the rule-based system, followed by Section 3.4 and 3.5 which detailed the training of single classifiers and multiple classifiers, respectively. Section 4 then provides the results of prediction by different methods using the testing dataset as well as the evaluation of their performance in terms of prediction accuracy。

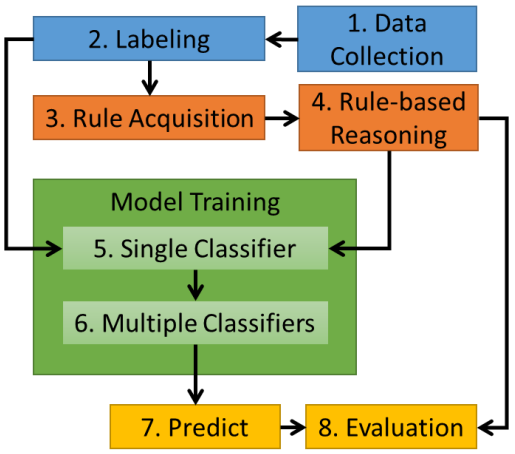
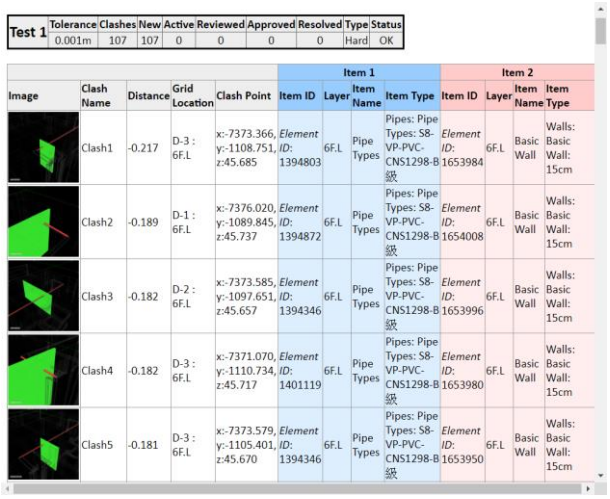


Figure 1. Process of research methodology

3.1 Data Collection

This study used a large shopping mall with nine floors above ground and four floors underground as a sample case. Three floors of the structure were extracted, and Autodesk Navisworks was employed for clash detection. To reduce the load of subsequent labelling process, this study only selected water supply pipelines within the MEP model against the structural model for clash detection. Figure 2 shows the clash detection report for models on the sixth floor . The table at the top shows the total number of clashes while the following table lists all details of each clash, including grid location, clash point, distance, and information about the two clashed items (e.g., item ID, layer, item name, and item type). In addition, two image icons are available. The image can also be viewed by clicking on the icon (Figure 3). Table 1 shows the statistics of the entire clash detection report including 415 structural clashes against flow segments such as pipes and fittings.



Test 1	Tolerance	Clashes	New	Active	Reviewed	Approved	Resolved	Type	Status
	0.001m	107	107	0	0	0	0	Hard	OK

Image	Clash Name	Distance	Grid Location	Clash Point	Item 1 ID	Item 1 Layer	Item 1 Name	Item 1 Type	Item 2 ID	Item 2 Layer	Item 2 Name	Item 2 Type
	Clash1	-0.217	D-3 : 6F.L	x:-7373.366, y:-1108.751, z:45.685	Element ID: 1394803	6F.L	Pipe Types	Pipes: Pipe Types: S8-VP-PVC-CNS1298-B 级	Element ID: 1653984	6F.L	Basic Wall	Walls: Basic Wall: 15cm
	Clash2	-0.189	D-1 : 6F.L	x:-7376.020, y:-1089.845, z:45.737	Element ID: 1394872	6F.L	Pipe Types	Pipes: Pipe Types: S8-VP-PVC-CNS1298-B 级	Element ID: 1654008	6F.L	Basic Wall	Walls: Basic Wall: 15cm
	Clash3	-0.182	D-2 : 6F.L	x:-7373.585, y:-1097.651, z:45.657	Element ID: 1394346	6F.L	Pipe Types	Pipes: Pipe Types: S8-VP-PVC-CNS1298-B 级	Element ID: 1653996	6F.L	Basic Wall	Walls: Basic Wall: 15cm
	Clash4	-0.182	D-3 : 6F.L	x:-7371.070, y:-1110.734, z:45.717	Element ID: 1401119	6F.L	Pipe Types	Pipes: Pipe Types: S8-VP-PVC-CNS1298-B 级	Element ID: 1653980	6F.L	Basic Wall	Walls: Basic Wall: 15cm
	Clash5	-0.181	D-3 : 6F.L	x:-7373.579, y:-1105.401, z:45.670	Element ID: 1394346	6F.L	Pipe Types	Pipes: Pipe Types: S8-VP-PVC-CNS1298-B 级	Element ID: 1653950	6F.L	Basic Wall	Walls: Basic Wall: 15cm

Figure 2. Illustration of the clash detection report produced by Autodesk Navisworks 2017

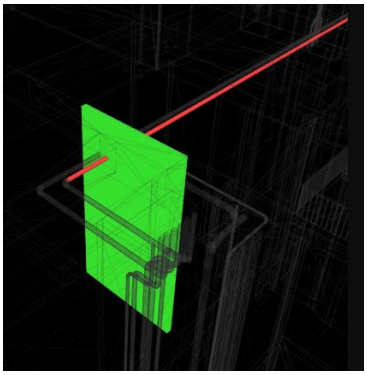


Figure 3. The snapshot of a clash produced by Autodesk Navisworks 2017

Table 1. Statistics of the clash detection report

	Pipes	Fittings	Total
Framings	153	25	178
Walls	155	68	223
Columns	1	1	2
Landings	12	0	12
Total	321	94	415

3.2 Labeling the Clash Types

After completing the clash detection report, the research team integrated the HTML-based clash report into a spreadsheet. Two senior engineers with experience in clash resolution were requested to classify the clash types. A column with a drop-down menu in the spreadsheet was provided for the experts to carry out labelling (Figure 4). This spreadsheet contains all the information in the clash detect report of Figure 2, including a hyperlink to view the clash image, as shown in Figure 3. The drop-down menu consists of four

options with intuitive common vocabulary: severe clashes, negligible clashes, legal interventions, and unknown. These four options were mapped to the four categories suggested in the literature: errors, pseudo clashes, deliberate clashes, and unknown, respectively. The research team also provided the experts with 2D CADs and 3D models of the building for their references.

The labeling process took approximately 10 days. Table 2 shows the summary of results by the two experts. After a detailed comparison was undertaken, 89 clashes were labelled with different answers by the two experts, accounting for approximately 20% of the total number of clashes. Therefore, the remaining 326 cases were used for subsequent analysis.

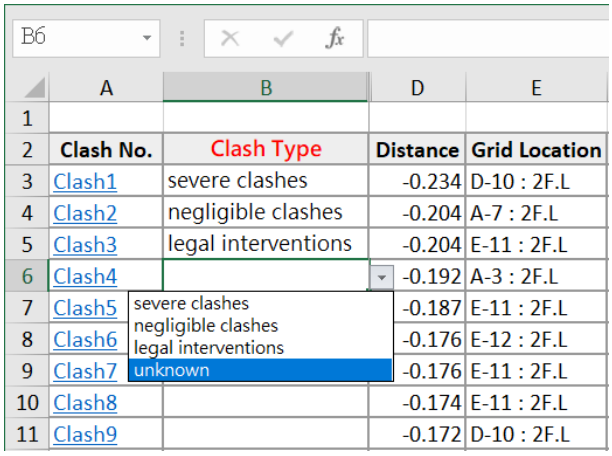


Figure 4. Spreadsheet-based clash detection report for labelling by human experts

Table 2. Statistics of training and testing data

Clash Type	Expert #	Expert #2
Serious clashes	159	208
Neglectable clashes	0	0
Legal intervenes	174	148
Unknown	82	59
Total	415	415

3.3 Rule-based reasoning

After obtaining the labelling results from the experts, the research team acquired 4 rules of clash type classification rules by interviewing the two experts as follows:

1. Beam rule: If the structural component of a clash is a beam, then the clash type is either a severe clash (i.e., error) or negligible clash (i.e., pseudo clash). Severe clashes located near two ends of the beam can affect structural safety, whereas negligible clashes do not affect structural safety but require perforation processing.
2. Column rule: If the structural component of a clash is a column, then the clash type must be a severe clash (i.e., an error).
3. Slab rule: If the structural component of a clash is a slab, then the clash type must be a legal intervention (i.e., a deliberate clash).
4. Wall rule: If the structural component of a clash is a wall and a load-bearing wall, the clash type must be a severe clash (i.e., an error); otherwise, it is a negligible clash.

Some rules involve complex spatial operations or require other supporting information to be processed. For example, the identification of clash position and wall type is required for Beam and Wall rules, respectively. The limited information provided in the report did not allow for corresponding calculations. Therefore, under a conservative consideration, this study simplified the aforementioned Beam and Wall rules as follows:

1. Simplified beam rule: If the structural component of a clash is a beam, then the clash type is a severe clash (i.e. errors).
2. Simplified wall rule: If the structural component of a clash is a wall, then the clash type is unknown.

Numbers in parentheses in Table 3 represent the numbers of clashes that are consistent with expert classification results. Because the simplified rule tends to be conservative, the prediction accuracy rate will be lower, especially for the Wall rule, which had an accuracy rate of only 15%, resulting in an overall classification accuracy of less than 60% (189/326).

A further modification was made to the Wall rule: “If the structural component of a clash is a wall, then the clash type is negligible (i.e., a deliberate clash).” This increases the overall accuracy rates to 87% (283/326). However, this modification may carry a potential risk of mis-classification

Table 3. Results of rule-based reasoning using simplified rules

Clash Type	Simp. beam rule	Column rule	Slab rule	Simp. wall rule	Total
Serious clashes	174 (154)	2 (2)			176 (156)
Neglectable clashes			12 (12)		12 (12)
Unknown				138 (21)	138 (21)
Accuracy	0.89	1.00	1.00	0.15	0.58

3.4 Feature selection and pre-processing for machine learning

As described in the literature review, the first step in developing a machine learning algorithm involves selecting features that are relevant to problem solving. In addition, because the development tool used in this study (i.e., scikit-learn) can only process numerical data, the nominal or text features must be coded as numerical features. Table 4 shows the selection and manipulation of each feature that we used for training machine learning algorithms. Among them, numerical features such as "Distance", "Floor-1", and "Floor-2" remain unchanged; "Clash Point" is the coordinate with a mixture of numbers and text and is separated into three numerical features, namely "Clash Point_x", "Clash Point_y", and "Clash Point_z"; As for "ItemType-1" and "ItemType-2", which are nominal features, one-hot encoding is performed separately, thus deriving six features as shown in Table 4.

In addition, to test whether the results of rule-based reasoning (see Section 3.3) contribute to the improvement of the prediction accuracy of the machine learning algorithm, another feature "Rule-Tag" was also added to label the result of the rule-based reasoning. Since this feature is nominal, one-hot encoding is also performed before the training process.

Table 4. Summary of feature selection and pre-processing

Original Feature	Data Type	Example Value	Revised Feature
Distance	numeric	-0.234	unchanged
Floor-1	numeric	2	unchanged
Floor-2	numeric	2	unchanged
Clash Point	text	x:-7370, y:-1171, z:24	Clash Point_x Clash Point_y Clash Point_z
ItemType-1	nominal	Pipes	{Pipes, Fittings}
ItemType-2	nominal	Framings	{Framings, Walls, Slabs, Columns}

3.5 Modeling by machine learning classifiers

The research team adopted a free software machine learning library, scikit-learn, as the implementation environment, which provides a rich set of tools for classification, regression and clustering using machine learning algorithms such as decision trees, SVM, k-NN, random forests, gradient boosting, k-means and DBSCAN. Researchers suggested that no single classifier can work well across all scenarios. Therefore, this study implemented three common single non-linear classifiers and three multiple classifiers of ensemble learning. We will evaluate the performances of those different classifiers later in Section 4.

Table 5 summarizes the manipulation of training classifiers by this study. In order to test and evaluate classifier performances, the research team randomly selected 30% of the entire 326 cases as testing dataset, i.e. 98 cases, while the rest of cases is further split into training and validation datasets with a proportion of 4:1 when applying k-fold cross-validation (k=5). For classifiers like kNN, SVM, Voting, the features of training dataset were standardized before the training process.

Table 5. Summary of machine learning manipulations

Measures	Description
Data splitting	7:3 (228 cases for training; 98 cases for testing)
Performance metric	Confusion matrix
Classifiers	Decision Tree, SVM, kNN, Voting, Bagging, Random Forest
Evaluation	k-fold cross-validation (k=5)

3.5.1 Single classifiers

1. Decision Tree

First, the research team implemented a classifier using decision tree algorithm with a maximum tree depth of 6 and a criterion of Gini impurity. Figure 5 shows its learning curves with the best f1-score of 0.92. Both the training and validation curves converge at a high prediction accuracy indicating the classifier doesn't under-fit the real situation. However, two curves converge with a large gap indicating the model may suffer from over-fitting.

2. K Nearest Neighbors (k-NN)

Next, a classifier using k-NN algorithm with 3 neighbors and a uniform weight. Figure 6 shows its learning curves with the best f1-score of 0.90. Both curves in Figure 6 also converge at a high prediction accuracy with a small gap indicating the classifier doesn't suffer from under-fitting, but a bit from over-fitting.

3. Support Vector Machine (SVM)

The last single classifier implemented in this study is using SVM algorithm with 1.0 penalty and a linear kernel. Figure 7 shows its learning curves with the best

f1-score of 0.90. Although the curves do not converge well as the previous two classifiers, its high prediction accuracy and small gap still indicate the classifier doesn't suffer from high variance nor high bias.

Figure 5. The learning curves of the classifier of a decision tree with a maximum depth of 6

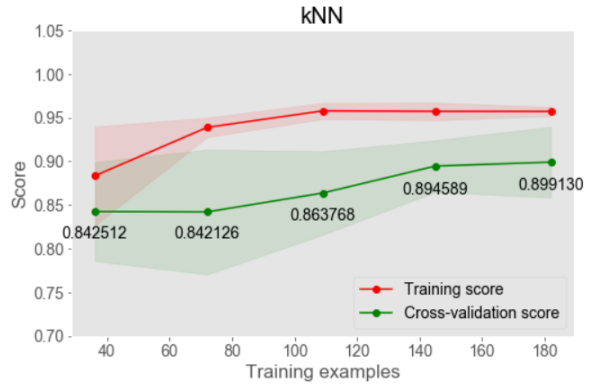


Figure 6. The learning curves of a k-NN classifier with 3 neighbors and a uniform weight

Figure 7. The learning curves of a SVM classifier with 1.0 penalty and a linear kernel

3.5.2 Multiple classifiers by ensemble learning

All single classifiers implemented previously have performed well in terms of prediction accuracy so the research team then moved forward to build up a set of classifiers following the principles of ensemble learning. The benefit of ensemble methods is to combine different classifiers into a multiple classifier that can often have a better predictive performance than each individual classifier alone. Those multiple classifiers implemented by this study include Voting classifier, Bagging classifier, and Random Forest classifier.

1. Voting

The first implemented ensemble learning algorithm is the Voting classifier, which combined three individual classifiers implemented previously, namely, a decision tree classifier, a k-NN classifier, and a SVM classifier, with a soft voting weight of 5:1:1, respectively.

Figure 8 shows its learning curves where both training and validation curves converge with a small gap at the best f1-score of 0.94 indicating the classifier doesn't suffer from under-fitting, but slightly over-fits.

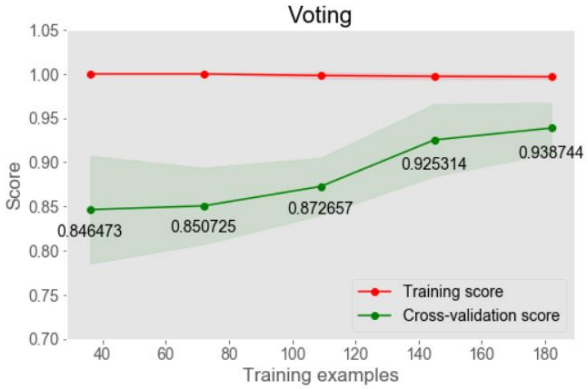


Figure 8. The learning curves of a Voting classifier combining a decision tree, k-NN, and SVM classifiers

2. Bagging

The research team then implement a Bagging classifier with 100 default estimators, where are decision tree classifiers. The learning curves shown in Figure 9 indicate that the classifier has gained a higher prediction accuracy (0.96) than any of previous classifiers. The convergence of both training and validation curves also indicates this classifier does not over-fit nor under-fit.

3. Random Forest

Finally, the research team implemented the commonest ensemble learning classifier, Random Forest,

with also 100 default estimators, or the decision trees. Figure 10 shows its learning curves with a very similar shape with the Bagging classifier.

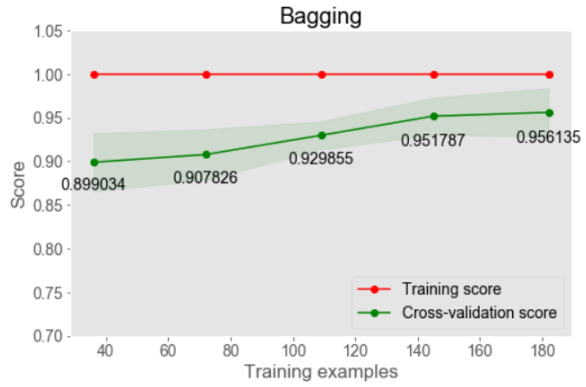


Figure 9. The learning curves of a Bagging classifier with 100 decision tree classifiers

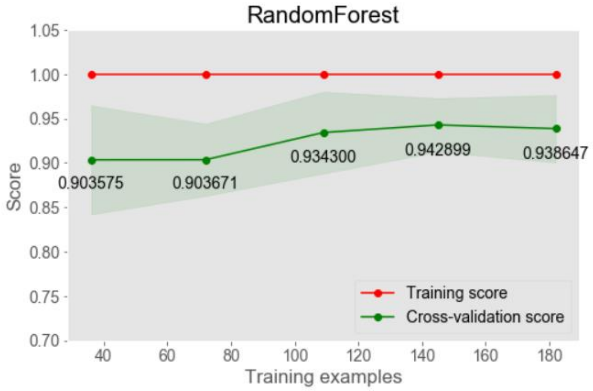


Figure 10. The learning curves of a Random Forest classifier with 100 decision tree classifiers

4 Results and Evaluation

In this section, the testing dataset of 98 cases is used to test the performance of those trained classifiers. The confusion matrix is adopted for the evaluation, which reports the counts of the true positive, true negative, false positive, and false negative predictions of a classifier, as shown in Figure 11. Three rates derived from the confusion matrix are recorded as the performance metric against all trained classifiers, including precision, recall, and f1-score, which are defined as in formulas (1), (2), and (3).

Besides those quantitative measurements, feature importance suggested by each classifier is also addressed to reveal which features explain more about the prediction results than others. Only the decision tree classifier and Bagging classifier are selected for evaluation.

		Predicted labels	
		True Positives (TP)	False Negatives (FN)
Actual labels		False Positives (FP)	True Negatives (TN)

Figure 11. The confusion matrix used to evaluate the performance of the trained classifiers

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$f1 - score = 2 * \frac{Precision * Recall}{TPrecision + Recall} \quad (3)$$

4.1 Results

Table 6 illustrate the testing results of 98 cases against the decision tree classifier with 6 levels. The average f1-score is 0.92 indicating a good predictive accuracy. Table 7 show the testing results of the bagging classifier with 100 decision trees. The average f1-score reaches an even better predictive accuracy of 0.94. Compared with the classification accuracy obtained by the rule-based reasoning, both classifiers can produce a much better predictive performance.

Table 6. Performance matrix of the decision tree classifier

	precision	recall	f1-score
Deliberate clashes	1	1	1.00
Errors	0.89	0.95	0.92
Unknown	0.78	0.58	0.67
Average	0.93	0.93	0.92

Table 7. Performance matrix of the Bagging classifier

	precision	recall	f1-score
Deliberate clashes	0.98	1.00	0.99
Errors	0.93	0.95	0.94
Unknown	0.80	0.67	0.73
Average	0.94	0.94	0.94

4.2 Evaluation

According to the confusion matrix of the decision tree classifier shown in Figure 12, none actual “errors” is classified as deliberate clashes or pseudo clashes, which provides a conservative prediction with a lower risk. In addition, the classifier filters out 42 “noises” (in

this case, deliberate clashes), accounting for 43% of all clashes from the original clash report, remaining only 57% to be inspected by human. Similarly, none actual "errors" is mis-classified by the Bagging classifier. The classifier also filters out 43% of clashes from the original clash report.

Figure 13 lists the top 6 features that have the higher importance to contribute the prediction results. Item type-1 "Framing" is the most discriminative feature in the dataset, which aligns with the result of rule-based reasoning in Section 3.3. Likewise, the feature importance suggested by the Bagging classifier presents a similar distribution to that by the decision tree classifier.

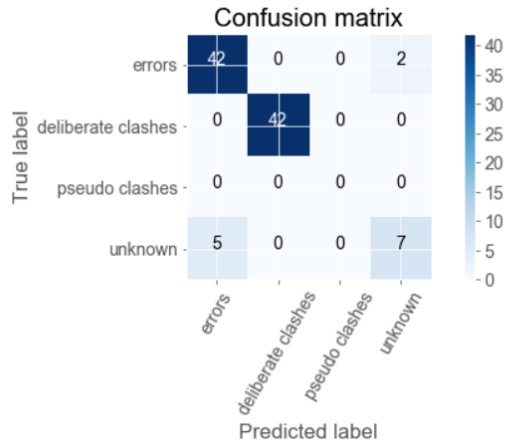


Figure 12. The confusion matrix of the decision tree classifier

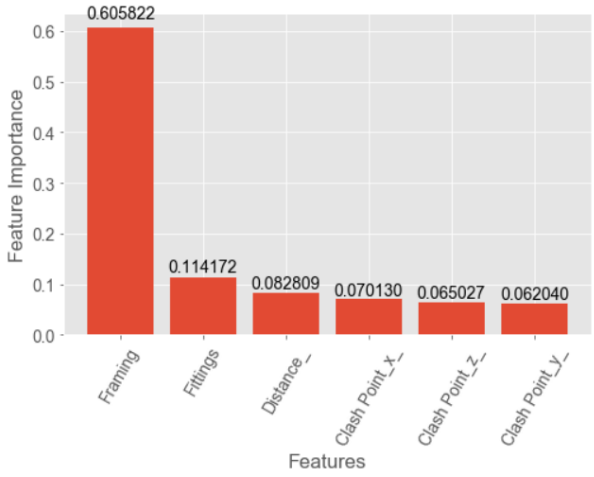


Figure 13. Feature importance suggested by the decision tree classifier

5 Conclusions

This study developed a tiny rule-based reasoning system to classify those conflicts detected by BIM software, which obtained a predictive accuracy of 58%. Then, both individual machine learning classifiers and multiple classifiers were also implemented to perform the same task. Preliminary testing results showed that both the decision tree classifier and 3 ensemble learning classifiers can obtain a much better predictive accuracy than the rule-based version.

However, the current model and its predictive performance can only be applied to this setting since the dataset used to train and test the machine learning classifiers only contains clashes between structural components and piping components. The dataset needs to be extended to include other MEP components, such as ducts, conduits, fire alarm devices, or lighting devices. Besides, most classifiers built in this paper have an over-fitting issue according to their learning curves. This could also be resulted from a relatively small number of training cases in our dataset. This limitation originates from that fact that labeling the dataset highly relied on human experts' experiences and labor work. All the mentioned issues remain to be solved in the future.

References

[1] E. A. Pärn, D. J. Edwards, Michael C.P. Sing, Origins and probabilities of MEP and structural design clashes within a federated BIM model, *Automation in Construction*, 85: 209-219, 2018.

[2] Lopez, R., et al., Design error classification, causation, and prevention in construction engineering. *Journal of Performance of Constructed Facilities*, 24(4): 399-408, 2010.

[3] J. Won, G. Lee. How to tell if a BIM project is successful: a goal-driven approach, *Automation in Construction*, 69: 34–43, 2016.

[4] Lin Y. C., Chou S. H. and Wu I. C. Conflict Impact Assessment between Objects in a BIM system. In *Proceedings of the 30th International Symposium on Automation and Robotics in Construction*, Montreal, Canada, 2013.

[5] H. Son, F. Bosche, C. Kim, As-built data acquisition and its use in production monitoring and automated layout of civil infrastructure: a survey, *Advanced Engineering Information*, 29 (2) (2015) 172–183.

[6] Raschka S. *Python Machine Learning*. Packt Publishing, Livery Place, 35 Livery Street, Birmingham B3 2PB, UK, 2015.