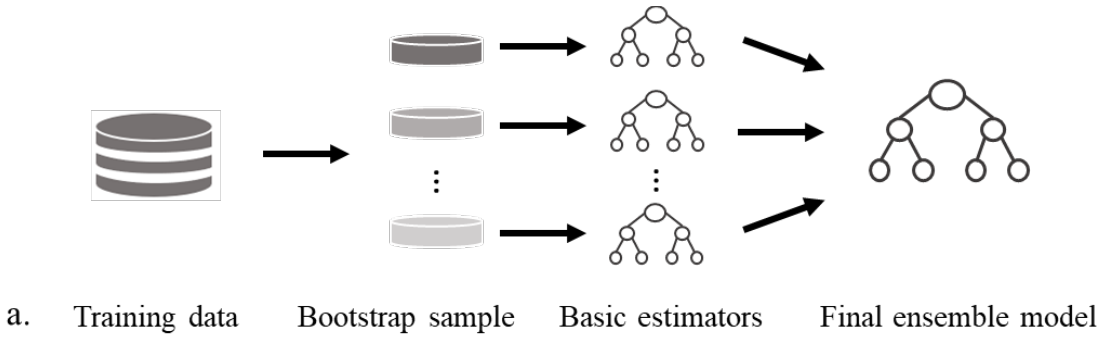
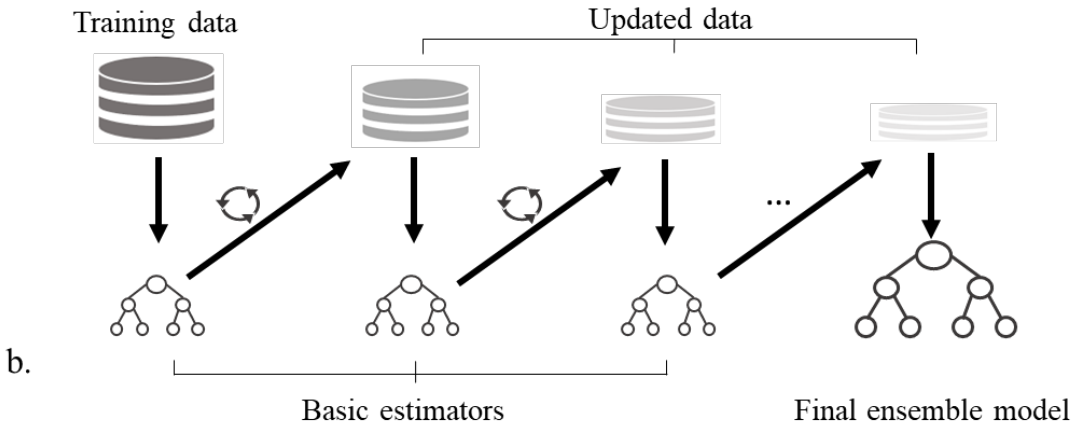
**Interpretable AI Explores Effective Components of CAD/CAM Resin Composites**

Hefei Li, Takahiko Sakai, Ainosuke Tanaka, Momoko Ogura, Chunwoo Lee, Satoshi Yamaguchi, Satoshi Imazato

**Appendix**




**Appendix Figure 1.** Schematic illustration of bagging and boosting approach. **a**. Bagging: Each basic model was trained from the bootstrap sample. The final ensemble model took

the average value as the predictions. **b.** Boosting: The training set was adaptively updating according to the performance of previous created basic estimators. Each basic model is created sequentially.

**Appendix Algorithm basis**

As discussed in the discussion section, in an ensemble, a set of base learners are trained to act together as a strong learner, thereby providing more accurate predictions (Marani and Nehdi 2020). Bagging and boosting are the two most frequently used approaches for constructing ensemble models. The schematic illustration of bagging and boosting were shown in **Appendix Figure 1**. The ensemble algorithms used in this study including RF, ET, GBDT, LightGBM and XGBoost. RF and ET algorithms use the bagging approach, and the rest of the three algorithms use boosting approach. A brief introduction of the five algorithms implemented in this study is provided below.

**Random Forest (RF)**

Breiman developed the random forest algorithms for both regression and classification purposes (Breiman 2001). As the base constituents of the ensemble are tree-structured predictors, and since each of tree is constructed using an injection of randomness, the method is called “random forests” (Segal 2004). This tree-structured predictor is called a decision tree (DT). It by continuously splitting data based on a certain parameter and a tree having decision nodes and leaf nodes is obtained. A bootstrap sample was used to train each decision tree as the basic estimator. The number of the decision tree and how

the tree grows were controlled by the hyperparameters in the algorithm. Each decision tree will give a predicted value for the regression problem. The final output value in RF model is the unweighted average of all the predicted values obtained from all decision trees, which could be expressed as follows:

$$\hat{Y} = \frac{1}{t}\sum_{i=1}^{t} \hat{y}^{(i)}$$

Where $\hat{Y}$ is the output value from RF model, $t$ is the number of decision tree and $\hat{y}$ is the prediction value of each decision tree.

**Extra Tree (ET)**

Geurts developed the extra tree algorithm as an extension from the random forest algorithm (Geurts et al. 2006). ET is also a tree-based ensemble algorithm, which has some differences from that of RF. ET uses the whole training dataset to train each decision tree, while RF uses the bootstrap sample. RF obtains the best feature and value to split to two branches for decision tree by calculating specific mathematic parameter. While ET obtains the splitting feature and value completely randomly.

**Gradient Boosting Decision Tree (GBDT)**

GBDT is a widely used machine learning algorithm proposed by Friedman which integrates multiple DTs into a strong final ensemble model using the boosting approach (Friedman 2001; 2002; Guelman 2012). The training set was adaptively updating

according to the performance of previous created DT (predecessor). Each basic model is created sequentially trying to correct its predecessor. The process could be expressed as follows:

$$f_t(x) = f_{t-1}(x) + \alpha h_t(x)$$

Where $f_t(x)$ is the GBDT model, $h_t(x)$ is the basic model (DT), and $\alpha$ is called the learning rate, which is a regularization parameter (Marani and Nehdi 2020). It scales the length of the step for finding the optimum solution. Larger alpha leads to faster iteration speed, while smaller alpha leads to lower iteration speed, which is more possible to find the optimum solution, but it requires more computational cost. Except from regularization through shrinkage of the contributed basic models, randomness was also incorporated as an integral part of the fitting procedure.

**eXtreme Gradient Boosting (XGBoost)**

Under the framework of GBDT, XGBoost has been proposed with higher computation efficiency and better capability to deal with overfitting problems (Chen and Guestrin 2016). There are some main differences between GBDT and XGBoost, one is the different definition in objective function, in which a regularization function was implemented alongside with the loss function.

$$Obj^{(t)} = \sum_{k=1}^{n} l(\bar{y}_i, y_i) + \sum_{k=1}^{t} \Omega(f_i)$$

Where $l$ is the loss function, $n$ is the number of observations used and $\Omega$ is the regularization term to prevent overfitting issue (Fan et al. 2018). The other main difference is that GBDT only uses the first-order derivative information of the loss function when optimizing the objective function, while XGBoost performs a second-order Taylor expansion on the loss function, and both the first-order and second-order derivatives are used (Guelman 2012; Marani and Nehdi 2020).

**Light Gradient Boosting Machine (LightGBM)**

LightGBM was proposed by Microsoft still based on GBDT. When handling with big data, the conventional implementations of GBDT need to scan every feature for all instances to estimate the split point which are very time-consuming. To solve the above issue, LightGBM uses an improved histogram-based algorithm to improve training speed and space efficiency. Compared with the traditional method level (depth)-wise growing of DTs in GBDT, LightGBM uses a leaf-wise generation strategy to reduce training data and exclusive feature bundling technique to reduce the feature numbers. More details about the algorithm are in (Ju et al. 2019; Ke et al. 2017).

**Appendix Table 1.** Tuned hyperparameters for the implemented algorithms

| Models | Tuned hyperparameters |
|---|---|
| RF | max_depth=6, n_estimators=280, min_samples_split=3, max_features=14, random_state=100 |
| ET | max_depth=6, n_estimators=41, random_state=3 |
| GBDT | n_estimators=120, max_depth=5, learning_rate=0.04, random_state=4, min_weight_fraction_leaf=0.04 |
| LightGBM | boosting_type='dart', objective='regression', importance_type='gain', n_estimators=141, max_depth=3, learning_rate=0.36, min_child_samples=5, reg_alpha=0.01,random_state=2 |
| XGBoost | n_estimators=127,max_depth=3,learning_rate=0.55,booster='dart', min_child_weight=6, reg_alpha=0.01, random_state=3 |

**Detailed illustration of Fig. 2**

There are 15 features used in this study as shown in **Appendix Table 2**. For feature 1 to feature 14, 1 and 0 were used to indicate this component is containing or not. For feature 15, that is, filler content, we set 24 different weight percentages from 62 to 85 (wt%) for exhaustive search. After combination, we got $2^{14}$×24 (393,216) different combinations of the features. Part of the combinations were demonstrated in Appendix Figure 2:

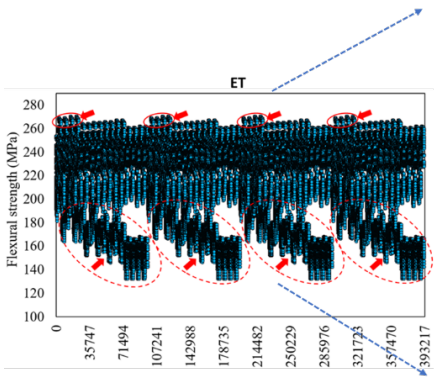
| | | |
|---|---|---|
| … | … | … |
| 212757 | (1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 82) | 269.4939477 |
| 212758 | (1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 83) | 269.4939477 |
| 212759 | (1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 84) | 269.4939477 |
| 212760 | (1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 85) | 269.4939477 |
| … | … | … |
| 370753 | (1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 62) | 132.2225308 |
| 370754 | (1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 63) | 132.2225308 |
| 370755 | (1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 64) | 132.2225308 |
| 370756 | (1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 65) | 132.2225308 |
| … | … | … |

**Appendix Figure 2.** The combination numbers are from 1 to 393,216. Some of the numbers are shown in the red square. For each combination, it represents a specific composition for the CAD/CAM RCB that was formed by 0 and 1 as shown in the yellow

square. By checking the corresponding feature name, we could know what is containing in this composition and its filler content. In the meantime, the model will predict the flexural strength for each combination, which was shown in the blue square.

The horizontal axis of **Fig 2** represents the combination number (red square) and the vertical axis represents the corresponding flexural strength predicted by the model (blue square). Take Figure 2c for example, the predictions of the low prediction group shown in red dotted line varied from 132.2 MPa to 180.1 MPa (70,837 compositions). By contrast, the high prediction group shown in red circle ranged from 256.6 MPa to 269.5 MPa (1,884 compositions). Next, the composition differences between these two groups were compared by checking how many times does a specific component contained among those compositions and calculate how much percentage of the compositions contained this specific component, the detailed results are shown in the **Appendix Figure 3**.

a.



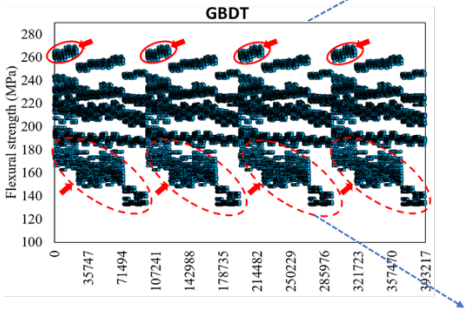

| Filler | Percentage(%) |
|---|---|
| $SiO_2$ | 100.0 |
| $ZrO_2$ | 50.0 |
| $ZrSiO_4$ | 32.5 |
| Micro-fumed silica | 5.5 |
| Barium glass | 56.5 |
| $Al_2O_3$ | 50.0 |
| Methacrylate mixed filler | 100.0 |
| $SiO_2$-$ZrO_2$ filler | 42.3 |

| Monomer | Percentage(%) |
|---|---|
| UDMA | 54.4 |
| Bis-MEPP | 44.2 |
| TEGDMA | 0 |
| NPGDMA | 50.1 |
| Bis-GMA | 0 |
| Bis-EMA | 54.4 |
| **Filler content (wt%)** | |
| 81-85 | |

| Filler | Percentage(%) |
|---|---|
| $SiO_2$ | 100.0 |
| $ZrO_2$ | 49.6 |
| $ZrSiO_4$ | 67.6 |
| Micro-fumed silica | 58.3 |
| Barium glass | 50.0 |
| $Al_2O_3$ | 60.5 |
| Methacrylate mixed filler | 45.8 |
| $SiO_2$-$ZrO_2$ filler | 50.3 |

| Monomer | Percentage(%) |
|---|---|
| UDMA | 54.3 |
| Bis-MEPP | 50.2 |
| TEGDMA | 100 |
| NPGDMA | 48.9 |
| Bis-GMA | 50.1 |
| Bis-EMA | 48.8 |
| **Filler content (wt%)** | |
| 62-69 | |

b.




| Filler | Percentage(%) |
|---|---|
| $SiO_2$ | 100.0 |
| $ZrO_2$ | 51.1 |
| $ZrSiO_4$ | 27 |
| Micro-fumed silica | 1.1 |
| Barium glass | 61.6 |
| $Al_2O_3$ | 51.2 |
| Methacrylate mixed filler | 54.2 |
| $SiO_2$-$ZrO_2$ filler | 46.3 |

| Monomer | Percentage(%) |
|---|---|
| UDMA | 53.9 |
| Bis-MEPP | 49.3 |
| TEGDMA | 0 |
| NPGDMA | 50.7 |
| Bis-GMA | 0 |
| Bis-EMA | 53.9 |
| **Filler content (wt%)** | |
| 82-85 | |

| Filler | Percentage(%) |
|---|---|
| $SiO_2$ | 100.0 |
| $ZrO_2$ | 50.1 |
| $ZrSiO_4$ | 59.2 |
| Micro-fumed silica | 60.6 |
| Barium glass | 51.3 |
| $Al_2O_3$ | 50.1 |
| Methacrylate mixed filler | 50.1 |
| $SiO_2$-$ZrO_2$ filler | 50.2 |

| Monomer | Percentage(%) |
|---|---|
| UDMA | 52.1 |
| Bis-MEPP | 53.5 |
| TEGDMA | 53.5 |
| NPGDMA | 50.7 |
| Bis-GMA | 50.6 |
| Bis-EMA | 50.7 |
| **Filler content (wt%)** | |
| 62-69 | |

**Appendix Figure 3.** The detailed results of exhaustive search for ET and GBDT model. Larger figures of ET and GBDT model was shown in the manuscript **Fig 2b** and **c**.

**a**. The detailed analysis of the exhaustive results predicted by ET model. The predictions of the bottom group shown by red dotted circle ranged from 132.0 MPa to 180.1 MPa, and those of the top group shown in red circle ranged from 263.7 MPa to 270.0 MPa. The tables pointed out by the blue dotted arrow indicates the percentage of the compositions contained the specific filler or monomer in both groups. Among the compositions in high prediction group, the filler content ranged from 81 to 85(wt%) and ranged from 62 to 69(wt%) in the low prediction group.

**b**. The detailed analysis of the exhaustive results predicted by GBDT model. The predictions of the bottom group shown by red dotted circle ranged from 132.2 MPa to 180.1 MPa, and those of the top group shown in red circle ranged from 256.6 MPa to 269.5 MPa. The tables pointed out by the blue dotted arrow indicates the percentage of the compositions contained the specific filler or monomer in both groups. Among the compositions in high prediction group, the filler content ranged from 82 to 85(wt%) and ranged from 62 to 69(wt%) in the low prediction group

**Appendix references**

Breiman L. 2001. Random forests. Mach Learn. 45(1):5-32.

Chen TQ, Guestrin C. 2016. Xgboost: A scalable tree boosting system. Kdd'16: Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining.785-794.

Fan JL, Wang XK, Wu LF, Zhou HM, Zhang FC, Yu X, Lu XH, Xiang YZ. 2018. Comparison of support vector machine and extreme gradient boosting for predicting daily global solar radiation using temperature and precipitation in humid subtropical climates: A case study in china. Energ Convers Manage. 164:102-111.

Friedman JH. 2001. Greedy function approximation: A gradient boosting machine. Ann Stat. 29(5):1189-1232.

Friedman JH. 2002. Stochastic gradient boosting. Comput Stat Data An. 38(4):367-378.

Geurts P, Ernst D, Wehenkel L. 2006. Extremely randomized trees. Mach Learn. 63(1):3-42.

Guelman L. 2012. Gradient boosting trees for auto insurance loss cost modeling and prediction. Expert Syst Appl. 39(3):3659-3667.

Ju Y, Sun GY, Chen QH, Zhang M, Zhu HX, Rehman MU. 2019. A model combining convolutional neural network and lightgbm algorithm for ultra-short-term wind power forecasting. Ieee Access. 7:28309-28318.

Ke GL, Meng Q, Finley T, Wang TF, Chen W, Ma WD, Ye QW, Liu TY. 2017. Lightgbm: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems 30 (Nips 2017). 30.

Marani A, Nehdi ML. 2020. Machine learning prediction of compressive strength for phase change materials integrated cementitious composites. Constr Build Mater. 265.

Segal MR. 2004. Machine learning benchmarks and random forest regression. UCSF: Center for Bioinformatics and Molecular Biostatistics.1-14.