**Evaluation of machine learning and empirical models for concrete creep compliance prediction**

Viet-Linh Tran1,2 and Jin-Kook Kim1\*

1 Department of Civil Engineering, Seoul National University of Science and Technology, 232 Gongneung-ro, Nowon-gu, Seoul 01811, Republic of Korea

2 Department of Civil Engineering, Vinh University, Vinh 461010, Vietnam

\* Corresponding author : [jinkook.kim@seoultech.ac.kr](mailto:jinkook.kim@seoultech.ac.kr).

This study uses widely used tree-based ML algorithms, including decision tree (DT), Random forest (RF), adaptive gradient boosting (AGB), gradient boosting (GB), and extreme gradient boosting (XGB), to compare with the GWO-LGB model for concrete creep compliance prediction. The following section will briefly introduce these ML algorithms.

*S.1. Decision Tree*

DT [1] creates the tree-like structure from the training data. This process includes three critical steps. First, the training data from the root node is recursively partitioned into branches or subsets by the Gini or entropy index criterion [2]. Then, tree pruning is applied to handle the overfitting issue during tree creation. Finally, a smoothing operation compensates for the sharp discontinuities between adjacent linear models at the pruned tree leaves. This splitting process ends when no split can be found to reduce the used metrics or the maximum depth of the tree is reached. The advantage of the DT is that it can handle both categorical and numeric data. Hence, the DT is relatively easier than other models. However, the disadvantage of the DT is model instability. In addition, the DT model is prone to overfitting the data. Therefore, more complex tree-based algorithms are often considered more reliable.

*S.2. Random Forest*

RF [3] is based on several decision trees. It produces multiple independent DTs from the bootstrap aggregating technique and each DT is entirely random; therefore, the RF algorithm is less prone to overfitting and has improved generalization ability. First, multiple sampling rounds are conducted using bootstrap sampling. During each sampling round, a subset of input features is randomly chosen. Then, several individual DTs can be trained based on each new dataset and the corresponding features. The final prediction of the RF algorithm is obtained by averaging the DT predictions. The advantage of the RF is that it significantly reduces instability. However, the drawback of the RF is the inherent overfitting of each DT. To prevent overfitting, RF creates multiple decision trees or through regulation.

*S.3. Adaptive Gradient Boosting*

AGB is the first boosting algorithm developed by Freund and Schapire [4]. In AGB, each sample in the dataset is initially assigned equal weight. When the first weak learner is trained, it will give more priority to samples that it incorrectly predicts, while decreasing the weights of others. Following the performance of the previous weak learner, training sample weights are updated each iteration. The process is stopped once the predetermined error or the specified number of iterations has been reached.

*S.4. Gradient Boosting*

It is similar to AGB in that GB [5] combines weak learners to integrate strong learners and adds predictors sequentially to compensate for previous errors. However, GB improves the performance of the model by gradient descent. The GB produces the new base learners to be the maximum correlation with the negative gradient of the loss function. Moreover, GB creates a new predictor from the residual error by the previous one.

*S.5. Extreme Gradient Boosting*

XGB [6] can be seen as an improved algorithm of GB. The GB uses the first-order derivative for optimization, while XGB employs a second-order Taylor expansion on the loss function and performs both first-order and second-order derivatives. Notably, the loss function of XGB adds a regularization term to prevent overfitting and reduce model complexity [7]. In addition, XGB can automatically select the best default splitting and process missing values.

**References**

[1] R.A.O.L.B.J.F. C.J. Stone, Classification and regression trees Boca Raton, Chapman an, 1984.

[2] A.J. Ferreira, M.A.T. Figueiredo, Boosting Algorithms: A Review of Methods, Theory, and Applications, in: Ensemble Mach. Learn., Springer US, Boston, MA, 2012: pp. 35–85. https://doi.org/10.1007/978-1-4419-9326-7\_2.

[3] L. Breiman, Random forests, Mach. Learn. 45 (2001) 5–32. https://doi.org/10.1023/A:1010933404324.

[4] Y. Freund, R.E. Schapire, A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting, J. Comput. Syst. Sci. 55 (1997) 119–139. https://doi.org/10.1006/jcss.1997.1504.

[5] J.H. Friedman, Greedy function approximation: A gradient boosting machine., Ann. Stat. 29 (2001) 1189–1232. https://doi.org/10.1214/aos/1013203451.

[6] T. Chen, C. Guestrin, XGBoost: A scalable tree boosting system, in: Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., ACM, New York, NY, USA, 2016: pp. 785–794. https://doi.org/10.1145/2939672.2939785.

[7] B. Kim, D.-E. Lee, G. Hu, Y. Natarajan, S. Preethaa, A.P. Rathinakumar, Ensemble Machine Learning-Based Approach for Predicting of FRP–Concrete Interfacial Bonding, Mathematics. 10 (2022) 231. https://doi.org/10.3390/math10020231.