# Advanced Machine Learning Approaches for Prediction of Compressive Strength of Sustainable Concrete

**Karthik Sai Repala[1], Dr. Kusuma Sundara Kumar[2]**
[1]UG Scholar, Department of CSE-AIML, [2]Professor, Department of Civil Engineering,
Ramachandra College of Engineering-|Eluru, Andhra Pradesh, India
[1]karthiksai22102005@gmail.com, [2]skkusuma123@gmail.com

---------------------------------------------------------------------***---------------------------------------------------------------------

## ABSTRACT

Predicting the compressive strength of concrete using SCMs is difficult since the relationships between the mix components are not linear. The fact that these parts interact with one another in complex ways is one explanation. In order to address this, five distinct kinds of math-based tools were evaluated. Random Forest, which aggregates findings, was one of these methods. Gradient boosting is an alternative method that iteratively improves upon previous mistakes. Next up is XGBoost, a popular choice for situations when speed is crucial. As an alternative to making a sudden leap, AdaBoost refines its predictions incrementally. The Multiple Linear Regression method, on the other hand, is exclusive to linear patterns. How was the data utilized? A database containing 1030 distinct recipe mixtures, with 80% taught and 20% tested. Performance wasn't evaluated based on intuition but rather four metrics: R squared demonstrated fit, root mean square error highlighted the significance of large mistakes, mean absolute error quantified the average amount of errors, and percentage deviation demonstrated the extent to which estimations fell off. The reliability and consistency of the predictions were shown numerically. Out of all the techniques that were tested, boosting emerged as the clear winner, easily outperforming bagging and conventional linear models. With a test R squared value of 0.927, XGBoost achieved the best score. How does SCM concrete react under pressure? Tangled designs are much easier to handle with these tools. When the issue fights back fiercely, a steady, step-by-step improvement might emerge victorious.

Keywords – ML Models, SCM concrete, Random Forest, Gradient Boosting, XGBoost

## 1. INTRODUCTION

Environmentally aware decisions drive emerging building styles. Demand has led to the widespread use of fly ash and crushed granulated blast furnace slag in modern concrete mixes. Regular cement is gradually disappearing from recipes. Changing it out causes a decrease in $CO_2$ levels. These improvements significantly increase the usable range of raw materials [1]. This makes it simpler to have a long and healthy life. It becomes more difficult to forecast the mix strength. How long it takes to cure, how particles behave, the functions of aggregates, and how much water are all factors in how well it holds up. Small changes got big news everywhere. The contents aren't the only determinant of its performance. This is where conventional mathematical methods fail. Connections among pieces are commonly overlooked in combinations with a lot of variation [2]. Using data-based approaches to record materials' behavior under stress is one way to circumvent these limitations. Machine learning discovers hidden patterns directly from measurements, rather than relying on fundamental ideas. However, most current studies only examine a single algorithm [3]. It is difficult to determine the optimal approach due to a lack of comprehensive laboratory data. This is why findings are often inadequate when extrapolated from specific examples [4-6].

Consistent testing of different machine learning approaches using the same data pool is one way to look about SCM-based concrete. The performance of straight-line fits compared to group-vote methods is important when dealing with large, realistic mixtures. Which technologies are most reliable in everyday usage may be shown by looking at what occurs over thousands of blend cases [7–10]. One conventional statistics model, two collaborative voting tree ensembles, and two incrementally improving while correcting mistakes along the way are even out by a collection of 1030 mix designs [11–13]. Differences are logical, not the result of setup gimmicks, since each run follows matching rules. Changes in mistake magnitude and reaction to out-of-the-ordinary inputs are shown by the results, which do not indicate winners but rather behaviors under stress. Expectations for future, more environmentally friendly mixes are shaped by learning patterns here [14–15].

## 2. DATASET AND PARAMETERS

One thousand thirty mix designs pulled from past lab reports made up the data pool. From old research papers came every sample used here. Different blends fill the set - some aged longer, others mixed fresh. Not just one type of recipe shows up across these entries. Eight things stood out as central to how strong each batch got. Cement amount mattered, along with how much slag went in. Fly ash played a role, just like the quantity of water added. Superplasticizer levels shifted outcomes more than expected. Coarse stone volume counted, so did the finer sand portion. Time spent curing changed results every single time. These factors link straight to how concrete hardens and binds. Hydration speed depends on them; also, how dense the mixture becomes. Pozzolan activity rises when certain materials replace part of the cement. Variety in replacements kept the patterns from getting predictable. The full collection reflects real differences seen in actual mixes. No artificial stretch was needed to cover common construction cases. Concrete's ability to resist pressure, recorded after set periods of hardening, serves as the main value predicted here. Patterns within the organized collection of data support solid model practice - each ingredient blend ties somehow to how strong the final piece becomes.

## 3. METHODOLOGY

This paper follows a straightforward, step-by-step methodology based on actual data and looks at how well different learning algorithms predict the strength of blended concrete. Although each of the five algorithms was constructed in a somewhat different way (one was linear while the others used groups or sequences), they were all evaluated in tandem using the same parameters. When all other factors, such as data, cleaning procedure, training/testing split, and scoring criteria, remain constant, any discrepancies in the results directly relate to the learning process of the individual models. Results are reflective of models' reasoning abilities alone, unaffected by changes in measurement or preparation, as all models were subject to the same inputs and tests.

The batch of 1030 concrete mix records was promptly organized and cleaned up. Carefully, the outcomes of the inputs' matching strengths were separated. It was split in half: 80% for pattern learning and 20% for prediction verification. Aside from that, no changes were made, no copies were made, and no adjustments were made at all. Completely for the purpose of preserving the integrity of each sample.Algorithms may sense patterns in very different ways. To explore how more traditional methods handle complex supply chain information, a test was first run using Multiple Linear Regression. Random Forest plants many trees together, minimizing uncertainty by blending their voices, rather than leaving them alone. In Gradient Boosting, XGBoost, and AdaBoost, trees are stacked instead of grown together, with each tree repairing the one before it. Their fundamental principles determine how accurately they foretell the results of green concrete when tested side by side in the same manner.

### *A: Preprocessing Data*

The 1030 concrete mix entries were organized in a dependable and consistent manner before the models were built. Separate from the measured outcome—the strength of the concrete—were the mix components and curing time. Randomly selected in the same manner for each run, eight out of 10 instances were used to train the system, while the other two were kept for further checks. The divide allows for the evident emergence of patterns during learning while yet ensuring that performance is properly tested when presented with fresh cases. Since they do not take into account changes in variable size or units, tree techniques such as Random Forest, Gradient Boosting, XGBoost, and AdaBoost do not need scaled features. Natural handling of varying data scales is accomplished via splitting rules within trees. The original numerals were preserved, ensuring that their practical significance in certain contexts was maintained. For the sake of objectivity, all models followed this identical preparation procedure. Treating each approach equally was essential for fair assessment.

### *B. Example Models*

Five separate approaches, influenced by different learning styles such as straight-line modeling, group consensus, or step-by-step improvement, emerged as dominant approaches to forecast compressive strength in SCM-based concrete. They reasoned in different ways, yet they all sought the same goals without stealing from one another.

### *1.Random Forest (Bagging):*

Random Forest (Bagging): One perspective is that it is a network of smaller models that share the original data but use different portions. These trees are constructed independently and provide different outcomes, which are then blended. They don't simply follow one decision-making process; their combined output determines the outcome. When you average those results, any uncertainty in any one of them disappears. It works well with complex patterns, including those in which variables unexpectedly spiral around one another. What you have usually works just well, so there's no need to alter it much before feeding inputs. Because extremes tend to cancel each other out among members, there is less jitter in forecasts. If you work together as a team, you won't be able to focus on the strange characteristics of any one case.

### *2.Gradient Boosting (Boosting):*

Gradient Boosting, sometimes known as boosting, is a method where trees are trained one after the other, with each iteration learning from and improving upon the previous one. Because each round builds on the one before it, mistakes don't pile up but rather grow smaller as time goes on. Surprisingly, the cunning curving patterns hidden in SCM data are easily handled when approached piecemeal. Through a series of incremental adjustments, the pattern gradually emerges.

### *3.XGBoost (Advanced Boosting):*

What sets XGBoost (Advanced Boosting) apart from the competition? For models to avoid learning from noise, an internal penalty mechanism is used. Working efficiently across several cores allows for faster processing times. Smartly tuning each step forward, rather than just adding them, is its approach. When making numerical predictions, many people initially turn to it because of how well it handles complicated real-world columns. The way it handles splits and limits depth ensures that complexity is handled rather than disregarded. Efficiency isn't an afterthought; it pushes all the layers forward. Its reputation was built subtly, by actions rather than rhetoric.

### *C. Measures for Assessment*

Although not all predictions were accurate, they were all subjected to four common mathematical tests. How about R squared? That one shows the degree to which the dots adhere to the line. Root mean square error is the most sensitive to mistakes that are becoming larger. Mean absolute error simply adds up the gaps without skipping signs. Accuracy may be better understood by calculating average errors as percentages. When you use both, you can see how close the predictions are to the real figures. The projected values are in good agreement with the actual concrete strength if the value is near to one. The amount of strength variation that the model successfully explains is what it represents.

To evaluate a technique's performance, one may compare its results on training and test data; this reveals how well the approach deals with new cases. Recurring results indicate dependability rather than batch-specific random chance. When a method consistently produces the desired results, confidence grows that it is capable of more than just remembering past patterns. Different datasets provide different results, which shows how readily new information may confuse a model. People who are able to adapt to new circumstances are less reliant on the details of the past.

Using five different approaches, we may see how different pedagogical approaches impact predictions for SCM-made concrete. Methods based on sequential improvement, collaborative work, or linear thinking fall under this category. Different perspectives on learning patterns are reflected in each approach. They react differently to data indications rather than assuming one approach fits all. Their decisions show the importance of structure and adaptability in predicting strong results.

1. Random Forest (Bagging): Cooperative trees may be very effective at times. At the same time, a forest constructs several decision routes, each of which is fed randomly selected data points. Each tree makes a unique error since it can only observe a portion of the scene. The responses are aggregated into a single group estimate as they arrive. This arrangement results in less swing than when a single split-heavy model is used independently. Entwined patterns? It

becomes simpler to capture them. It succeeds even when inputs are twisted in strange ways, all without requiring extensive pre-processing reshaping.

2. Gradient Boosting, often known as boosting, is a method whereby one tree fixes the faults made by the previous tree. Over time, remaining mistakes decrease as a result of successive learning cycles. This method is useful for highlighting complicated twists in SCM data.

3. A Look at XGBoost (Advanced Boosting): What Sets It Apart? It employs clever adjustments, such as complexity penalties built in, to distribute tasks among processors and handle complicated datasets with ease. Both accuracy and speed are improved. Many consider it the best boosted model for numerical prediction since it prevents overfitting while capturing intricate relationships between variables.

## 4. RESULTS AND DISCUSSIONS

### *A. Table for Comparing Performance*

Results for both training and test data may be shown by comparing the accuracy of each model's predictions using the selected metrics. Looking at it this way, it's easy to see which approach performs better with the same actual data set and the same parameters. The data reveal what really transpired behind the scenes in a way that makes each strategy stand out. Comparing results side by side allows one to more easily understand discrepancies without having to speculate as to their causes. When results are presented without any additional noise, what is most important becomes apparent. Upon examination of the data, it is evident that the Multiple Linear Regression model has challenges in dealing with the intricate patterns seen in SCM-blended concrete, as it exhibits the lowest $R^2$ scores and the most errors. Due to its multi-decision-tree merging mechanism, Random Forest significantly outperforms the previous model in terms of both reliability and variability. The fact that it does worse on test data than training data suggests that overfitting is becoming a problem.

Following closely after is Gradient Boosting, which maintains a strong position in terms of prediction quality. Notable is XGBoost's dominant performance, which includes the lowest mistakes and best test $R^2$. Impressive strength and consistency are shown while dealing with unknown data. Although MLR and Random Forest are both beaten by the trio of boosters, AdaBoost falls somewhat behind in terms of accuracy. Though not all of them succeed to the same extent, every approach improves upon prior models. Boosting, as opposed to bagging or straight-line approaches, clearly manages complex patterns in SCM concrete, as seen in the table.

TABLE I Performance Metrics

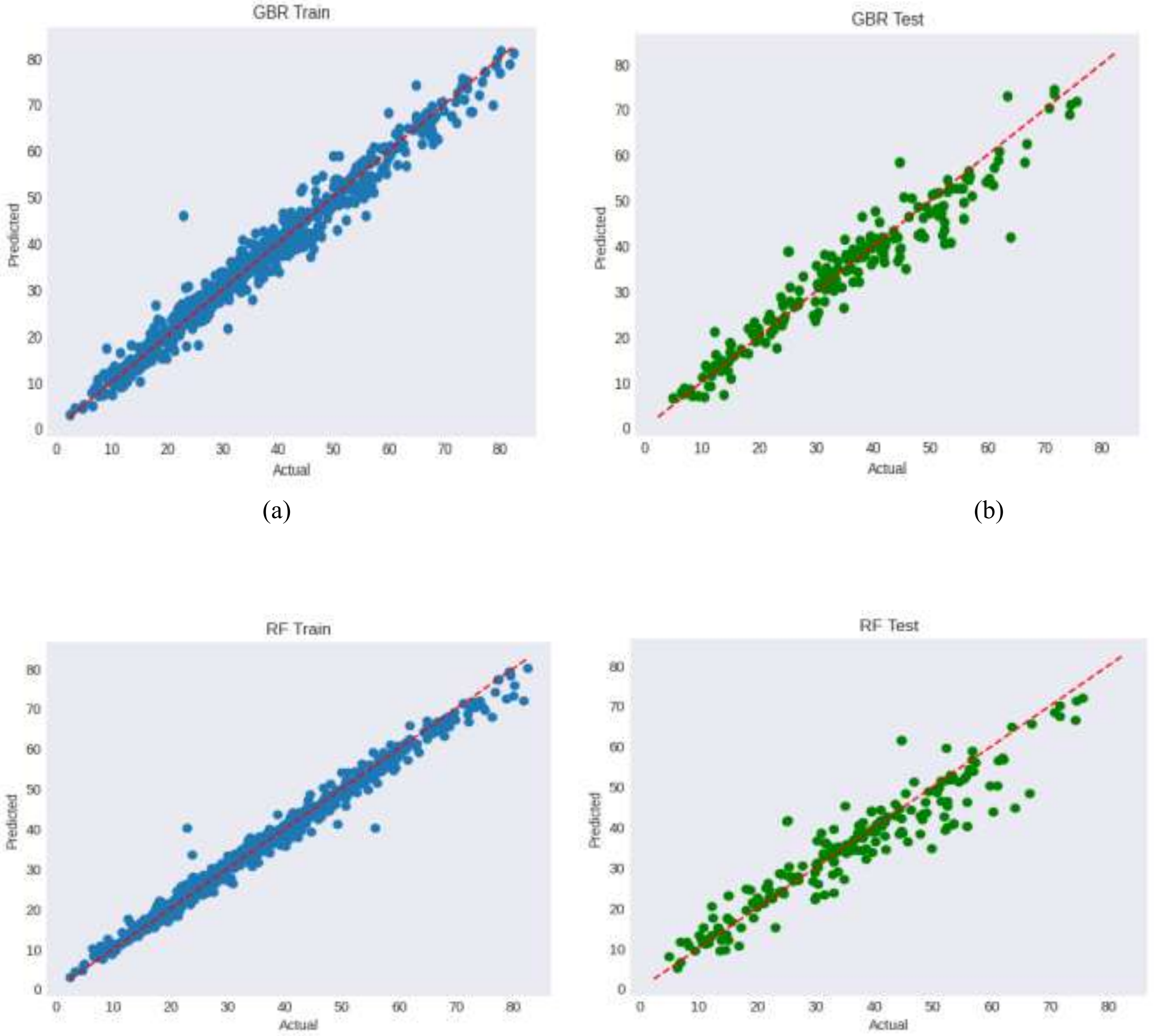
| **Model** | **Train $R^2$** | **Test $R^2$** | **Test RMSE** | **Test MAE** | **Test MAPE (%)** |
|---|---|---|---|---|---|
| Random Forest | 0.9872 | 0.8817 | 5.5223 | 3.7777 | 12.39 |
| Gradient Boosting | 0.9742 | 0.9155 | 4.6655 | 3.3018 | — |
| XGBoost | 0.992 | 0.927 | 4.329 | 3.013 | — |
| AdaBoost | ~0.81 | ~0.74 | ~8.2 | ~6.7 | — |
| MLR | 0.610 | 0.628 | 9.7967 | 7.745 | 29.27 |

### *B: Important Notes*

When comparing approaches using SCM-blended concrete data, the results reveal clear gaps, which is not unexpected.Water mixing, delayed pozzolan work, wetness fluctuations, and extended cure periods all contribute to these mixes' twisted alterations, which are too complex for straight-line arithmetic to unravel. It makes use of a large number of diversely trained decision trees simultaneously. It doesn't depend on a single route but instead uses all those trees to collect predictions. The total estimate is likely to remain constant as each tree receives a garbled copy of the data. Something is still lacking, however, when practice outcomes do not match training time. Sure, bagging helps, but data complexity still gets through. Boosting improves predictions step by step by fixing mistakes. Repetitive adjustments bring complex data shapes into focus, but not all at once. XGBoost is exceptional because it performs very well on test data, has small margins of error, and produces good R² values. Despite the lack of showy gimmicks, Gradient Boosting maintains a respectable position.

### *C. Why Boosting Wins*

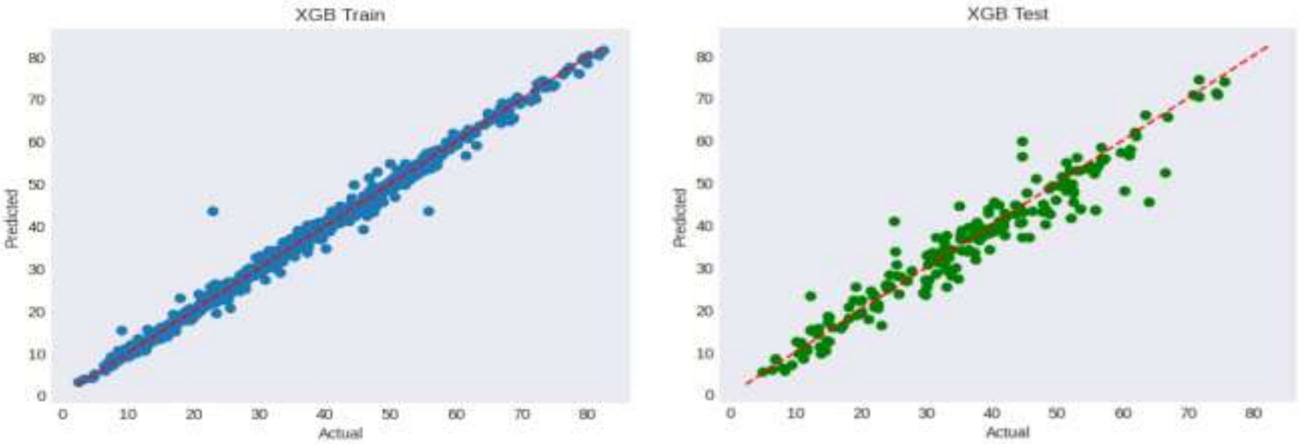
One after another, models take shape, each stepping in to fix what the last missed. One step at a time, errors fade, letting the system see links that single models miss. As materials in SCM-based mixes behave oddly, gradual fixes start to shine. Each cycle brings better predictions; smoothing imperfections more reliably than rigid or clustered approaches ever do.



(a)

(b)

(c) (d)



(e) (f)

FIG. 1 PERFORMANCE METRICS-SCATTER PLOTS

## 5. CONCLUSION

- The choice of learning technique significantly influences the accuracy of compressive strength prediction for SCM-blended concrete.
- **XGBoost** consistently achieved the highest performance across all evaluation metrics, demonstrating superior capability in handling complex material interactions and time-dependent strength development.
- The model showed strong generalization ability, maintaining low error rates even for unconventional or less-represented concrete mixes.
- XGBoost performance remained stable despite variations in ingredient proportions, indicating robustness and adaptability.
- **Random Forest** performed better than linear models due to its ensemble tree structure but was less effective than boosting methods.Random Forest exhibited mild overfitting tendencies, which limited its predictive reliability compared to boosting algorithms.
- Boosting algorithms improve prediction accuracy iteratively by correcting previous errors in a sequential manner.
- Overall, boosting-based models provide more reliable and accurate strength predictions for sustainable SCM concrete than linear or instance-based approaches.
- The findings confirm that advanced boosting techniques are better suited for real-world durability and performance assessment of green concrete systems.

## REFERENCES

[1] H. Fu, Y. Zhang, X. Li, and J. Wang, “Prediction of compressive strength of concrete using gradient boosting trees, random forest, and neural networks,” *Materials*, vol. 18, no. 21, p. 4567, 2025. https://doi.org/10.3390/ma18214567

[2] H. Chen, X. Li, Y. Wu, and Y. Zhou, “Compressive strength prediction of high-performance concrete using machine learning and deep learning algorithms,” *Constr. Build. Mater.*, vol. 371, p. 131982, 2025. https://doi.org/10.1016/j.conbuildmat.2025.131982

[3] Z. Wan, “Influence of dimensionality reduction on machine-learning-based prediction of compressive strength of concrete,” *Constr. Build. Mater.*, vol. 368, p. 131245, 2025. https://doi.org/10.1016/j.conbuildmat.2025.131245

[4] A. Alzlfawi, “Precision assessment of supervised machine learning models and parametric optimization for predicting compressive strength of lightweight cellular concrete,” *Compos. Struct.*, vol. 330, p. 117645, 2025. https://doi.org/10.1016/j.compstruct.2025.117645

[5] S. Paudel, A. Pudasaini, R. K. Shrestha, and E. Kharel, “Compressive strength of concrete material using machine learning techniques,” *Cleaner Eng. Technol.*, vol. 15, p. 100661, 2023. https://doi.org/10.1016/j.clet.2023.100661

[6] H. U. Ahmed, F. Farooq, and K. A. Ostrowski, “Innovative soft computing techniques to predict the compressive strength of environmentally friendly concrete,” *Polymers*, vol. 15, no. 3, p. 612, 2023. https://doi.org/10.3390/polym15030612

[7] V. Rathakrishnan, A. Arulrajah, and S. Horpibulsuk, “Predicting compressive strength of concrete with fly ash and slag using boosting algorithms,” *Sci. Rep.*, vol. 12, p. 14589, 2022. https://doi.org/10.1038/s41598-022-18761-9

[8] Z. Li, Y. Wang, J. Wang, and X. Zhao, “Machine learning in concrete science: Applications, challenges, and opportunities,” *npj Comput. Mater.*, vol. 8, p. 48, 2022. https://doi.org/10.1038/s41524-022-00731-3

[9] Y. Song, J. Liu, and S. Wang, “Prediction of compressive strength of fly-ash-based concrete using ensemble and non-ensemble machine-learning approaches,” *Appl. Sci.*, vol. 12, no. 1, p. 98, 2022. https://doi.org/10.3390/app12010098

[10] A. Beskopylny, A. Lyapin, H. Anysz, and A. Meskhi, “Concrete strength prediction using machine learning methods CatBoost, KNN, and SVR,” *Appl. Sci.*, vol. 12, no. 21, p. 10912, 2022. https://doi.org/10.3390/app122110912

[11] W. Ahmad, F. Farooq, and K. A. Ostrowski, “Application of advanced machine learning approaches to forecast compressive strength of concrete containing fly ash and blast furnace slag,” *J. Mater. Civ. Eng.*, vol. 33, no. 9, p. 04021233, 2021. https://doi.org/10.1061/(ASCE)MT.1943-5533.0003862

[12] A. Gholampour and T. Ozbakkaloglu, “Machine learning-based prediction of compressive strength of geopolymer and SCM concrete,” *Constr. Build. Mater.*, vol. 240, p. 117920, 2020. https://doi.org/10.1016/j.conbuildmat.2019.117920

[13] I.-C. Yeh, “Modeling of strength of high-performance concrete using artificial neural networks,” *Cem. Concr. Res.*, vol. 28, no. 12, pp. 1797–1808, 1998. https://doi.org/10.1016/S0008-8846(98)00165-3

[14] J.-S. Chou, C.-F. Tsai, A.-D. Pham, and Y.-H. Lu, “Machine learning techniques for predicting compressive strength of concrete,” *Autom. Constr.*, vol. 20, no. 4, pp. 449–460, 2011. https://doi.org/10.1016/j.autcon.2010.11.013

[15] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 785–794, 2016. https://doi.org/10.1145/2939672.2939785