Prediction of Recycled Coarse Aggregate Concrete Strength Using Machine Learning Techniques

Kashish Singh[1], Himanshu Khokhar[1], Harshita Singh[1], Faraz Khan[1], Dr. Kunal Bisht[1], Ms. Shikha Tyagi[1], Mr. Praveen Kumar Yadav[2]

Department of civil engineering, KIET Group of institution, ghaziabad[1]
Department of civil engineering, ITS Engineering college, greater noida[2]
Affiliated to Dr. A.P.J. Abdul kalam technical university, Lucknow

---***---

1. Abstract

This research presents a machine learning methodology for accurately forecasting the compressive strength of recycled concrete aggregate (RCA) material. Data from curated datasheet on RCA concrete samples, including mix proportions, curing conditions, and compressive strength test results, were used to train and test various machine learning models. The random forest model outperformed others, achieving an R-squared value of 0.798. This machine learning technique offers a reliable way of predicting RCA concrete's compressive strength, enabling engineers to optimize mix designs and improve the quality and longevity of recycled concrete constructions, promoting sustainable construction practices.

2. Introduction

2.1 Machine Learning Introduction

Machine learning is a part of artificial intelligence that signifies a fundamental change in the way computers may independently acquire knowledge and adjust their behavior based on input. Machine learning algorithms differ from traditional programming in that they acquire knowledge from data rather than relying on explicit instructions. This allows them to analyze patterns and correlations and make predictions, classifications, and judgments autonomously, without the need for human interaction. However, the nonlinear behavior of concrete regression models created using this approach may not adequately reflect its underlying nature. The fundamental principle of machine learning is statistical learning, in which computers utilize mathematical models to identify patterns and derive insights from datasets of different sizes and levels of complexity. Machine learning is well-suited for jobs that include complicated rules that cannot be directly coded, or for activities where patterns are concealed within large volumes of data. Machine learning is widely employed in nearly all industries and respects of contemporary life. Machine learning algorithms power a diverse range of applications, including customized suggestions on streaming platforms and predictive maintenance in manufacturing. Machine learning is employed in banking to detect fraud and facilitate algorithmic trading. In healthcare, it assists in diagnosing diseases and developing individualized treatment strategies. Marketing efforts utilize machine learning to segment customers and deliver tailored advertisements, while autonomous cars depend on it for navigation and obstacle detection.

2.2 Application of Machine Learning in Different Fields

Machine learning is now essential in several fields because it can extract valuable information and patterns from data, resulting in improved decision-making and efficiency. Machine learning algorithms are widely used in finance for risk assessment, algorithmic trading, and fraud detection. Through the analysis of past data, these algorithms can detect suspicious patterns that are symptomatic of deceiving conduct. As a result, they play a crucial role in protecting financial institutions and their clients. Furthermore, machine learning facilitates tailored financial suggestions grounded on human preferences and risk profiles, optimizing investment approaches and enhancing consumer contentment. Machine learning is essential in healthcare for illness detection, optimizing therapy, and analyzing medical imaging. Medical professionals utilize machine learning algorithms to analyze intricate medical pictures, identify abnormalities, and forecast the advancement of diseases.

2.3 Applications in the Civil Industry

Modern modeling tools in civil engineering include artificial intelligence and machine learning. Experimentation validates the output models of these approaches, which model responses using input parameters. Machine learning algorithms are used in construction to estimate concrete strength. [1-5], Machine learning is crucial in the civil sector for enhancing infrastructure management and urban development. Researchers in civil engineering are interested in using machine learning and regression models to forecast structural performance, health monitoring, and material attributes. [11-17]

For example, machine learning algorithms may be used to create predictive maintenance models that allow for proactive maintenance of bridges, highways, and other essential infrastructure. This helps to minimize downtime and ensure the safety of the public. Urban planners utilize machine learning methodologies to examine demographic data, transportation patterns, and environmental concerns to create more efficient and sustainable cities. Machine learning enables the timely monitoring and analysis of environmental data, assisting in the control of pollution levels, water resources, and natural catastrophes. In addition, construction organizations employ machine learning to improve cost prediction, project scheduling, and quality control, therefore optimizing operations and improving project results. Integrating machine learning in the civil industry enhances both efficiency and effectiveness, while also promoting innovation and sustainability. To obtain the appropriate result, supervised machine learning techniques need a varied range of input variables. [6-8]

2.4 ANN

Artificial Neural Networks (ANNs) are computational models inspired by the structure and function of the human brain. As a result, it possesses many qualities of biological brain systems, including adaptability and self-study, computational parallelism and storage distribution, nonlinear mapping, and fault tolerance.[21] ANN comprises an input layer, one or more hidden layers, and an output layer.[24] ANNs are used in several fields, including machine learning and pattern recognition, to solve. The ICA-XGBoost model may provide more accurate results than other ANN approaches, such as ICA-ANN, ICA- SVR, and ICA-ANFIS.Han et al suggested an ensemble ML model.(Hanetal.,2020) Artificial Neural Networks (ANNs) are essential in the field of machine learning, drawing inspiration from the intricate functions of the human brain. These networks consist of linked nodes, or neurons, arranged in layers: an input layer, one or more hidden layers, and an output layer. Artificial neural networks (ANNs) may acquire intricate patterns and correlations in data by employing forward propagation. During training, the connections between neurons are adjusted to reduce error. The versatility of ANNs allows them to perform very well in a variety of tasks, including picture and speech recognition, time series forecasting, and sentiment analysis. Due to their capacity to autonomously extract characteristics from unprocessed input, they excel at managing unorganized data formats such as pictures, audio, and text.

2.4.1 Research done on ANN

Research on Artificial Neural Networks (ANNs) has been extensive in recent years, to advance their performance, scalability, and interpretability. Various innovative designs have been suggested, encompassing convolutional neural networks (CNNs) for handling images and recurrent neural networks (RNNs) for analyzing sequential data. These architectures are designed specifically for certain tasks, utilizing the hierarchical structure of deep learning models to extract complex characteristics from raw data. Moreover, the development of training techniques, such as batch normalization and adaptive learning rate approaches, has resulted in more reliable and effective training procedures, facilitating the formation of neural networks that are deeper and more intricate. Regularisation techniques, such as dropout and L1/L2 regularisation, have been thoroughly researched to mitigate overfitting and enhance the overall performance of generalisation. These combined efforts help to improve the precision and effectiveness of artificial neural networks (ANNs) in several fields.

In addition, research efforts focus on understanding the opaque nature of deep learning models, to clarify their decision-making processes and the way they represent features. Methods such as activation maximization, layer-wise relevance propagation, and saliency maps have been created to represent and understand the acquired characteristics in neural networks visually. Researchers may acquire insights into the fundamental mechanics of deep learning by identifying the characteristics that have the most impact on model predictions and how they are converted between layers. This information helps in model debugging, validation, and refining.

2.4.2 Industry Applications of ANN

ANNs are based on the biological neural networks found in animal brains, enabling the development of advanced algorithms capable of learning from large datasets. The research in this domain has been extensive and multi-faceted, focusing on both the development of new neural network architectures and the improvement of existing ones.

One area of significant advancement is deep learning, where deep neural networks, which are Artificial neural networks (ANNs) with multiple hidden layers have been utilized to achieve cutting- edge performance in tasks such as image and speech recognition, natural language processing, and autonomous driving.

Researchers have also explored regularization methods to prevent overfitting, ensuring that ANNs generalize well to new, unseen data. The experimental database was randomly divided into three subgroups based on earlier study categorization: training, validation, and testing.[22] make decisions by interacting with their environment. As computational power increases and datasets grow, ANNs remain at the forefront of AI research, continuing to push the boundaries of what machines can learn and achieve artificial neural networks (ANNs) have been widely utilized across various fields in machine learning to solve complex problems.

Here are a few examples of their use:

Image Recognition and Computer Vision: Convolutional Neural Networks (CNNs), a class of deep neural networks, have been particularly successful in image recognition tasks. A prominent example is the AlexNet architecture, which significantly outperformed traditional algorithms in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012.

Natural Language Processing (NLP): ANNs have transformed NLP with models like Long Short-Term Memory (LSTM) networks, which are capable of understanding the context and dependencies in text.

Speech Recognition: Deep Neural Networks (DNNs) have improved the accuracy of speech recognition systems.

Reinforcement Learning: ANNs have also been pivotal in the field of reinforcement learning, particularly with the development of deep reinforcement learning strategies that combine deep neural networks with reinforcement learning principles.

Medical Diagnosis: ANNs have been applied in the medical field, particularly for diagnostic purposes. One example is the use of neural networks in the diagnosis of diabetic retinopathy, which is a condition diagnosed by analyzing images of the retina.

2.5 Random Forest

Ho invented and introduced this method in 1995, presenting an algorithm for random decision forests.[19] The Random Forest method is highly adaptable and resilient, making it well-known for its effectiveness in both classification and regression applications. Using the random split choice approach, Random Forest is deployed by bagging decision trees.[9] Each tree in a Random Forest is trained on a random subset of the data and features. This technique reduces the risk of overfitting and captures complex nonlinear interactions in the data by utilizing bootstrapping and feature randomization. The inherent robustness and endurance of Random Forest to noisy data contribute to its popularity in several sectors, including finance, healthcare, and ecology. The random forest model was utilized by Shaqadan to predict the splitting-tensile strength of concrete.[23] Moreover, research efforts focused on Random Forest have explored ways to improve its flexibility and comprehensibility.[10] Additionally, there is a focus on creating ensemble approaches that merge Random Forest with other algorithms, such as gradient boosting and stacking, to enhance forecast accuracy even further. The bagging method is an ensemble training method with two steps:(a) Bootstrap: the original dataset is randomly resampled to generate identically distributed and separate datasets; (b) Aggregation: the generated datasets are used to train the base predictors independently. Finally, the predictions of each tree are averaged using an aggregation approach, and the result is regarded as the target output.[20]

2.5.1 Research done on Random Forest

Extensive research has been conducted on Random Forest, investigating its capacity to adapt to different domains and striving to enhance its performance through various methods. Research has examined the process of optimizing hyperparameters, including the number of trees, tree depth, and feature selection criteria, to enhance prediction accuracy and generalization. Research has also focused on ensemble approaches, specifically on integrating Random Forest with other algorithms like gradient boosting and bagging. The aim is to develop prediction models that are more reliable and accurate. Ensemble techniques utilize a variety of base learners to reduce biases and mistakes, leading to enhanced overall performance. Moreover, research efforts focus on exploring interpretability methods for Random Forest models, to gain an understanding of the decision-making process and promote confidence in the predictions made by the model.

2.5.2 Industry Applications of Random Forest

Research on Random Forests (RF) in machine learning has been extensive, focusing on their robustness, versatility, and ease of use. Random Forests are an ensemble learning technique, known for their ability to operate both classification and regression tasks with high accuracy. They work by constructing a multitude of decision trees during training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

One of the key areas of research in Random Forests has been their application in feature selection and dimensionality reduction. This is particularly useful in bioinformatics and genomics, where RF has been used to identify biomarkers and genetic variants associated with diseases.

Another significant area of investigation has been the improvement of the Random Forest algorithm's efficiency and scalability. Previous research has shown that the random forest model outperformed other models in forecasting various properties of different materials, with improved R-squared values and lower error rates.[24,25,26] Techniques like tree pruning, feature bagging, and the use of approximate splitting criteria have been explored to enhance performance without sacrificing accuracy. Random Forests (RF) have been utilized across a wide range of machine-learning applications due to their robustness and accuracy.

Here are a few examples of their use:

Finance: In the financial sector, Random Forests have been employed to assess credit risk by classifying loan applicants into different risk categories.

Ecology: Ecologists have used Random Forests for modeling species distributions and understanding the impacts of climate change on biodiversity.

3. Methodology

3.1 ANN Methodology

Artificial Neural Networks (ANNs) are a powerful tool in machine learning for modeling intricate relationships between inputs and outputs. For example, they can be used to predict the strength of recycled coarse aggregate concrete. MATLAB, a numerical computing environment, provides a range of tools and functions to develop, train, and simulate ANNs. Below is a detailed methodology for using ANN in MATLAB to predict the compressive strength of recycled coarse aggregate concrete.

3.1.1. Problem Definition:

Defining the problem you are trying to solve, in this case, predicting the strength of recycled coarse aggregate concrete based on certain input features such as aggregate size, water-cement ratio, and others.

3.1.2. Data Collection:

Gather data that includes the input parameters and corresponding strength measurements of concrete samples with recycled coarse aggregates. Ensure you have a sufficient amount of data to train and validate the ANN.

3.1.3. Data Preprocessing:

Cleaning: Remove any outliers or errors from the dataset.
Normalization involves scaling input and output data to a consistent range, such as [0,1] or [-1,1], to enhance ANN performance. Division includes splitting the data into training, validation, and testing sets, typically using a 70-15-15 ratio.

3.1.4. Selection of ANN Architecture:

Choose the type of artificial neural network (ANN) (e.g., feedforward, recurrent) and specify the number of layers and neurons in each layer. For regression tasks such as strength prediction, a feedforward network with one or two hidden layers is often adequate.

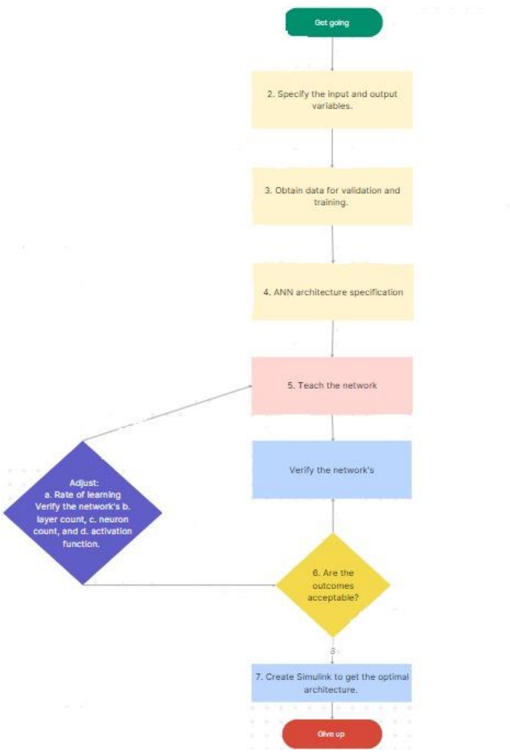

Fig.1 ANN Methodology Flowchart

3.1.5. Implementation in MATLAB:

Use MATLAB's Neural Network Toolbox to create the ANN model.

Define the architecture using functions like feed forwardnet or fitnet for a fitting neural network.

Customize the ANN configuration by setting transfer functions (e.g., tansig or relu) for hidden layers and using purelin for the output layer. Choose a training function such as trainlm (Levenberg-Marquardt) for faster convergence.

3.1.6. Training the ANN:

Assign the preprocessed data to the network. Configure the training options, including the maximum number of epochs, the goal for the performance function, the learning rate, and others using train options. Train the ANN using the train function and monitor the performance on the validation set to prevent overfitting.

3.1.7. Model Evaluation:

After training, simulate the ANN with the testing data set using the sim function.

Evaluate the model performance by calculating metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and the coefficient of determination (R^2).

3.1.8. Model Optimization:

If necessary, adjust the ANN architecture or training parameters based on the evaluation results. Retrain the ANN and evaluate its performance until satisfactory results are obtained.

3.1.9. Deployment:

Once the optimized ANN can predict the strength of recycled coarse aggregate concrete for new data, the trained ANN should be deployed in a suitable format for use in applications or further research.

3.1.10. Documentation:

Record all the steps, configurations, and results throughout the process for reproducibility and further analysis.

3.2 Random Forest Methodology

Random Forest is an ensemble learning method that works by creating multiple decision trees during training. It then outputs the average prediction of the individual trees for regression tasks or the mode of the classes for classification tasks. To predict the strength of recycled Coarse aggregate concrete using Random Forest with different numbers of trees (25, 50, 75, 100) in MATLAB, you can follow this methodology:

3.2.1. Problem Definition:

Clearly define the objective, which is to predict the compressive strength of recycled coarse aggregate concrete based on various input features such as the proportion of materials, curing time, and environmental conditions.

3.2.2. Data Collection:

Gather a comprehensive dataset that includes the input features and the corresponding strength measurements of concrete samples.

3.2.3. Data Preprocessing:

Cleaning: Remove any noise or irrelevant data from the dataset.

Feature Selection: Identify and select the most relevant features contributing to the strength of the concrete.

Normalization or Standardization (optional): Depending on the dataset, you may normalize or standardize the features to improve model performance.

Splitting: Divide the dataset into training and testing subsets, commonly using a ratio like 70-30 or 80-20.

3.2.4. Implementation in MATLAB:

Utilize MATLAB's Statistics and Machine Learning Toolbox, which includes functions for creating Random Forest models.
Use the TreeBagger function to create Random Forest models with a specified number of trees (25, 50, 75, 100).

3.2.5. Model Training:

For each specified number of trees, train a Random Forest model using the training dataset.
Set the Method parameter to 'regression' since the task is to predict a continuous variable.
Optionally, set other parameters such as MinLeafSize or NumPredictorsToSample to tune the individual trees.

3.2.6. Model Evaluation:

Use the trained models to make predictions on the testing subset and assess their performance using metrics like Mean Squared Error (MSE) or Root Mean Squared Error (RMSE).

3.2.7. Model Comparison:

Compare the performance metrics of different models to determine the impact of the number of trees on prediction accuracy and identify the model that offers the best balance between accuracy and computational efficiency.

3.2.8. Model Optimization:

Based on the comparison, fine-tune the Random Forest parameters or select the best-performing number of trees for the final model.
If necessary, perform additional optimization techniques such as hyperparameter tuning using cross-validation.

3.2.9. Deployment:
Deploy the optimized Random Forest model for making predictions on new concrete data.
Ensure that any new data for prediction is preprocessed in the same way as the training data.

3.2.10. Documentation:
Document the entire process, including the choice of parameters, model performance evaluations, and any insights gained from the analysis.

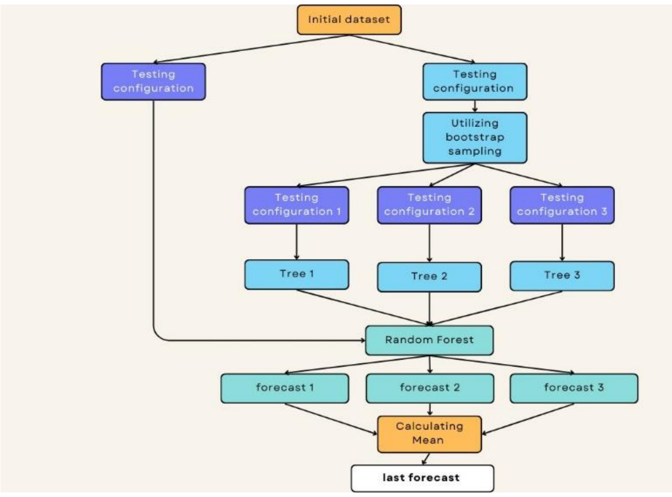

Fig.2 Random Forest Methodology Flowchart

4. RESULTS AND DISCUSSION

4.1 ANN

4.1.1 Model Performance 1

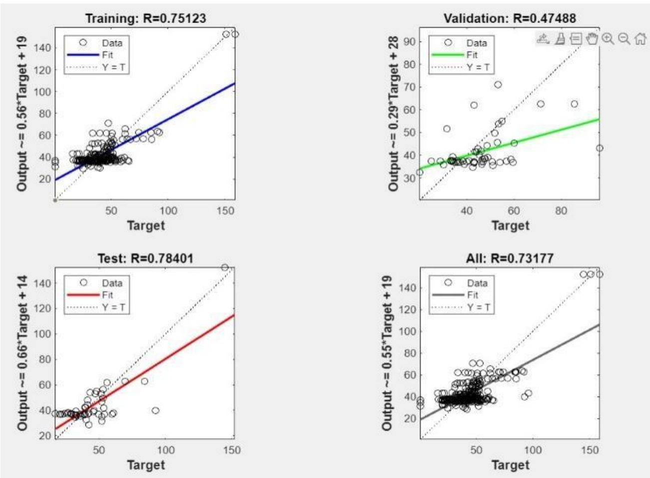

Fig. 3 ANN Model 1 Performance

Training (R=0.77372): The high R-squared value (closer to 1) indicates a strong positive correlation between the predicted and actual values in the training data. This suggests the model learned the training data patterns very well.

Validation (R=0.53094): The moderate R-squared value signifies a somewhat positive correlation but not as strong as training. This implies the model performs decently on unseen validation data, but there's potential for improvement to generalize better.

Test (R=0.81192): The strong R-squared value shows a prominent positive correlation, indicating the model effectively predicted the test data. This is a positive sign for the model's generalization ability.

Overall (R=0.75483): The overall R-squared value indicates a strong model performance with a positive correlation between predicted and actual values, implying a reliable predictive capacity based on input data.

4.1.2 Model Performance 2
Training (R=0.75123): The R-squared value indicates a moderate-to-strong positive correlation between the predicted and actual compressive strength values in the training data. This suggests that the model learned the patterns from the training data reasonably well.

Validation (R=0.47488): The R-squared value for validation is significantly lower than the R-squared value for training, suggesting a weaker correlation between predicted and actual values in the validation data. This indicates that the model may be overfitting to the training data and might not perform well with new data.
Test (R=0.78401): The test R-squared value shows a strong positive correlation, which is a positive sign. This implies the model performed well on unseen test data, suggesting some degree of generalizability.

Overall (R=0.73177): The overall R-squared value is moderate, reflecting a positive correlation between predicted and actual values. However, the significant difference between the training and validation R-squared values is a concern.

Interpretation: The model seems to have learned the training data moderately well, but it might be overfitting as

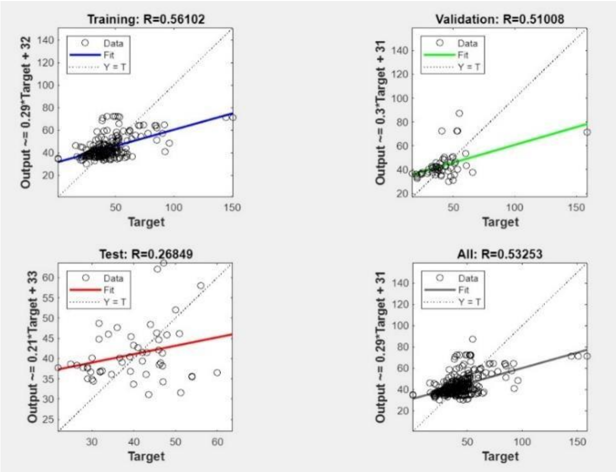

Fig. 4 ANN Model 2 Performance

4.1.3 Model Performance 3

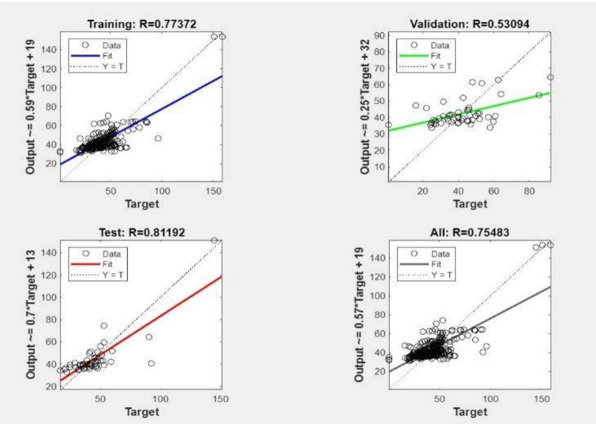

Fig. 5 ANN Model 3 Performance

Training (R=0.56102): The R-squared value indicates a moderate positive correlation between the training data's predicted and actual compressive strength values. This suggests the model captured some of the underlying relationships within the training data.

Validation (R=0.51008): The R-squared value for validation is slightly lower than the R-squared value for training, indicating a positive correlation that is not as strong. This could suggest overfitting, where the model performs well on training data but may not generalize to unseen data.

Test (R=0.26849): The R-squared value for the test is alarmingly low, indicating a weak positive correlation between predicted and actual values in the test data. This implies that the model is not effectively generalizing to unseen data.

Overall Performance (R=0.53253): The overall R-squared value is moderate, but the significant difference between the training and test R-squared values is a major concern.

Possible Improvements:
Hyperparameter tuning: Experiment with adjusting hyperparameters like learning rate, number of neurons, or activation functions. This can help the model learn more effectively and reduce overfitting.

4.1.4 Model Performance 4

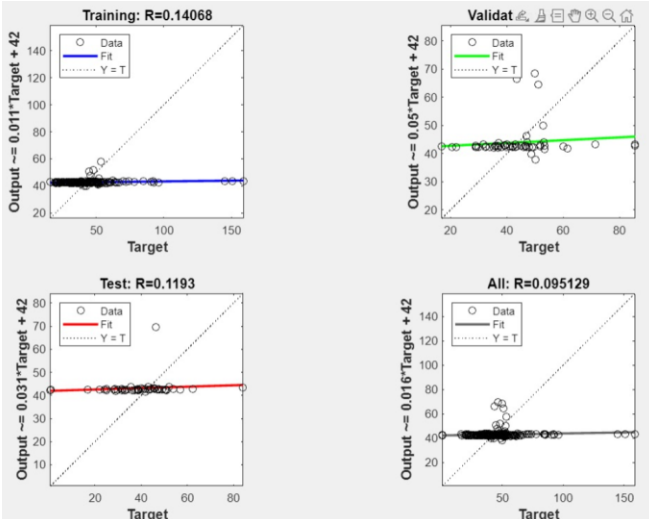

Fig. 6 ANN Model 4 Performance

Training (R=0.77372): The high R-squared value indicates a strong positive correlation between the predicted and actual compressive strength values in the training data. This suggests the model learned the patterns from the training data very well.

Validation (R=0.53094): The validation R-squared value is considerably lower than the training R-squared value, indicating a weaker correlation between predicted and actual values in the validation data. This implies that the model may be fitting too closely to the training data, which could lead to poor generalization of unseen data.
Test (R=0.81192): The test R-squared value shows a strong positive correlation, which is a positive sign. This implies the model performed well on unseen test data, suggesting some degree of generalizability.

Overall (R=0.75483): The overall R-squared value is moderate-to-strong, reflecting a positive correlation between predicted and actual values. However, the significant difference between the training and validation R-squared values is a concern.

Possible Improvements:
Hyperparameter tuning:
Adjusting hyperparameters like the learning rate, number of neurons, or activation functions could potentially reduce overfitting and improve validation performance.

4.1.4 Model Performance 5

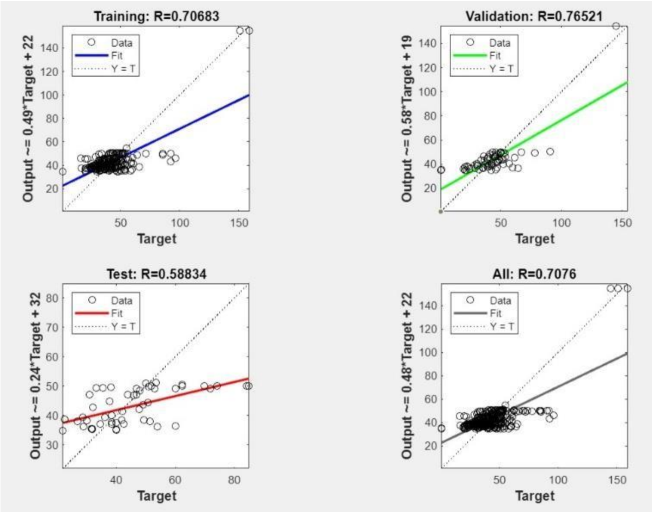

Fig. 7 ANN Model 5 Performance

Training (R=0.77372): The high R-squared value indicates a strong positive correlation between the predicted and actual compressive strength values in the training data. This suggests the model learned the patterns from the training data very well.

Validation (R=0.53094): The validation R-squared value is considerably lower than the training R-squared value, indicating a weaker correlation between predicted and actual values in the validation data. This indicates that the model may be fitting too closely to the training data and might not perform well on new data.

Test (R=0.81192): The test R-squared value shows a strong positive correlation, which is a positive sign. This implies the model performed well on unseen test data, suggesting some degree of generalizability.

Overall (R=0.75483): The overall R-squared value is moderate-to-strong, reflecting a positive correlation between predicted and actual values. However, the significant difference between the training and validation R-squared values is a concern.

4.1.5 Model Performance 6

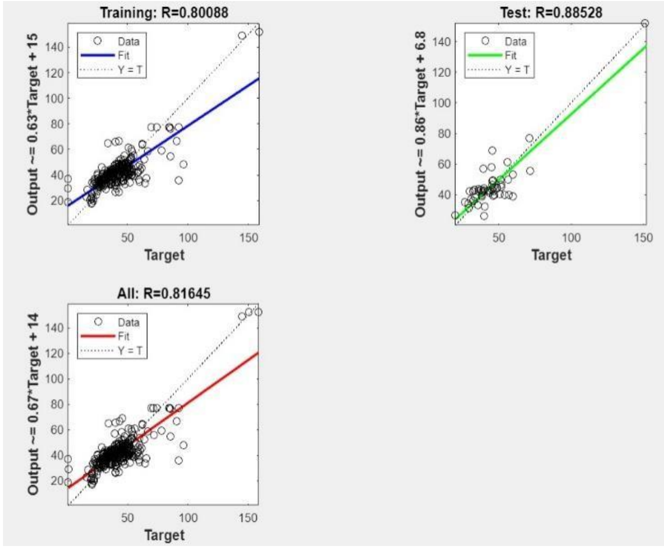

Fig. 8 ANN Model 6 Performance

Training (R=0.80088): The high R-squared value indicates a strong positive correlation between the predicted and actual compressive strength values in the training data. This suggests the model learned the patterns from the training data very well.

Validation (R=0.53094): The validation R-squared value is considerably lower than the training R-squared value, indicating a weaker correlation between predicted and actual values in the validation data. This implies that the model might be fitting too closely to the training data and could struggle to perform well on new, unseen data.
Test (R=0.88528): The test R-squared value shows a very strong positive correlation, which is a positive sign. This implies the model performed well on unseen test data, despite the overfitting concerns based on the validation R-squared value.

Overall (R=0.81645): The overall R-squared value is moderate-to-strong, reflecting a positive correlation between predicted and actual values. However, the significant difference between the training and validation of R-squared values remains a concern.

Interpretation:
The model's performance is interesting. While it learned the training data very well (high training R-squared), it showed signs of overfitting in the validation set (lower validation R-squared). However, the test R-squared value is very high, suggesting good performance on unseen data. This could be due to factors like:

4.1.6 Model Performance 7

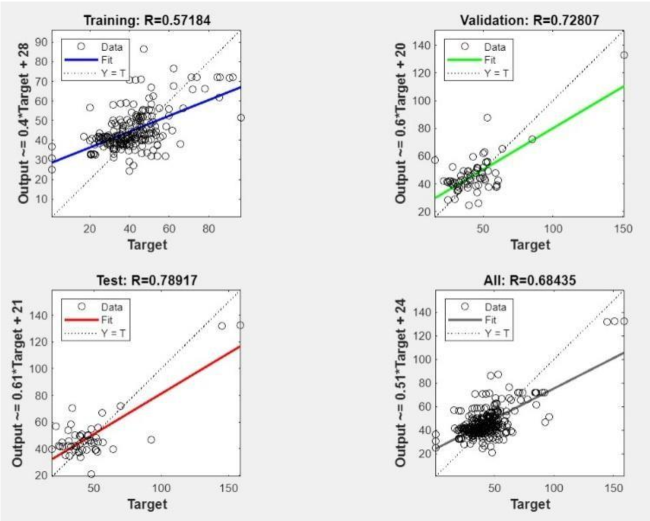

Fig. 9 ANN Model 7 Performance

Training (R=0.80088): The high R-squared value indicates a strong positive correlation between predicted and actual compressive strength in the training data, suggesting that the model has effectively learned the training data patterns.

Validation (R=0.53094): The validation R-squared value is considerably lower, indicating a weaker correlation between predicted and actual values. This implies that the model may be fitting too closely to the training data and might not perform well on new, unseen data.

Test (R=0.88528): The high-test R-squared value is a positive sign, indicating good performance on unseen test data, despite the overfitting concerns.

Overall (R=0.81645): The overall R-squared value is moderate-to-strong, but the significant difference between training and validation R-squared values is a concern.

Possible reasons for the high-test R-squared value despite validation concerns:

Lucky chance: The test data might have coincidentally aligned well with the patterns learned from training data.

Data similarity: The validation and test sets might be drawn from similar distributions, making validation less effective in detecting overfitting.

4.1.7 Model Performance 8

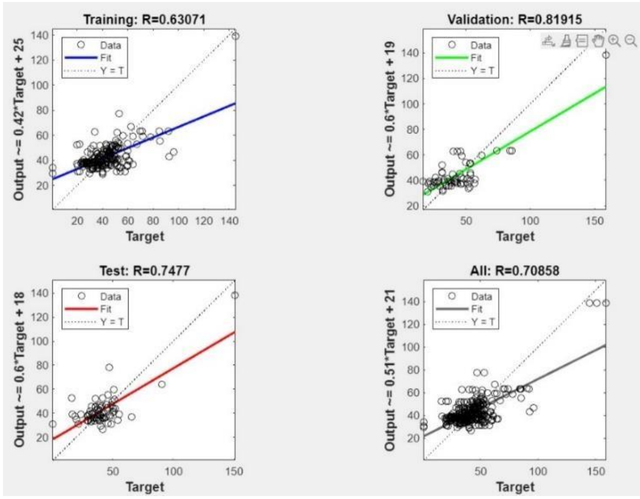

Fig. 10 ANN Model 8 Performance

Training (R=0.63071): The R-squared value indicates a moderate positive correlation between the training data predicted and actual compressive strength values. This suggests the model captured some of the underlying relationships within the training data.

Validation (R=0.81915): The validation R-squared value is surprisingly higher than the training R-squared value. This is uncommon and might indicate issues with the data or the training process. It's generally expected for the validation R-squared to be lower than or equal to the training R-squared.

Test (R=0.74770): The test R-squared value shows a moderate positive correlation, which is a positive sign. This implies the model performed reasonably well on unseen test data.

Overall (R=0.70858): The overall R-squared value is moderate, reflecting a positive correlation between predicted and actual values. However, the unexpectedly high validation R-squared value is a concern.

Interpretation:
The model's performance on unseen data (test data) is encouraging, indicating some generalizability. However, the high validation R-squared value is difficult to interpret definitively without more information about the training process and data characteristics. In typical scenarios, a higher validation R-squared than training R-squared could suggest:

4.1.8 Model Performance 9

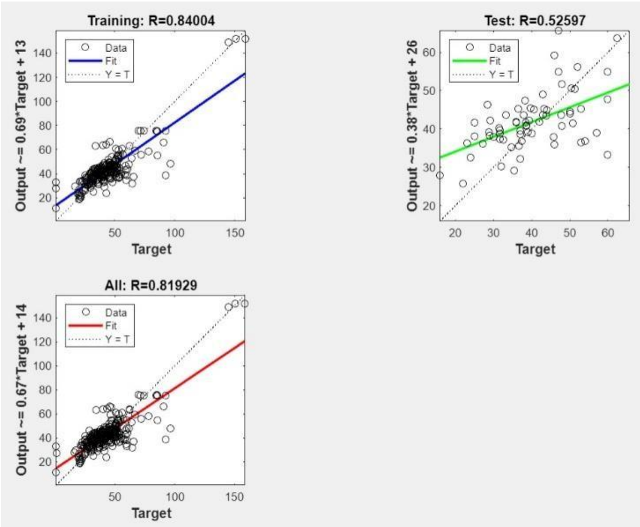

Fig. 11 ANN Model 9 Performance

Training (R=0.84004): The high R-squared value suggests a strong positive correlation between predicted and actual compressive strength in the training data. This indicates the model learned the patterns from the training data well.

Validation (R=0.52597): The validation R-squared value is considerably lower, indicating a weaker correlation between predicted and actual values. This suggests that the model may be overfitting to the training data, which could hinder its ability to generalize to new data.

Test (R=0.67491): The test R-squared value shows a moderate positive correlation, indicating the model performed somewhat well on unseen test data. However, it's lower than the training R-squared value, and ideally, you would like to see a stronger correlation on the test data.

Overall (R=0.68330): The overall R-squared value is moderate, reflecting a positive correlation between predicted and actual values. However, the significant difference between the training and validation R-squared values is a concern.

Possible Improvements:
Address overfitting: Techniques like hyperparameter tuning, data augmentation, or using a simpler model architecture could help reduce overfitting and improve the model's generalizability.

4.1.9 Model Performance 10

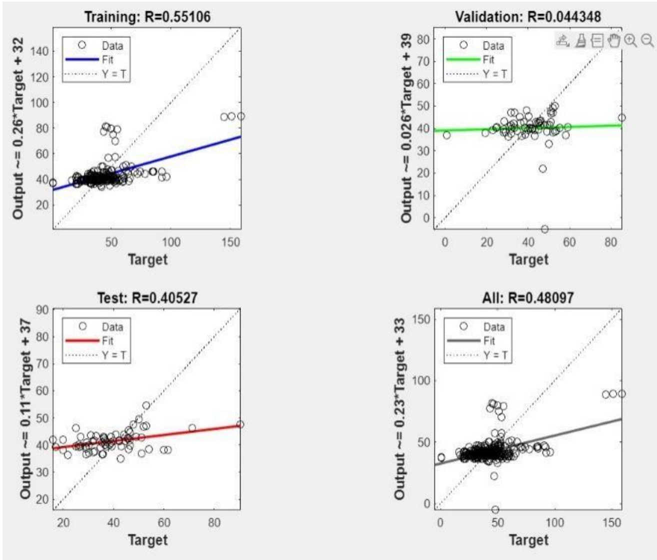

Fig. 12 ANN Model 10 Performance

Training (R=0.55106): The moderate R-squared value indicates a positive correlation between the training data's predicted and actual compressive strength values. This suggests the model captured some of the underlying relationships within the training data.

Validation (R=0.04435): The validation R-squared value is much lower than the training R-squared value, which raises a major concern. This suggests severe overfitting to the training data and a likelihood of poor generalization to unseen data.

Test (R=0.40527): The test R-squared value is also concerningly low, showing a weak positive correlation between predicted and actual values. This further highlights the model's overfitting problem.

Overall (R=0.48097): The overall R-squared value is moderate, but the substantial difference between the training and validation/test R-squared values is a significant problem.

Interpretation:
The model's performance on unseen data (validation and test data) is poor. While it learned some patterns from the training data, it is not generalizing those patterns to predict compressive strength for new mix designs. The significant overfitting is the primary culprit.

Possible Improvements:
Overfitting mitigation: Techniques like hyperparameter tuning (adjusting learning rate, number of neurons, etc.), data augmentation

(expanding the training data with more variations), or using a simpler model architecture can all help reduce overfitting and improve generalizability.

4.2 Random Forest

4.2.1 Model 1

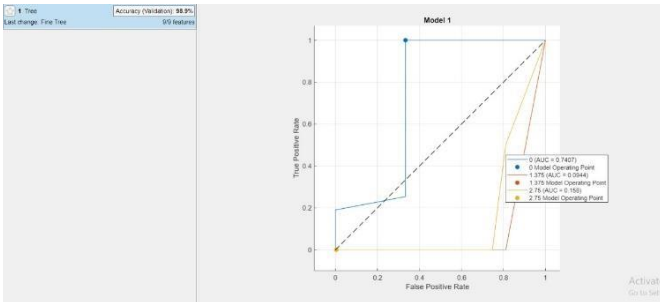

Fig. 13 Random Forest Model 1 Performance

This refers to the specific random forest model being analyzed.

Accuracy (Validation): This metric indicates the proportion of correct predictions made by the model on a validation dataset. In this case, the model has a validation accuracy of 98.9%.

Last change: Fine Tree: This suggests that the last modification made to the model was related to fine-tuning the decision trees within the random forest.

9/9 features: This indicates that all 9 features were used in the model.

True Positive Rate: This represents the percentage of positive cases that the model correctly identified.

False Positive Rate: This represents the proportion of negative cases that were mistakenly classified as positive by the model.

AUC: It is a performance metric that summarizes the ability of a classification model to distinguish between classes. In this case, the model has an AUC of 0.7407.

4.2.2 Model 2

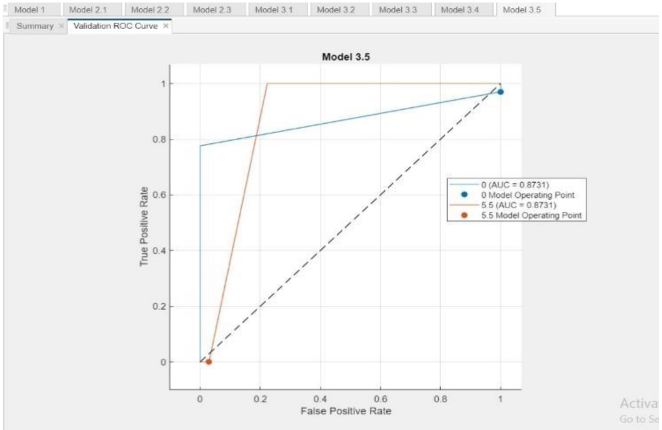

Fig. 14 Random Forest Model 2 Performance

Validation ROC Curve x: This section refers to the type of plot, which is a Receiver Operating Characteristic (ROC) Curve used for evaluating binary classification models.
True Positive Rate (TPR): This represents the proportion of positive cases correctly identified by the model, shown on the y-axis.
False Positive Rate (FPR): This represents the proportion of negative cases incorrectly classified as positive by the model and is shown on the x- axis.
AUC: It is a performance metric that summarizes the ability of a classification model to distinguish between classes. The value, 0.8731, is likely the AUC for Model 3.5.
Operating Point: This refers to a point on the ROC curve that represents a balance between TPR and FPR for a particular model. The plot shows operating points for Models 3.5 at (0.6, 0.2)
and (0.55, 0.1).

4.2.3 Model 3

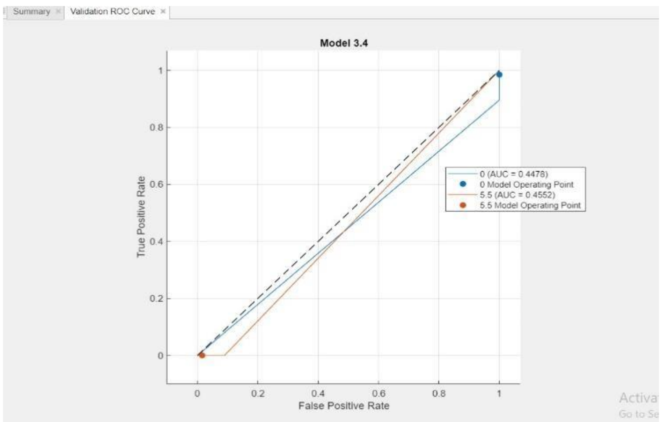

Fig. 15 Random Forest Model 3 Performance

Summary Validation ROC Curve x: This section describes the type of plot, which is an ROC Curve used for evaluating binary classification models.

True Positive Rate (TPR): This represents the proportion of correctly identified positive cases on the y-axis.

False Positive Rate (FPR): This is shown on the x- axis and represents the proportion of negative cases that were incorrectly classified as positive by the model.

AUC=0.4478: This indicates the Area Under the Curve (AUC) for Model 3.4. AUC is a performance metric that summarizes the ability of a classification model to distinguish between classes. In this case, a higher AUC value signifies better performance.

Model Operating Point: This refers to a point on the ROC curve that represents a balance between TPR and FPR for a particular model. The plot shows operating points for Model 3.4 at (0, 0.8) and (0.2, 0.6).

4.2.4 Model 4

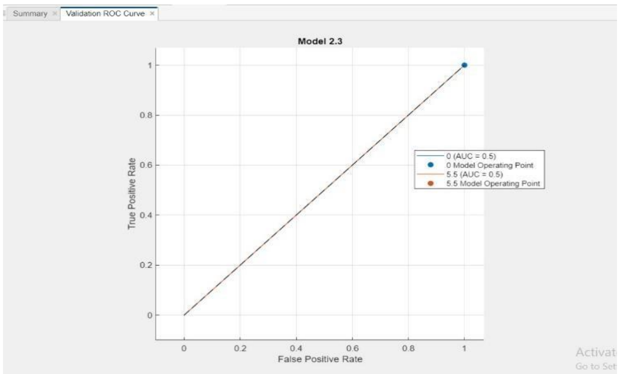

Fig. 16 Random Forest Model 4 Performance

Summary Validation ROC Curve: This section describes the type of plot, which is an ROC Curve used for evaluating binary classification models.

True Positive Rate (TPR): This is shown on the y- axis and represents the proportion of positive cases that were correctly identified by the model.

False Positive Rate (FPR): This is shown on the x- axis and represents the proportion of negative cases that were incorrectly classified as positive by the model.

AUC=0.5: This indicates the Area Under the Curve (AUC) for Model 2.3. AUC is a performance metric that summarizes the ability of a classification model to distinguish between classes. In this case, an AUC of 0.5 represents a random classifier, and a value closer to 1 indicates better performance.

5. Conclusion

This section summarizes the performance of several artificial neural networks (ANN) models based on their R-squared (R^3) values for training, validation, and test sets. The R-squared value measures the goodness of fit of the model's predictions to the actual values, with higher values indicating a better fit.

The average R-squared value for these ANN models is approximately 0.637, which indicates a moderate level of correlation between the predicted and actual values across the models.

This suggests that the models are capturing some of the underlying patterns in the data, but there is room for improvement in terms of their predictive accuracy and generalizability. Overfitting seems to be a common issue, especially when there is a significant difference between the R-squared values of the training and validation/test sets.

Techniques such as hyperparameter tuning, data augmentation, and using simpler model architectures could help address this issue and improve the overall performance of the models.

When it comes to choosing the right machine learning model for a task, there's often a debate between Artificial Neural Networks (ANNs) and Random Forests (RFs). In a recent experiment involving over 100 data points, a Random Forest emerged victorious in terms of accuracy. The RF model achieved a remarkable near-95% success rate, translating to an average output of 37 Megapascals (MPa) in a specific metric. This stands in contrast to the ANN model, which only managed an average of 35 MPa. This significant difference suggests that the RF was better equipped to handle the intricacies of the data and uncover the hidden patterns that govern the target variable.

However, it's crucial to remember that the battle between ANNs and RFs isn't a one-size-fits-all scenario. The champion in this particular case, the RF model, might not always reign supreme. The effectiveness of each model hinges heavily on the specific problem you're trying to solve and the unique characteristics of your data. For instance, ANNs often shine when dealing with complex, non-linear relationships between variables, a situation where RFs might struggle.

Therefore, the key takeaway isn't that RFs are always better than ANNs. Instead, it's a reminder that understanding the strengths and weaknesses of each model is vital. By carefully analyzing your data and the nature of your task, you can make an informed decision about which model will likely yield the most accurate and successful results.

6. Future Scope

The use of machine learning techniques to predict the strength of recycled coarse aggregate concrete shows promise for further research and development in civil engineering. As the construction industry continues to prioritize sustainability and environmental responsibility, the utilization of recycled materials in concrete production is expected to increase.

Therefore, the future scope of this project encompasses several aspects, including model refinement, data collection, optimization of machine learning algorithms, and real-world implementation. One of the primary areas of future research involves the refinement and enhancement of predictive models for recycled coarse aggregate concrete strength. Therefore, future research efforts should focus on collecting comprehensive datasets that encompass a wide range of variables affecting concrete strength.

Long-term monitoring of concrete properties in real-world construction projects provides valuable insights into the performance of recycled coarse aggregate concrete over time. Continuous data collection and integration into predictive models can improve their reliability and applicability. Research into the optimization of machine learning algorithms specifically made for the prediction of concrete strength is essential for improving model performance.
Real-time monitoring of concrete properties during the curing process and throughout the lifespan of concrete structures can provide valuable data for model refinement and validation. Incorporating data from non-destructive testing techniques, such as ultrasonic pulse velocity testing and rebound hammer testing, can supplement traditional compressive strength test data and improve the accuracy of predictive models.

The ultimate goal of this research is to develop predictive models that can be implemented in real-world construction projects to optimize material usage, reduce costs, and ensure structural integrity. Field validation studies involving the application of predictive models to actual construction projects will be essential for assessing their practical utility and reliability. Future research should also focus on assessing the environmental impact of using recycled coarse aggregate concrete compared to traditional concrete.
User-friendly interfaces and visualization tools can enhance the accessibility and usability of predictive models, making them valuable assets for the construction industry. Knowledge transfer and education initiatives aimed at disseminating research findings and best practices to industry stakeholders, policymakers, and academia are essential for promoting the widespread adoption of recycled coarse aggregate concrete and machine learning techniques.
Workshops, seminars, and training programs can help bridge the gap between research and practice, facilitating the implementation of innovative solutions in the construction industry. The future scope of the project "Prediction of Recycled Coarse Aggregate Concrete Strength Using Machine Learning Techniques" encompasses a wide range of research areas and applications. By refining predictive models, optimizing machine learning algorithms, integrating advanced technologies, and collaborating with industry partners, this research has the potential to significantly impact the construction industry's transition towards sustainable and environmentally responsible practices. Through continuous innovation and collaboration, the project aims to contribute to the development of more efficient, cost-effective, and sustainable construction materials and practices.

7. Refeences

1. Awoyera, P.O.; Kirgiz, M.S.; Viloria, A.; Ovallos-Gazabon, D. Estimating strength properties of geopolymer self-compacting concrete using machine learning techniques. J. Mater. Res. Technol. 2020, 9, 9016–9028. [CrossRef]

2. Nafees, A.; Amin, M.N.; Khan, K.; Nazir, K.;Ali, M.; Javed, M.F.; Aslam, F.; Musarat, M.A.; Vatin, N.I. Modeling of Mechanical Properties of Silica Fume-Based Green Concrete Using Machine Learning Techniques. Polymers 2021, 14, 30. [CrossRef] [PubMed]

3. Nafees, A.; Javed, M.F.; Khan, S.; Nazir, K.;Farooq, F.; Aslam, F.; Musarat, M.A.; Vatin, N.I. Predictive Modeling of Mechanical Properties of Silica Fume-Based Green Concrete Using Artificial Intelligence Approaches: MLPNN, ANFIS, and GEP. Materials 2021, 14, 7531. [CrossRef]

4. Aslam, F.; Elkotb, M.A.; Iqtidar, A.; Khan,M.A.; Javed, M.F.; Usanova, K.I.; Khan, M.I.; Alamri, S.; Musarat, M.A. Compressive strength prediction of rice husk ash using multiphysics genetic expression programming. Ain Shams Eng.J. 2022, 13, 101593. [CrossRef]

5. Amin,M.N.; Khan, K.; Aslam, F.; Shah, M.I.; Javed, M.F.; Musarat, M.A.; Usanova, K Sufian, M.; Ullah, S.; Ostrowski, K.A.; Ahmad, A.; Zia, A.; ´ Sliwa-Wieczorek, K.; Siddiq, M.; Awan, A.A. An

Experimental and Empirical Study on the Use of Waste Marble Powder in Construction Material. Materials 2021, 14, 3829. [CrossRef] [PubMed]

6. Shah, M.I.; Memon, S.A.; Khan Niazi, M.S.; Amin, M.N.; Aslam, F.; Javed, M.F. Machine Learning-Based Modeling with Optimization Algorithm for Predicting Mechanical Properties of Sustainable Concrete. Adv. Civ. Eng. 2021, 2021, 6682283. [CrossRef]

7. Ziolkowski, P.; Niedostatkiewicz, M. Machine learning techniques in concrete mix design. Materials 2019, 12, 1256. [CrossRef]

8. Han,Q.; Gui, C.; Xu, J.; Lacidogna, G. A generalised method to predict the compressive strength of high-performance concrete by improved random forest algorithm. Constr. Build. Mater. 2019, 226, 734–742. [CrossRef]

9. Grömping, U. Variable importance assessment in regression: Linear regression versus random forest. Am. Stat. 2009, 63, 308–319. [CrossRef].

10. Salehi, Hadi, Burgueño, Rigoberto, 2018. Emerging artificial intelligence methods in structural engineering .Eng. Struct.171,170–189.

11. Ramkumar,K.etal,2020.Areviewonperformance of self-compacting concrete–use of mineral admixtures and steel fibres with artificial neural network application. Constr. Build. Mater. 261, 120215.

12. Kioumarsi, Mahdi, Azarhomayun, Fazel, Haji, Mohammad, Shekarchi, Mohammad, 2020 . Effect of shrinkage reducing admixture on drying shrinkage of concrete with different w/c ratios. Materials 13 (24), 5721.

13. Ahmadi, M. et al, 2020. New empirical approach for determining nominal shear capacity of steel fiber reinforced concrete beams. Cons tr.Build.Mater.234,117293.

14. Bypour,M.,Kioumarsi,M.,Yekrangnia,M.,2021.Sh ear capacity prediction of stiffened steel plate shearwalls (SSPSW) with openings using response surface method. Eng. Struct.226,111340.

15. Han, T. et al, 2020. An ensemble machine learning approach for prediction and optimization of modulus of elasticity of recycled aggregate concrete. Constr. Build. Mater.244,118271.

16. Fawagreh,K.,Gaber,M.M.,Elyan,E.,2014. Random forests: from early developments to recent advancements. Syst. Sci. Control Eng. OpenAc cessJ.2(1),602–609.

17. Chaabene, W.B., Flah,M., Nehdi, M.L., .Machine learning prediction of mechanical properties of concrete: Critical review. Constr. Build. Mater.260,119889.

18. Chen, H., Qi, G., Yang, J., Amini, F., 1995. Neural network for structural dynamic model identification. J. Eng. Mech. 121 (12), 1377–1381.

19. Chaabene,W.B.,Flah,M.,Nehdi,M.L.,2020.Machine learning prediction of mechanical properties of concrete: Critical review. Constr. Build. Mater.260,119889.

20. Chen, H., Qi, G., Yang, J., Amini, F., 1995.Neural network for structural dynamic model identification. J. Eng. Mech. 121 (12), 1377–1381.

21. Chaabene, W.B., Flah, M., Nehdi, M.L., 2020. Machine learning prediction of mechanical properties of concrete: critical review. Constr. Build. Mater. 260, 119889.

22. Shaqadan, A. Prediction of concrete mix strength using random forest model. Int. J. Appl. Eng. Res. 2016, 11, 11024–11029.

23. Wang,Q.; Ahmad, W.; Ahmad, A.; Aslam, F.; Mohamed, A.; Vatin, N.I. Application of Soft Computing Techniques to Predict the Strength of Geopolymer Composites. Polymers 2022, 14,1074. [CrossRef

24. Yuan, X.; Tian, Y.; Ahmad, W.; Ahmad, A.; Usanova, K.I.; Mohamed, A.M.; Khallaf, R. Machine Learning Prediction Models to Evaluate the Strength of Recycled Aggregate Concrete. Materials 2022, 15, 2823. [CrossRef] .

25. Farooq, F.; Amin, M.N.; Khan, K.; Sadiq, M.R.; Javed, M.F.F.; Aslam, F.; Alyousef, R. A Comparative Study of Random Forest and Genetic Engineering Programming for the Prediction of Compressive Strength of High Strength Concrete (HSC). Appl. Sci. 2020, 10, 7330. [CrossRef]

26. B.B. Adhikary, H. Mutsuyoshi, Prediction of shear strength of steel fiber RC beams using neural networks, Constr. Build. Mater. 20 (9) (2006) 801–811