applied sciences

MDPI

Article

A New Approach to Machine Learning Model Development for Prediction of Concrete Fatigue Life under Uniaxial Compression

Jaeho Son and Sungchul Yang *

School of Architectural Engineering, Hongik University, Sejong 30016, Korea
* Correspondence: scyang@hongik.ac.kr; Tel.: +82-44-860-2561

Abstract: The goal of this work is to show how machine learning models, such as the random forest, neural network, gradient boosting, and AdaBoost models, can be used to forecast the fatigue life (N) of plain concrete under uniaxial compression. Here, we developed our final machine learning model by generating the following three data files from the original data used in the work of Zhang et al.: (a) grouped data with the same input variable value and different output variable log$N$ value, (b) data excluding outliers selected by three or more outlier detection methods; (c) average data excluding outliers, created by averaging the grouped data after excluding outliers from among the grouped data. Excluding the sustained strength of the concrete variable, originally treated as the seventh input variable in the work of Zhang et al., resulted in improving the determination coefficient (R^2) values. Moreover, the gradient boosting model showed a high R^2 value at 0.753, indicating a high accuracy in predicting outcomes. Further analysis using data excluding outliers shows that the R^2 value increased to 0.803. Moreover, the average data excluding outliers provided the best R^2 value at 0.915. Finally, a permutation feature importance (PFI) analysis was carried out to determine the strength of the relationship between the feature and the target value for the gradient boosting model. The analysis results showed that the maximum stress level (S_{max}) and loading frequency (f) were the most significant input variables, followed by compressive strength (f'_c) and maximum to minimum stress ratio (R). Shape and height to width ratio (h/w) were the features with a non-significant influence on the model. This trend was previously confirmed by a Pearson and Spearman correlation analysis.

Keywords: fatigue; machine learning; stress level; frequency; compressive strength; specimen shape; outliers; correlation

check for updates

Citation: Son, J.; Yang, S. A New Approach to Machine Learning Model Development for Prediction of Concrete Fatigue Life under Uniaxial Compression. *Appl. Sci.* **2022**, *12*, 9766. https://doi.org/10.3390/app12199766

Academic Editor: Laurent Daudeville

Received: 27 August 2022
Accepted: 24 September 2022
Published: 28 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Concrete structures are subjected to repeated loading (N) from many sources, such as dead load and live loads in buildings, traffic loads in civil structures, or environmental loads, such as temperature and humidity changes. It is commonly known that concrete strength under repeated loading will be lower than that under static loading [1,2]. Concrete structures subjected to many repeated loadings will experience an increase in deflections, crack widths, and eventually lead to the reduction in durability and fatigue failure [3].

A classic fatigue equation for plain concrete is typically represented by an S-N diagram, where the stress level (S) is defined as the percentage of the static strength, with respect to the logarithm of N. Most previous research results on fatigue have been analyzed with a simple linear equation. However, it is well-known that the single S-N curve (known as a Wohler curve) is inappropriate to describe fatigue behavior [1], as it is affected by other factors.

In addition to S, concrete fatigue is affected by various factors, such as concrete compressive strength, concrete mix proportions, and loading parameters [1,4–33]. As stated in [1], unlike insensitivity to the details of mix design and concrete compressive strength, concrete is highly sensitive to fatigue loading parameters, such as maximum stress level (S_{max}), maximum stress to minimum stress level (R), frequency (f), and fatigue loading history [1].

Moreover, high strength concrete yields a different fatigue pattern, while various mix designs proportioned by different water-binder ratios, including the use of fibers, also produce different fatigue patterns [1]. Recently, incorporating supplementary cementitious materials (SCMs), such as slag, fly ash, metakaolin, and silica fume, in the concrete mix is widely regarded as the most economical means of improving durability and reducing CO_2 emission issues [34,35]. Thus, in the near future, it will be essential to understand the fatigue behavior of the waste materials, as well as the SCMs. However, the fatigue behavior of innovative concrete materials combined with the above-mentioned mixture constituents is difficult to estimate. In addition, the concrete structures will be exposed to diverse fatigue loading parameters, such as different stress levels and frequencies, as mentioned before. Therefore, the traditional statistical treatment of accurately predicting concrete fatigue behavior has reached its limit, due to its inability to consider the complicated combined effects of those influential parameters.

To overcome this limitation inherent in the traditional regression-based statistics methods, a machine learning (ML) method has been introduced to solve complex concrete material properties in terms of durability, as well as mechanical strengths [4,36–62]. In recent years, the ML method has become more widely used for structural and material design in civil engineering. Various ML methods have been frequently used since 2020 for predicting basic mechanical strength properties [36–61] and mixture optimizations [36,39,62]. The main concrete property that has been predicted using the ML method is the compressive strength of various concretes, such as normal concrete [36–41], high performance concrete (HPC) [38,42–44], concrete with industrial wastes, including supplementary cementitious materials (SCMs) [45–54], recycled aggregate (RA) concrete [52,55–57], geopolymer concrete [58,59], and concrete with fibers [53]. In addition, the split tensile strength [45,57,60] and modulus of elasticity [61] of concrete were predicted by using ML techniques.

Among the ML methods, artificial neural network (ANN) models are widely used [36–38,42,43,45–50,52,54,55,58–60,62]. In addition to ANN, the prediction of mechanical strength properties and mix proportions of concrete using other regression models has recently gained popularity, including the use of support vector regression [39,47,52], decision tree [40,57,61,62], random forest [36,44,47,56], AdaBoost [40,41,52,57,59,61], gradient boost [40,53], and ensemble algorithm [51,61].

In 2019, an ANN-based concrete fatigue strength model was proposed by Abambres and Lantsoght [63]. They used 203 data points gathered from the literature. Predicted values analyzed from the ANN model were compared to the existing code expressions. Their ANN model includes the compressive strength of concrete, maximum stress level, and minimum stress level. In 2021, a strength degradation model of concrete under fatigue loading was proposed by Zhang et al. [4] using several ML algorithms, such as the random forest, support vector machine, and artificial neural network models. About 1000 experimental data were collected from various independent experiments [5–33]. Seven independent variables were chosen in their study, including the compressive strength of concrete, sustained strength of concrete, height to width ratio and shape of the test specimens, maximum stress level, minimum to maximum stress ratio, and loading frequency. The analysis results revealed that the random forest model produces the highest value of the correlation coefficient at 0.85.

Due to the nature of the fatigue strength test, outliers can remarkably occur in this test compared to other material strength properties tests. In statistics, an outlier is a data point that differs significantly from other observations [64,65]. An outlier may be due to variability in the measurement or it may indicate experimental error; the latter is sometimes excluded from the data set. There are various methods of outlier detection, such as Grubbs's test [64], Chauvenet's criterion [66], Peirce's criterion [67], Dixon's Q-test [68], the generalized extreme studentized deviation test [69], Thompson and Tau test [70], and the IQR-test [71,72].

In this study, 1300 samples of experimental data [5–33] of concrete fatigue tests originally carried out by Zhang et al. [4] were treated using 4 kinds of machine learning models

(artificial neural network, random forest, and the gradient boosting and AdaBoost method). Unlike previous studies, this research adopts six independent values, excluding only the sustained strength of the concrete variable used from the work of Zhang et al. [4]. For our approach, three data files were generated to compare the actual number of fatigue repetition values (log$N$) against the predicted values (log$N$). The first data file uses the entire original dataset, which was treated by Zhang et al. [4]. However, unlike Zhang et al. [4], our research adds the second data file with the grouping data and the third data file that excludes outliers. In this work, Chauvenet's criterion, Pierce's criterion, the Thompson–Tau criterion and the IQR method were adopted to remove outliers. Finally, a permutation feature importance (PFI) analysis was carried out to determine which input variables are the most critical or minor in the fatigue life model. Our novel approach allows better fatigue life prediction than Zhang et al. [4]'s approach.

2. Input and Output DATA (Independent and Dependent Variables)

Six basic input features (variables) that influence the fatigue life span of plain concrete under a uniaxial fatigue test in compression were chosen, as shown in Table 1. One output variable is the logarithm value of the maximum number of cycles at failure, representing the fatigue life of the test. The number of the first group of the key input variables, which are related to the material and dimensional properties of the test specimens, included the compressive strength of concrete (f'_c), height to width ratio (h/w), and shape of the test specimens. The other three variables that reflect the loading conditions of the fatigue test specimens include the maximum stress level (S_{max}), minimum stress to maximum stress ratio (R), and loading frequency (f).

Table 1. Statistical features of original data.

Type	Variables	Mean	Median	Dispersion	Min.	Max.
Independent	f'_c (MPa)	60.8	56	0.568	11.6	145.1
	h/w	1.96	2	0.407	1	3
	Shape (1)	2.14	2	0.41	1	3
	S_{max}	0.805	0.8	0.0977	0.457	0.95
	R	0.180	0.118	0.795	0.0143	0.667
	f (Hz)	8.46	5	2.70	0.0625	150
Dependent	Log$N$	5.10	3.55	0.537	0.699	6.78

(1) Since shape is a categorical variable, the statistical features expressed in the table may not be meaningful.

This study covers low-strength hydraulic concrete (10~30 MPa), ordinary concrete (30~60 MPa), and high-strength concrete (60~120 MPa). The h/w of the test specimens ranged from 1.0 to 3.0, and the specimen's shape includes the cube, prism, and cylinder. The loading conditions were also greatly diverse, with the S_{max} ranging from 0.457 to 0.95, the R covering 0 to about 0.67, and a loading frequency ranging from 0.0625 to 150 Hz. The dataset used in this study is summarized below.

f'_c: the compressive strength of concrete by MPa;
h/w: height to width ratio of the tested specimens;
Shape: shape of the test specimens;
S_{max}: maximum stress level;
R: minimum stress to maximum stress ratio;
f (Hz): loading frequency by Hz;
Log$N$: logarithm number of cycles to failure of the specimen.

3. DATA Preparation for the Developed Model

Three data files were generated and used to develop the final ML model. Each data file is described below.

1. ORIGINAL DATA. These are data used in Zhang's paper, directly collected by the authors from papers [5–33]. The full-data spreadsheet is available in the Supplementary Materials. These are used as the reference data for this study. A total of 1298 data were collected, and statistical features such as the mean, median, dispersion, minimum, and maximum values of independent and dependent variables are summarized in Table 1. The ORIGINAL DATA were grouped by the same input variable value.
2. DATA Excluding OUTLIERS. If there are outliers in the group, these are the data created after removing them. These data are used as a basis for determining the average value after removing the outliers. A total of 1252 data were generated. Statistical features such as the mean, median, dispersion, minimum and maximum values of independent and dependent variables are summarized in Table 2.
3. AVERAGE DATA Excluding OUTLIERS. These are the data created by averaging the grouped data after excluding outliers from among the grouped data. In this process, the total number of data was reduced to 310. Statistical features such as the mean, median, dispersion, minimum and maximum values of independent and dependent variables are summarized in Table 3.

Table 2. Statistical feature of data excluding outliers.

Type	Variables	Mean	Median	Dispersion	Min.	Max.
Independent	f'_c (MPa)	60.7	56	0.570	11.6	145.1
	h/*w*	1.97	2	0.406	1	3
	Shape	2.14	2	0.41	1	3
	S_{max}	0.805	0.8	0.0988	0.457	0.95
	R	0.180	0.118	0.794	0.0143	0.667
	f (Hz)	8.48	5	2.70	0.0625	150
Dependent	Log*N*	5.05	3.51	0.523	0.699	6.55

Table 3. Statistical feature average data excluding outliers.

Type	Variables	Mean	Median	Dispersion	Min.	Max.
Independent	f'_c (MPa)	53.6	41.6	0.613	11.6	145.1
	h/*w*	2.20	2	0.298	1	3
	Shape	2.42	3	0.29	1	3
	S_{max}	0.779	0.8	0.116	0.457	0.95
	R	0.156	0.124	0.774	0.0143	0.667
	f (Hz)	9.71	5	2.66	0.0625	150
Dependent	Log*N*	5.14	3.91	0.424	1	6.48

Tables 1–3 illustrate the statistical analysis of variables, showing the numerous mathematical descriptions of the input and output values for each data set. Tables 4–6 describe the data process in which parts of the data from reference [5] are used to illustrate the process more clearly as an example. Table 4 represents a part of the grouped data in which data sets with the same input variable value, but with different output variable values, are grouped together. Table 4 consists of two groups. Group 1 is a data set with an f'_c value of 56 MPa, *h*/*w* value of 1, shape value of 1, S_{max} value of 0.85, R value of 0.3, and f value of 4 Hz, but with different output values *N*. Group 2 is a data set with an f'_c value of 56 MPa, *h*/*w* value of 1, shape value of 1, S_{max} value of 0.85, R value of 0.3, and *f* value of 1 Hz, but with different output values *N*.

To designate whether there are outlier data in each group, four commonly used outlier detection methods [70,71] were performed. If three or more of them were designated as an outlier, they were excluded from the data. The four methodologies are as follows:

1. Outlier detection method using Chauvenet's criterion;
2. Outlier detection method using Peirce's criterion;
3. Outlier detection method using Thompson–Tau criterion;
4. Outlier detection method using IQR (inter quartile range) criterion.

We performed all four of these methodologies on each group of data to determine which values were detected as outliers. All four of these outlier detection methodologies detected the N value of 22,570 (see Table 4) as an outlier for the Group 1 data. On the other hand, for the data in Group 2, the N value of 1571 (see Table 4) was detected as an outlier only by the Thompson–Tau methodology, but was not detected as an outlier by the other three methodologies. Table 5 represents the grouped data in which the data set with an N value of 22,570 is removed from Group 1. Even after removing outliers, the values of different output variables are recorded as experimental values in the same input variable values. With these data, it is difficult to make an accurate prediction model as long as the current input variables are maintained. One must suppose that the user predicts a function of y = sin(x). If several different values of the y experimental value for the sin(x) value are matched when x = 30, it will be difficult to create an ML model that predicts the sin(x) function. Therefore, in the case of grouped data having the same input variable value and different output variable values to eliminate this situation, the average value of all other output variable values is obtained. One average value is used as the output value for the same specific input variable value. This should provide more reasonable data for creating predictive ML models. Table 6 represents the average grouped data in Table 5.

Table 4. Outlier identified in grouped data.

Group	f'_c (MPa)	h/w	Shape	S_{max}	R	f (Hz)	N
1	56	1	1	0.85	0.3	4	8411
1	56	1	1	0.85	0.3	4	821
1	56	1	1	0.85	0.3	4	2485
1	56	1	1	0.85	0.3	4	1660
1	56	1	1	0.85	0.3	4	13,020
1	56	1	1	0.85	0.3	4	***22,570***
1	56	1	1	0.85	0.3	4	9521
1	56	1	1	0.85	0.3	4	4192
1	56	1	1	0.85	0.3	4	170
1	56	1	1	0.85	0.3	4	1578
1	56	1	1	0.85	0.3	4	1222
1	56	1	1	0.85	0.3	4	133
1	56	1	1	0.85	0.3	4	7038
2	56	1	1	0.85	0.3	1	282
2	56	1	1	0.85	0.3	1	23
2	56	1	1	0.85	0.3	1	759
2	56	1	1	0.85	0.3	1	1351
2	56	1	1	0.85	0.3	1	85
2	56	1	1	0.85	0.3	1	157
2	56	1	1	0.85	0.3	1	479
2	56	1	1	0.85	0.3	1	368
2	56	1	1	0.85	0.3	1	833
2	56	1	1	0.85	0.3	1	*1571*

Table 5. Grouped data excluding outliers.

Group	f'_c (MPa)	h/w	Shape	S_{max}	R	f (Hz)	N
1	56	1	1	0.85	0.3	4	8411
1	56	1	1	0.85	0.3	4	821
1	56	1	1	0.85	0.3	4	2485
1	56	1	1	0.85	0.3	4	1660
1	56	1	1	0.85	0.3	4	13,020
1	56	1	1	0.85	0.3	4	9521
1	56	1	1	0.85	0.3	4	4192
1	56	1	1	0.85	0.3	4	170
1	56	1	1	0.85	0.3	4	1578
1	56	1	1	0.85	0.3	4	1222
1	56	1	1	0.85	0.3	4	133
1	56	1	1	0.85	0.3	4	7038
2	56	1	1	0.85	0.3	1	282
2	56	1	1	0.85	0.3	1	23
2	56	1	1	0.85	0.3	1	759
2	56	1	1	0.85	0.3	1	1351
2	56	1	1	0.85	0.3	1	85
2	56	1	1	0.85	0.3	1	157
2	56	1	1	0.85	0.3	1	479
2	56	1	1	0.85	0.3	1	368
2	56	1	1	0.85	0.3	1	833
2	56	1	1	0.85	0.3	1	*1571*

Table 6. Average grouped data excluding outliers.

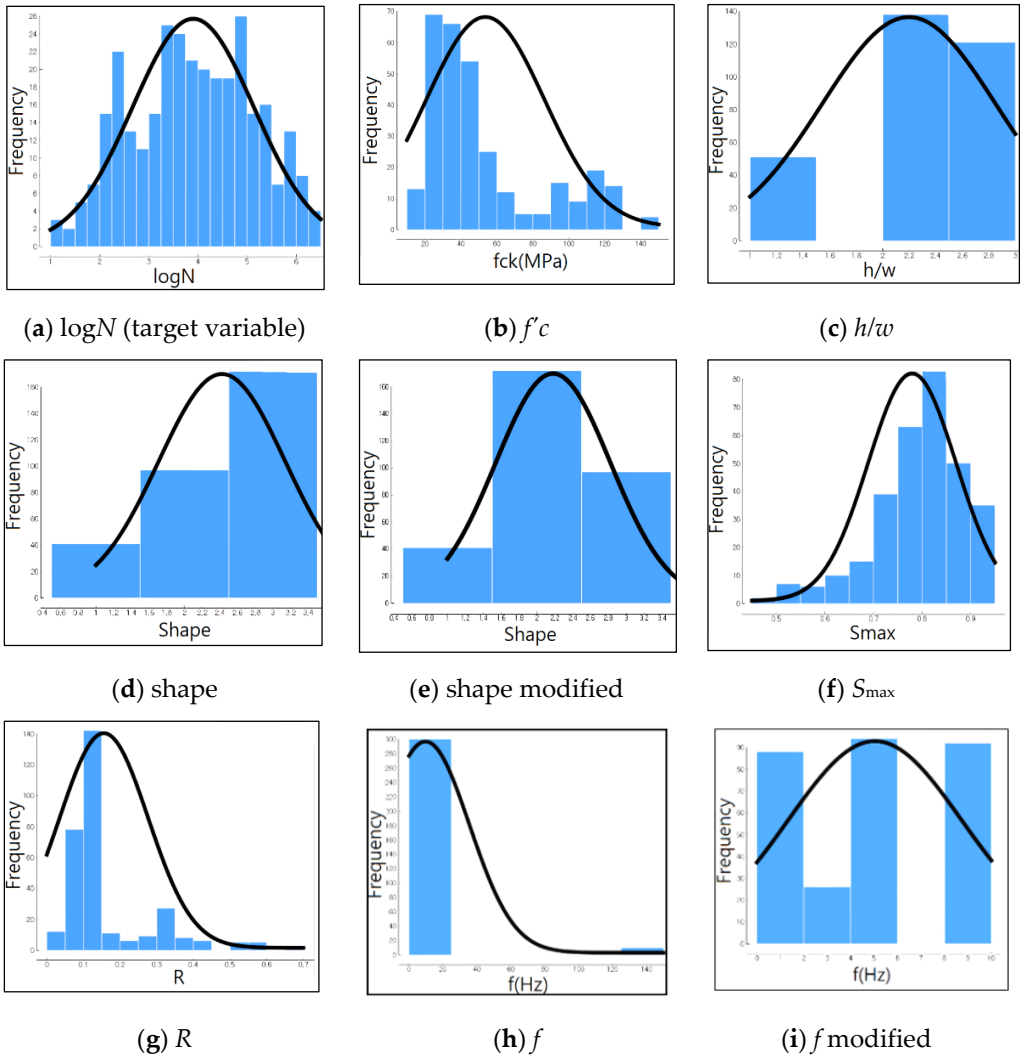
Group	f'_c (MPa)	h/w	Shape	S_{max}	R	f (Hz)	N
1	56	1	1	0.85	0.3	4	4187.6
2	56	1	1	0.85	0.3	1	590.8

Figure 1 depicts the relative frequency distributions of the six input variables and one output variable. The shape variable is not only a numerical variable, but also a categorical variable. In the model, shape = 1 is represented as a cube, shape = 2 as a prism, and shape = 3 as a cylinder. Since the numbers are meaningful in determining the category, (d) in Figure 1 can be changed to (e), which is more suitable for normal distribution. The f variable appears to be unsuitable for normal distribution, since some high-frequency values of 10 Hz exist in the data. The f variable appears to be unsuitable for normal distribution, since some high-frequency value of 10Hz exist in the data. If these high-frequency data are removed, the rest of the data are much more suitable for normal distribution in Figure 1i.

The relationships between various independent variables and log$N$ are plotted in Figure 2. Although not strong, one linear relationship is identified in Figure 2a (log$N$ vs. S_{max}). All other plots show non-linear behavior.

The most commonly used methods in correlation analysis are the Pearson correlation analysis and Spearman correlation analysis. Pearson correlation evaluates the linear relationship and direction between two variables using the values of the variables. Spearman correlation evaluates a monotonic relationship between two variables. In a monotonic relationship, the two variables tend to change together, but do not necessarily change at

a constant rate. The Spearman correlation coefficient is based on ranked values for each variable, not on raw data.

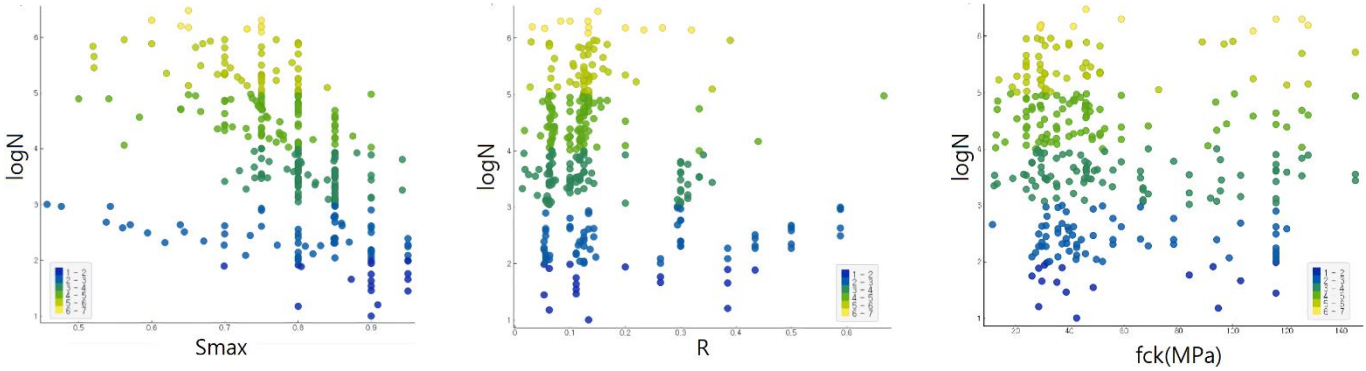


(**a**) log$N$ (target variable) (**b**) $f'c$ (**c**) h/w

(**d**) shape (**e**) shape modified (**f**) S_{max}

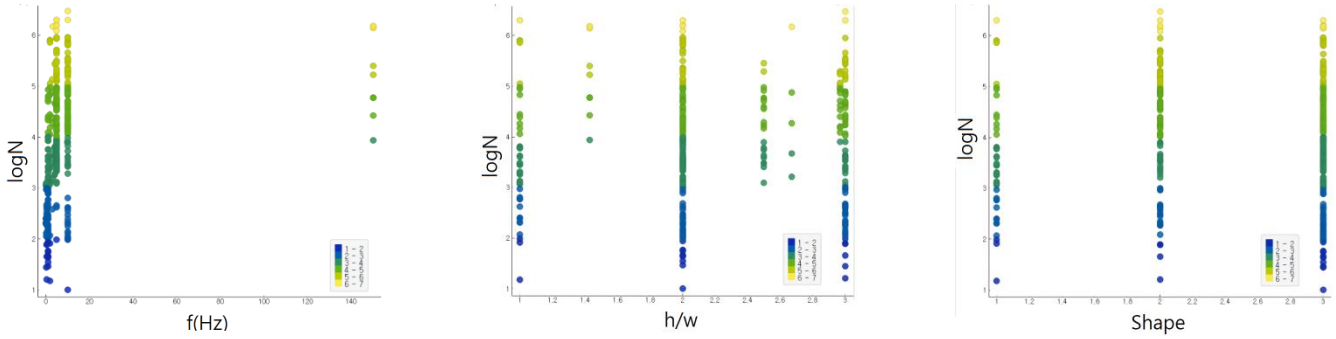
(**g**) R (**h**) f (**i**) f modified

Figure 1. Distribution of frequency of the variables used to run the models.



(**a**) log$N$ vs. S_{max} (**b**) log$N$ vs. R (**c**) log$N$ vs. $f'c$

Figure 2. *Cont.*



(**d**) log$N$ vs. f (**e**) log$N$ vs. h/w (**f**) log$N$ vs. shape

Figure 2. Scattered plot between independent and dependent variables.

Table 7 summarizes the Pearson correlation coefficient and Spearman correlation coefficient of the data used for our ML model. According to the Pearson correlation coefficient, S_{max} and log$N$ have a negative solid linear relationship, while f has a positive and, R has a negative moderate linear relationship with log$N$. f'_c, shape, and h/w have a non-significant linear relationship with log$N$. According to the Spearman correlation coefficient, S_{max} has a negative and f has a positive significant; f'_c has a negative moderate; R, shape, and h/w have a negligible monotonic relationship with log$N$.

Table 7. Pearson and Spearman correlation coefficient.

Coefficient	S_{max}	f (Hz)	R	Shape	f'_c (MPa)	h/w
Pearson (log$N$)	−0.460	+0.268	−0.248	+0.008	−0.088	+0.019
Spearman (log$N$)	−0.526	+0.532	−0.064	−0.020	−0.154	−0.003

Therefore, a complex relationship rather than linear mapping is critical for capturing variation and interaction. This is why it is necessary to create predictive systems using ML methods.

4. Methodology

Four types of predictive regression models were developed in this study using a neural network model, a random forest model, a gradient boosting model, and an AdaBoost model.

4.1. Neural Network

Artificial neural networks (ANNs) are an efficient learning tool inspired by biological neural networks. They are composed of the following three types of layers: input, hidden, and output. Training data are fed to the input layer, and the predicted value is calculated by the output layer through the hidden layer. Using the backpropagation algorithm, the weights connecting the input layer, the hidden layer, and the output layer are updated in a way that minimizes the error between the calculated value and the measured value [73,74]. Figure 3 shows the general structure of ANNs.

4.2. Random Forest

Random forest is one of the ensemble models. It is a method of forming multiple decision trees, passing new data through each tree, and voting based on the classification results of each tree, and then selecting the result with the most votes as the final classification result (see Figure 4). A random forest model can be viewed as a forest composed of random trees. Some trees in the random forest may be over fitted; however, there are many other trees that make up the forest. Therefore, there is no significant impact on the model [4,75].

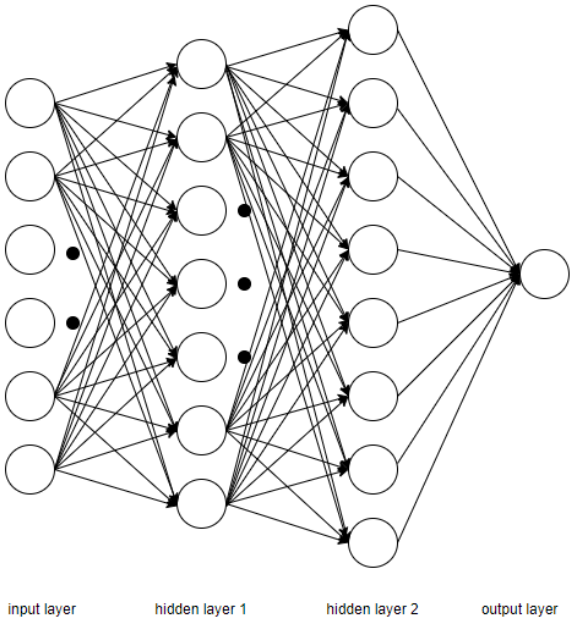


Figure 3. Structure of neural network model.

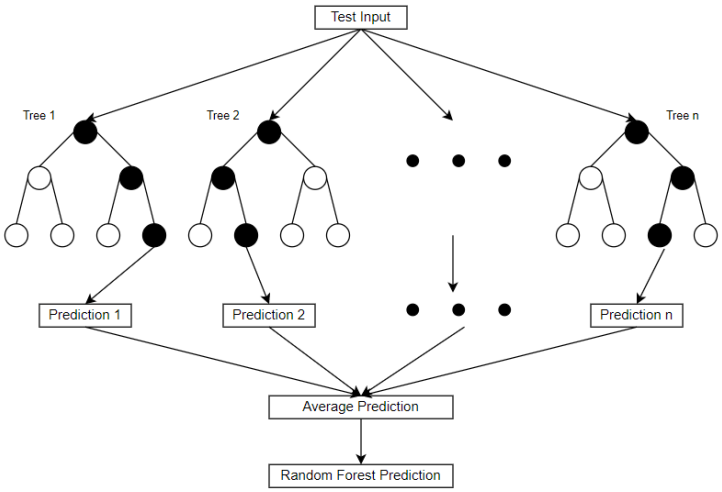


Figure 4. Structure of the random forest model.

4.3. Boosting Model

Boosting is an ensemble method that combines several weak learners to create a strong learner. It improves the performance of the next learning model, while reducing the errors of the previous learning model. There are several types of boosting methods, but AdaBoost and gradient boosting are representative models [75].

4.3.1. Gradient Boosting Method

Gradient boosting uses gradient descent to minimize the loss function of a model by adding weak learners (see Figure 5). By training the model's residuals, this gives more importance to misclassified observations. The contribution of each weak learner to the final prediction is based on a gradient optimization process to minimize the overall error of the strong learner [75,76].

4.3.2. AdaBoost Method

AdaBoost, or adaptive boosting, is a type of boosting algorithm that generates a final strong classifier by collecting weighted weak classifiers (see Figure 6) [75,77].

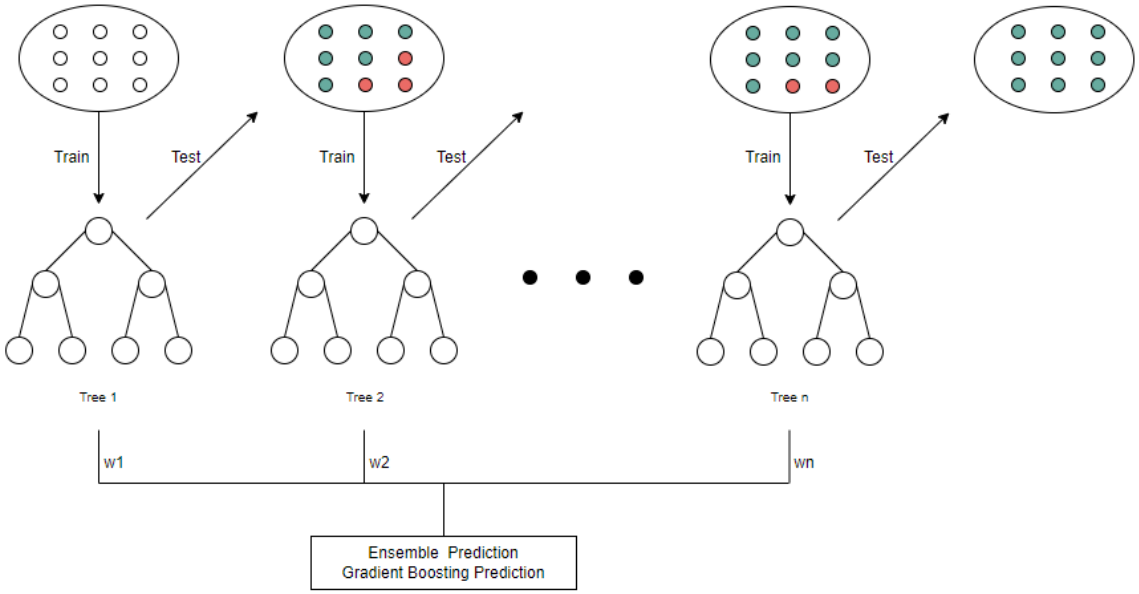


Figure 5. Structure of the gradient boosting model.

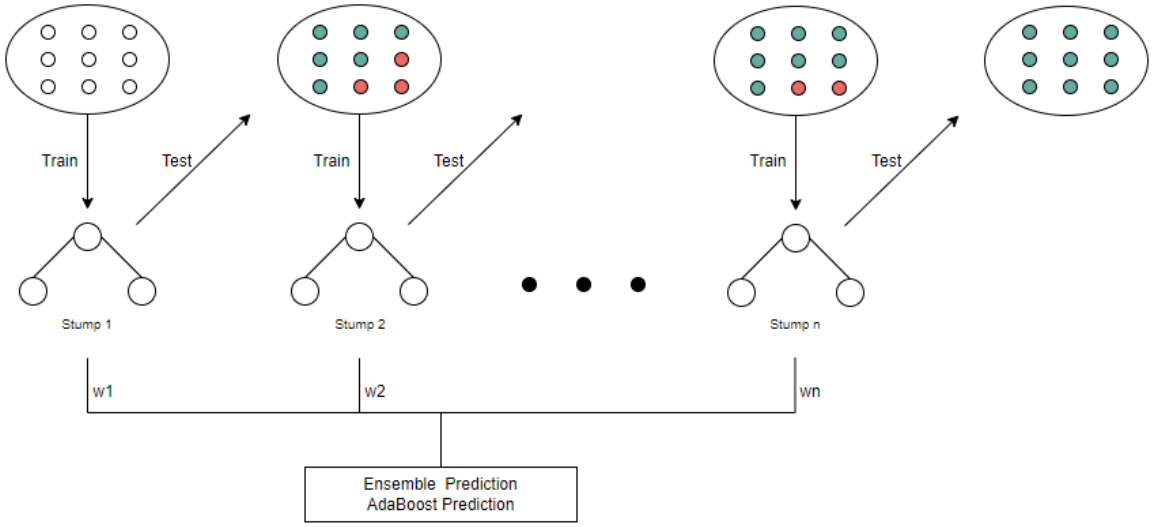


Figure 6. Structure of the AdaBoost model.

5. Model Development

The models for fatigue prediction were developed using Orange software, which is a popular open-source machine learning technology platform for statistical computing and data mining [78,79]. All data analysis in this research was carried out using Orange software (version 3.32.0, developed at Bioinformatics Laboratory, Faculty of Computer and Information Science, University of Ljubljana, Ljubljana, Slovenia, together with open source community.), which provides the most prevalent supervised ML algorithms. These algorithms were used to develop our novel ML model. Information regarding the input parameters and implementation of each machine learning algorithm are summarized in documentation and can be found at (https://orangedatamining.com/widget-catalog/, accessed on 4 April 2022). Orange provides a platform for developing the predictive modeling with big data. The schematic model developed using Orange Software is presented in Figure 7, and the specific parameters of each proposed model are shown in Figures 8–11. Unfortunately, the Orange 3 software used for this study does not have an optimizer function that automatically finds the hyper-parameters of the model. Thus, starting with the

default parameters provided by Orange 3, the authors manually adjusted the parameters to generate feasible output for each ML model.

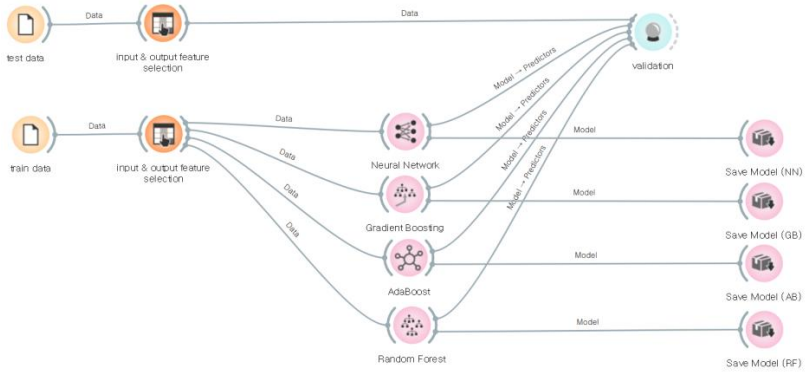


Figure 7. Model developed using Orange software [79].

In order to develop an ANN network model, the user has to set several important parameters, which are as follows. The number of hidden layers is set to two, and there are seven and eight neurons in each hidden layer, as shown in Figure 8. The rectified linear unit function is selected as the activation function for the hidden layer. As a solver for weight optimization, a stochastic gradient-based optimizer called Adam is used. As a regularization parameter, commonly called alpha, 0.0004 is used. Replicable training is allowed.

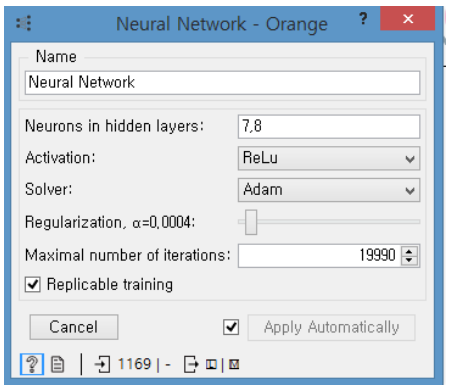


Figure 8. Parameters of the proposed ANN model [79].

In order to develop a random forest model, the user has to set several important parameters, which are as follows. As shown in Figure 9, 50 decision trees are included in the forest. Four attributes will be arbitrarily drawn for consideration at each node. Replicable training was permitted, while balance class distribution was not. The limit depth of individual trees has not been determined. One must select five subsets as the smallest subset that can be split.

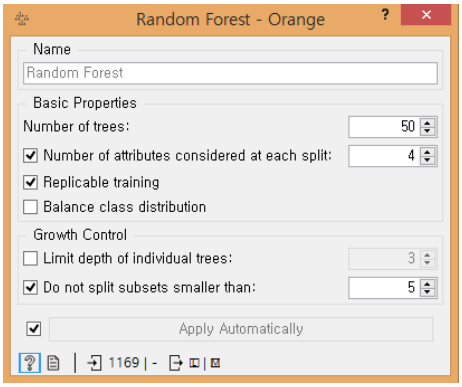


Figure 9. Parameters of the proposed random forest model [79].

In order to develop a gradient boosting model, the user has to set several important parameters, which are as follows. As shown in Figure 10, 150 gradient boosted trees are specified. A larger number usually results in better performance. The boosting rate is set to 0.2. Replicable training is allowed. The maximum depth of the individual tree is set to 4. One must select three subsets as the smallest subset that can be split. The fraction of training instances is set to 1. One must specify the percentage of the training instances for fitting the individual tree.

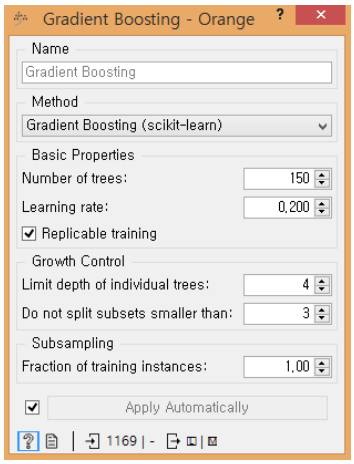


Figure 10. Parameters of the proposed gradient boosting model [79].

In order to develop an AdaBoost model, the user has to set several important parameters, which are as follows. The number of estimators is set to 50, as shown in Figure 11. The learning rate is set to 1. It determines to what extent the newly acquired information will override the old information. The number of 1 means that the agent considers only the most recent information. The number of 3 is set as a fixed seed to enable reproduction of the results. We decided to use SAMME as the classification algorithm, which updates the base estimator's weights with classification results. Among the regression loss function options, the linear option is selected.

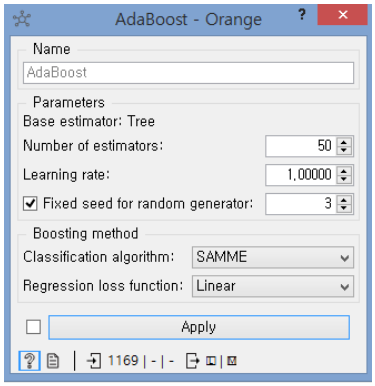


Figure 11. Parameters of the proposed AdaBoost model [79].

6. Results and Discussion

6.1. Model Developed with Original Data

In our novel ML model, about 1300 fatigue test results from the 29 paper data sets used by Zhang et al. [4] were collected and organized. For training and testing of the model, 90% of the total data was used for training and 10% was used for testing.

- Total data sets: 1298 data sets;
- Training data sets: 1169 data sets;
- Test data sets: 129 data sets.

The four ML models (random forest, neural network, gradient boosting, and AdaBoost) were run, and the results of training and testing for each model are shown in Table 8a,b below. Using the same data sets, Zhang et al. [4] reported that the MSE and correlation coefficient (r) from the random forest model are 0.44 and 0.85, respectively. The determination coefficient (R^2) in that case was about 0.723. In this study, excluding the sustained strength of the concrete variable, which was originally treated as the seventh input variable in the work of Zhang et al. [4], resulted in improving the MSE and R^2 values. Moreover, Table 8a,b shows that the gradient boosting model with the value of the minimum error and a high R^2 value is indicates high accuracy in predicting outcomes. Additionally, Zhang et al. [4] reported that the MSE and r values from the typical traditional regression fatigue formulae (represented as *S-N-T-R*) in terms of R, S_{max}, and rate of loading (T) were 1.46 and 0.50, respectively.

Table 8. (**a**) Result of ML models with training original data. (**b**) Result of ML models testing original data.

(**a**)				
Model	**MSE**	**RMSE**	**MAE**	R^2
Random forest	0.351	0.592	0.411	0.768
Neural network	0.551	0.742	0.547	0.635
Gradient boosting	0.334	0.578	0.393	0.779
AdaBoost	0.341	0.584	0.389	0.774
(**b**)				
Model	**MSE**	**RMSE**	**MAE**	R^2
Random forest	0.312	0.559	0.402	0.740
Neural network	0.416	0.645	0.461	0.655
Gradient boosting	0.297	0.545	0.390	0.753
AdaBoost	0.315	0.561	0.389	0.738

MSE: mean squared error, RMSE: root mean squared error, MAE: mean absolute error; R^2: coefficient of determination.

6.2. Model Developed with Data Excluding Outliers

The data used in this model are the data that excluded outliers among the data used in 6.1. A total of 46 data sets, approximately 3.5 % of the original total data sets, are treated as outliers. For training and testing of the model, 90% of the total data was used for training and 10% was used for testing. In addition, 90-10, 85-15, and 80-20 are the ratios of the most used training and testing data. When developing an ML model using average data, the number of data is reduced. Therefore, in order to secure as much of the training data as possible, a ratio of 90-10 was used.

- Total data sets: 1252 data sets;
- Train data sets: 1127 data sets;
- Test data sets: 125 data sets.

The four machine learning models (random forest, neural network, gradient boosting, and AdaBoost) were run, and the results of training and testing for each model are shown in Table 9a,b below. As shown in Table 9a, the gradient boosting model with training data provides the highest determination coefficient, R^2 = 0.809, followed by R^2 = 0.805 from the AdaBoost model, and then 0.795 from the random forest model. The neural network gave the lowest R^2 value at 0.726. As shown in Table 9b, the gradient boosting model provides the highest determination coefficient, R^2 = 0.803, followed by R^2 = 0.794 from the AdaBoost model, and then 0.791 from the random forest model. The neural network gave the lowest R^2 value at 0.726.

Table 9. (**a**). Result of ML models with training data excluding outliers. (**b**). Result of ML models with testing data excluding outliers.

(**a**)				
Model	**MSE**	**RMSE**	**MAE**	R^2
Random forest	0.296	0.544	0.379	0.795
Neural network	0.479	0.692	0.510	0.668
Gradient boosting	0.275	0.524	0.359	0.809
AdaBoost	0.282	0.531	0.355	0.805
(**b**)				
Model	**MSE**	**RMSE**	**MAE**	R^2
Random forest	0.321	0.566	0.414	0.791
Neural network	0.419	0.647	0.500	0.726
Gradient boosting	0.301	0.549	0.397	0.803
AdaBoost	0.315	0.561	0.417	0.794

6.3. Model Developed with Average Data Excluding Outliers

For the data used in Section 6.2, the data sets used with the same input variables have different output variable values. If there are many cases similar to this, it may be difficult to train the ML model. To eliminate this, one output data set value should be matched to one possible input data set value. For this purpose, average data are used. For training and testing of the model, 90% of the total data was used for training and 10% was used for testing.

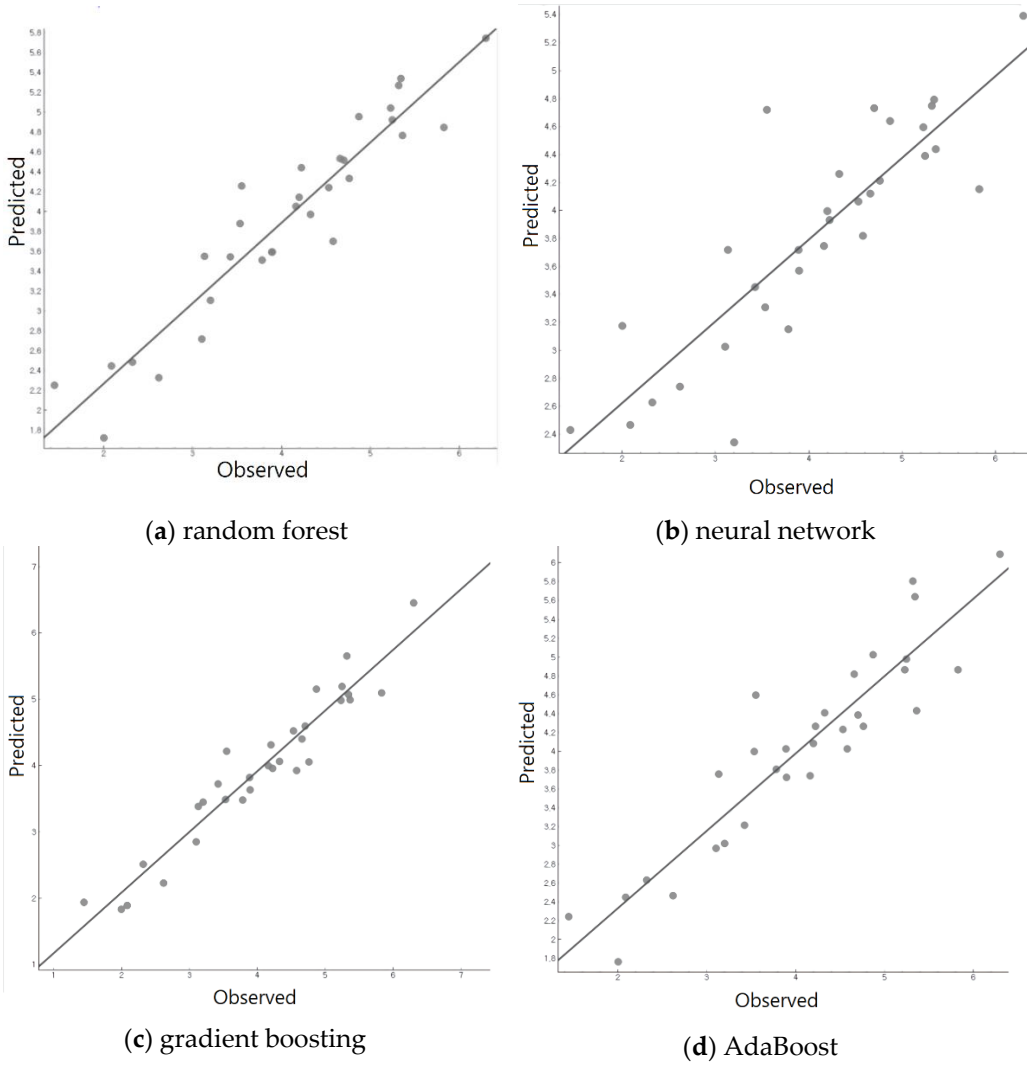
- Total data sets: 310 data sets;
- Training data sets: 279 data sets;
- Test data sets: 31 data sets.

The four machine learning models (random forest, neural network, gradient boosting, and AdaBoost) were run, and the results of training and testing for each model are shown in Table 10a,b below. As tabulated in Table 10a, the gradient boosting model with training data provides the highest determination coefficient, R^2 = 0.982, followed by R^2 = 0.973 from AdaBoost, then 0.887 from the random forest model. The neural network model showed the lowest R^2 value as 0.679. As tabulated in Table 10b, the gradient boosting model provides the highest determination coefficient, R^2 = 0.915, followed by R^2 = 0.893 from the random forest model, then 0.876 from the AdaBoost model. The neural network model showed the lowest R^2 value as 0.730. Three sets of data were used to develop the ML models in this study. The MSE, RMSE, MAE, and R^2 calculated with the average data excluding outliers were compared to the MSE, RMSE, MAE, and R^2 calculated with both the original data and the grouped data excluding outliers. As a result of comparing the values in Tables 8–10, the ML model developed with average data excluding outliers most closely matched the predicted value and the observed value.

Figure 12 depicts the actual values against the predicted values of log$N$ for machine learning models developed with the average data excluding outliers. The results of the gradient boosting model fit a straight line better than the other ML models, which indicates that the gradient boosting model is more accurate for predicting the log$N$. The scattered data of the gradient boosting model are closer to the linear regression line than the scattered data of the other models. Compared to the other models, the scatter plot of the neural network model does not fit well and its prediction is slightly off, which has a larger dispersity of scatter points. Among the four ML models developed with the average data excluding outliers, the gradient boosting model most closely fits the observed data.

Table 10. (**a**). Result of ML models with training average data excluding outliers. (**b**). Result of ML models with testing average data excluding outliers.

(**a**)				
Model	**MSE**	**RMSE**	**MAE**	R^2
Random forest	0.175	0.418	0.303	0.887
Neural network	0.495	0.704	0.534	0.679
Gradient boosting	0.027	0.166	0.094	0.982
AdaBoost	0.041	0.204	0.101	0.973
(**b**)				
Model	**MSE**	**RMSE**	**MAE**	R^2
Random forest	0.145	0.381	0.288	0.893
Neural network	0.367	0.606	0.493	0.730
Gradient boosting	0.115	0.339	0.280	0.915
AdaBoost	0.168	0.410	0.304	0.876



(**a**) random forest (**b**) neural network

(**c**) gradient boosting (**d**) AdaBoost

Figure 12. Predicted vs observed data in model developed.

The gradient boosting model often achieves state-of-the-art results on tabular data [80]. It is one of the most powerful ensemble algorithms that often has the highest predictive accuracy [81–83], and the results of this study show no exception; the gradient boosting model outperformed all the other ML models tested here.

Finally, the results of the developed models using the training average data and testing average data are shown in Figure 13. The gradient boosting model has the highest value of R^2 with the training dataset and testing dataset.

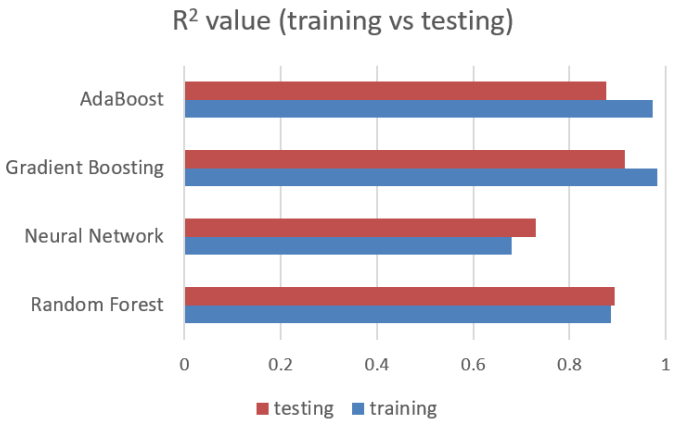


Figure 13. R^2 value (training vs. testing).

6.4. Sensitivity Analysis of ML Models

Sensitivity analysis was performed to find a better ML model with various training and testing ratios. The results of the sensitivity analysis are summarized in Table 11 and Figure 14. All ML models show the highest R^2 value when the training and testing ratio is 90:10. When the training and testing ratio is 90:10, the R^2 value of the GB model is 0.915, which is the best value among the sensitivity analysis results.

Table 11. Sensitivity analysis of ML models with different training and testing ratios.

Model	R^2 (75:25)	R^2 (80:20)	R^2 (85:15)	R^2 (90:10)
Random forest	0.756	0.751	0.825	0.893
Neural network	0.677	0.659	0.681	0.730
Gradient boosting	0.811	0.823	0.882	0.915
AdaBoost	0.742	0.749	0.839	0.876



Figure 14. Results of sensitivity analysis with various training and testing ratios.

6.5. Comprehensive Evaluation of ML Models

In addition to the classic model performance evaluation indices, such as R^2, MSE, MAE, new indices, such as VAF, PI, and $A_{10-index}$, are proposed to assess the efficiency of the developed models by Menemaran et al. [84]. It was noted that the smaller RMSE, MAE, PI indicate more trustable statistical impressions [84]. PI and $A_{10-index}$ are represented by Equations (1) and (2) [84].

$$PI = \frac{1}{|\bar{t}|} \frac{RMSE}{\sqrt{R^2 + 1}} \quad (1)$$

$$A_{10-index} = \frac{m_{10}}{M} \quad (2)$$

$\bar{t}$ is the mean of the observed values. In addition, M represents the sample number, and m_{10} is the number of data with a ratio of the measured to predicted value between 0.9 and 1.1 [84].

In this study, five model performance indices (RMSE, MAE, R^2, $A_{10-index}$, PI) were assessed in order to carry out comprehensive comparison. The models were scored from 1 to 4 based on each of the five indices; then, the scores were summed to assign a total score for each model. The results for this comparison score are listed in Table 12. Table 12 shows that the gradient boosting model has the best performance. On the other hand, the neural network model has the lowest accuracy for the testing data, respectively. Furthermore, the Taylor diagram of the four developed ML models is presented in Figure 15. It can be observed from the graph that the gradient boosting model has the best performance, while the neural network model has the worst performance with the average data excluding outliers.

Table 12. Comprehensive evaluation of ML models.

Model	RMSE	Score	MAE	Score	R^2	Score	$A_{10-index}$	Score	PI	Score	Total Score
RF	0.381	3	0.288	3	0.893	3	0.323	3	0.048	3	15
NN	0.606	1	0.493	1	0.730	1	0.290	2	0.081	1	6
GB	0.339	4	0.280	4	0.915	4	0.387	4	0.043	4	20
AB	0.410	2	0.304	2	0.876	2	0.226	1	0.052	2	9

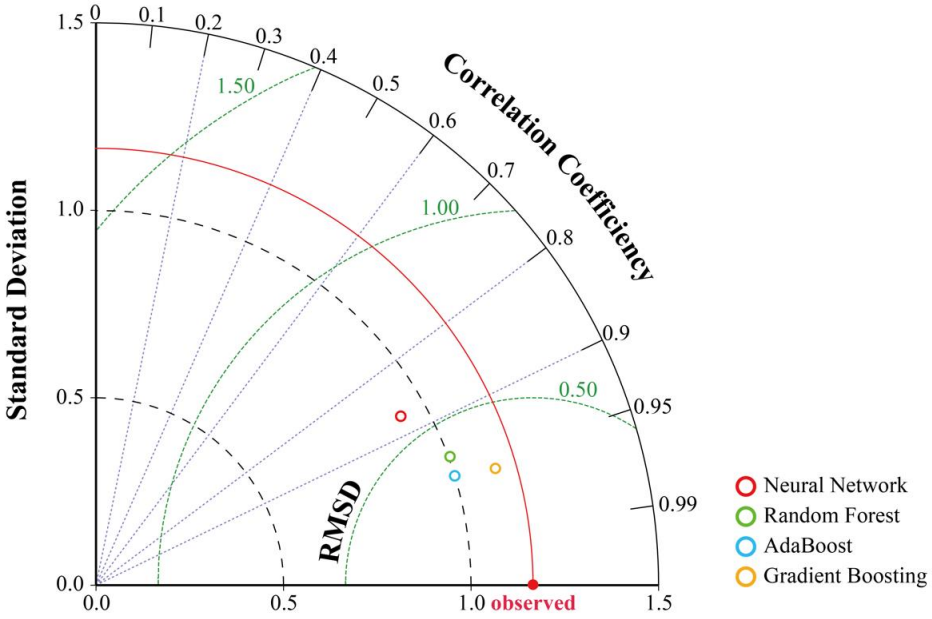


Figure 15. Taylor diagram of ML model developed.

6.6. Permutation Feature Importance

The correlation used to explain the model is, in fact, a methodology to explain the relationship between each input variable and output variable before model development; however, it is slightly insufficient to comprehensively explain the influence of a specific input variable on the prediction of the ML model [85,86]. Permutation feature importance (PFI) is used as a method to comprehensively determine the importance of variables in a model. To determine the strength of the relationship between the feature and the target value, the error increase in the model prediction is measured after the features are randomly removed. If the model error increases when randomly removing one feature, it is a "significant" feature because it indicates that the model depends on that feature when making predictions. Conversely, if there is no difference in error, the feature is said to be "non-significant" [87].

Figure 16 shows that S_{max} and f are very important input variables in the gradient boosting model. It also shows that f'_c and R are the next most important features, and shape and h/w are features with very weak influence on the gradient boosting model.

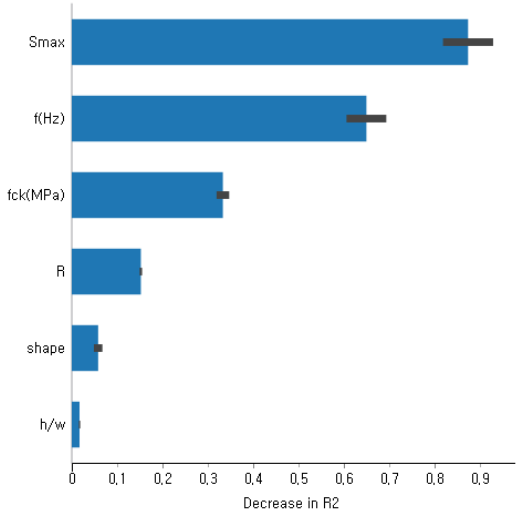


Figure 16. Permutation feature importance of the gradient boosting model.

7. Conclusions

The goal of this work was to show how ML models can be used to forecast the fatigue life (N) of plain concrete under uniaxial compression. The fatigue life was forecasted using random forest, neural network, gradient boosting, and AdaBoost models. The models were developed sequentially using three data sets. The first was developed with original data, the second was developed with outliers removed, and the last model was developed with the average value of data with different outputs in the same input. For training and testing of the models, a ratio of training and testing was used as 90:10 in order to secure as much of the training data as possible. From this, we were able to make the following conclusions.

1. Three data files were generated from the original data, which were used in the work of Zhang et al. [4]. These files were used to develop the final ML model and were as follows: (a) grouped data with the same input variable value and different output variable log$N$ value, (b) data excluding outliers selected by three or more outlier detection methods; (c) average data excluding outliers, created by averaging the grouped data after excluding outliers from the grouped data.
2. From the Pearson and Spearman correlation analysis, it was observed that the maximum stress level S_{max} had a solid negative relationship with log$N$, and the loading frequency f had a solid positive relationship with log$N$. Simultaneously, the height to width ratio (h/w) and shape of the tested specimens had weak relationships with log$N$.
3. Excluding the sustained strength of the concrete variable, originally treated as the seventh input variable in the work of Zhang et al. [4], resulted in improving the MSE

and determination coefficient R^2 values. Moreover, the gradient boosting model with the value of the result of minimum error and a high R^2 value at 0.753 was an indication of high accuracy in predicting outcomes.

4. Further analysis using the data excluding outliers caused the determination coefficient R^2 value to increase to 0.803. Moreover, the average data excluding outliers provided the best correlation with the R^2 value at 0.915.
5. Finally, to determine the strength of the relationship between the feature and the target value, a permutation feature importance (PFI) analysis was carried out for the gradient boosting model. The analysis results confirmed that the maximum stress level S_{max} and loading frequency f are critical input variables, followed by compressive strength f'_c and the maximum to minimum stress ratio R. Shape and h/w are features that have only a minor influence on the model.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/app12199766/s1, The following spreadsheet contains the supplementary data to this article: download spreadsheet (xxKB).

Author Contributions: J.S.: Methodology, Software, Validation and Writing; S.Y.: Conceptualization, Data curation and Writing. All authors have read and agreed to the published version of the manuscript.

Funding: This study was conducted under the research project (22POQW-C152690-04), funded by the Ministry of Land, Infrastructure and Transport (MOLIT) and the Korea Agency for Infrastructure Technology Advancement (KAIA). The authors would like to thank the members of the research team, MOLIT and KAIA, for their guidance and support throughout the project.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The full-data spreadsheet is available in the Supplementary Materials.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mindess, S.; Young, J.; Darwin, D. *Concrete*, 2nd ed.; Prentice Hall: Hoboken, NJ, USA, 2003.
2. Shah, S.; Chandra, S. Fracture of concrete subjected to cyclic and sustained loading. *ACI J.* **1970**, *67*, 816–827.
3. Shah, S. *Fatigue of Concrete*; ACI, SP-75; American Concrete Institute: Detroit, Michigan, USA, 1981.
4. Zhang, W.; Lee, D.; Lee, J.; Lee, C. Residual strength of concrete subjected to fatigue based on machine learning technique. *Struct. Concr.* **2021**, *23*, 2274–2287. [CrossRef]
5. Medeiros, A.; Zhang, X.; Ruiz, G.; Yu, R.; Velasco, M. Effect of the loading frequency on the compressive fatigue behavior of plain and fiber reinforced concrete. *Int. J. Fatigue* **2015**, *70*, 342–350. [CrossRef]
6. Isojeh, B.; El-Zeghayar, M.; Vecchio, F. Concrete damage under fatigue loading in uniaxial compression. *ACI Mater. J.* **2017**, *114*, 225–235. [CrossRef]
7. Dong, S.; Wang, Y.; Ashour, A.; Han, B.; Ou, J. Uniaxial compressive fatigue behavior of ultra-high performance concrete reinforced with super-fine stainless wires. *Int. J. Fatigue* **2021**, *142*, 105959. [CrossRef]
8. Lv, J.; Zhou, T.; Du, Q.; Li, K. Experimental and analytical study on uniaxial compressive fatigue behavior of self-compacting rubber lightweight aggregate concrete. *Constr. Build Mater.* **2020**, *237*, 117623. [CrossRef]
9. Yin, L. Fatigue Damage of Concrete under Uniaxial Compression. In Proceedings of the 7th International Conference on Energy and Environmental Protection (ICEEP 2018), Shenzhen, China, 4–15 July 2018; Atlantis Press: Hohhot, China, 2018; pp. 933–936.
10. Zhao, Z.; Zhang, L.; Li, Z. Model of strength degradation and the predictor method of life period for concrete under low-cycle fatigue loading. *Mech. Eng.* **2011**, *33*, 35–38.
11. Do, M.; Challal, O.; AItcin, P. Fatigue behavior of high-performance concrete. *J. Mater. Civ. Eng.* **1993**, *5*, 96–111. [CrossRef]
12. Dyduch, K.; Szerszen, M.; Destrebecq, J. Experimental investigation of the fatigue strength of plain concrete under high compressive loading. *Mater. Struct.* **1994**, *27*, 505–509. [CrossRef]
13. Yu, Z.; An, M.; Yan, G. Experimental research on the fatigue performance of reactive powder concrete. *China Railw. Sci.* **2008**, *29*, 35–40.
14. Ou, J.; Lin, Y. Experimental study on performance degradation of plain concrete due to high-cycle fatigue damage. *China Civil Eng. J.* **1999**, *32*, 15–22.
15. Yan, C.; Shi, Y.; Ding., C. Fatigue test of recycled concrete under cyclic loading. *Cem. Eng.* **2018**, *6*, 10–13.

16. Liu, K.; Luo, R.; Zheng, P.; Tong, K. High frequency fatigue accelerated life test of concrete. *J. Shang. Univ. (Nat. Sci.)* **2009**, *15*, 205–210.
17. Xiao, J.; Li, H. Investigation on the fatigue behavior of recycled aggregate concrete under uniaxial compression. *China Civil Eng. J.* **2013**, *46*, 62–69.
18. Kim, J.; Kim, Y. Experimental study of the fatigue behavior of high strength concrete. *Cem. Concr. Compos.* **1996**, *26*, 1513–1523. [CrossRef]
19. Matsushita, H.; Tokumitsu, Y. A study on compressive fatigue strength of concrete considering survival probability. *Jpn. Soc. Civ. Eng.* **1979**, *284*, 127–138. [CrossRef]
20. Mu, B.; Subramaniam, V.; Shah, S. Failure mechanism of concrete under fatigue compressive load. *J. Mater. Civ. Eng.* **2004**, *1561*, 566–572. [CrossRef]
21. Mun, J.; Yang, K.; Kim, S. Tests on the compressive fatigue performance of various concretes. *J. Mater. Civ. Eng.* **2016**, *28*, 04016099. [CrossRef]
22. Vicente, M.; Gonzalez, D.; Mínguez, J.; Tarifa, M.; Ruiz, G.; Hindi, R. Influence of the pore morphology of high strength concrete on its fatigue life. *Int. J. Fatigue* **2018**, *112*, 106–116. [CrossRef]
23. Oneschkow, N. Fatigue behaviour of high-strength concrete with respect to strain and stiffness. *Int. J. Fatigue* **2016**, *87*, 38–49. [CrossRef]
24. Ortega, J.; Ruiz, G.; Yu, R.; Afanador-García, N.; Tarifa, M.; Poveda, E.; Zhang, X.; Evangelista, F., Jr. Number of tests and corresponding error in concrete fatigue. *Int. J. Fatigue* **2018**, *116*, 210–219. [CrossRef]
25. Raju, N. Prediction of the fatigue life of plain concrete in compression. *Build. Sci.* **1969**, *4*, 99–102. [CrossRef]
26. Zhao, Z.; Zhang, L.; Li, Z. Research on fatigue residual strain of hydraulic concrete based on compression strength extrapolation. *Mech. Eng.* **2011**, *33*, 29–32.
27. Saucedo, L.; Yu, R.; Medeiros, A.; Zhang, X.; Ruiz, G. A probabilistic fatigue model based on the initial distribution to consider frequency effect in plain and fiber reinforced concrete. *Int. J. Fatigue* **2013**, *48*, 308–318. [CrossRef]
28. Liu, Z.; Wang, W.; Wen, H.; Gan, H.; Shi, Z.; Zhang, K. Study on compression fatigue of single axial of total light concrete after different temperature. *Concrete* **2019**, *5*, 48–53.
29. Tepfers, R.; Hedberg, B.; Szczekocki, G. Absorption of energy in fatigue loading of plain concrete. *Mater. Constr.* **1984**, *17*, 59–64. [CrossRef]
30. Wu, B.; Jin, H. Compressive fatigue behavior of compound concrete containing demolished concrete lumps. *Constr. Build. Mater.* **2019**, *210*, 140–156. [CrossRef]
31. Chen, Y.; Ni, J.; Zheng, P.; Azzam, R.; Zhou, Y.; Shao, W. Experimental research on the behaviour of high frequency fatigue in concrete. *Eng. Failure Anal.* **2011**, *18*, 1848–1857. [CrossRef]
32. You, F.; Luo, S.; Zheng, J. Experimental study on residual compressive strength of recycled aggregate concrete under fatigue loading. *Front. Mater.* **2022**, *9*, 817103. [CrossRef]
33. Wang, M.; Zhao, G.; Song, Y. Fatigue of plain concrete under compression. *China Civil. Eng. J.* **1991**, *24*, 39–47.
34. Fantilli, A.; Józwiak-Niedzwiedzka, D. Supplementary cementitious materials in concrete, Part I. *Materials* **2021**, *14*, 2291. [CrossRef] [PubMed]
35. Jalulski, R.; Józwiak-Niedzwiedzka, D.; Yakymechko. Calcined clay as supplementary cementitious materials. *Materials* **2020**, *13*, 13184204.
36. Motlagh, S.; Naghizadehrokni, M. An extended multi-model regression approach for compressive strength prediction and optimization of a concrete mixture. *Constr. Build. Mater.* **2022**, *327*, 126828. [CrossRef]
37. Asteris, P.; Mokos, V. Concrete compressive strength using artificial neural networks, V.G. *Neural Comput. Appl.* **2020**, *32*, 11807–11826. [CrossRef]
38. Golafshani, E.; Behnood, A.; Arashpourc, M. Predicting the compressive strength of normal and High-Performance Concretes using ANN and ANFIS hybridized with Grey Wolf Optimizer. *Constr. Build. Mater.* **2020**, *232*, 117266. [CrossRef]
39. Zhang, J.; Huang, Y.; Wang, Y.; Ma, G. Multi-objective optimization of concrete mixture proportions using machine learning and metaheuristic algorithms. *Constr. Build. Mater.* **2020**, *253*, 119208. [CrossRef]
40. Ekanayake, I.; Meddage, D.; Rathnayake, U. A novel approach to explain the black-box nature of machine learning in compressive strength predictions of concrete using Shapley additive explanations (SHAP). *Case Stud. Constr. Mater.* **2022**, *16*, e01059. [CrossRef]
41. Feng, D.; Liu, Z.; Wang, X.; Chen, Y.; Chang, J.; Wei, D.; Jiang, Z. Machine learning-based compressive strength prediction for concrete: An adaptive boosting approach. *Constr. Build. Mater.* **2020**, *230*, 117000. [CrossRef]
42. Abuodeh, O.; Abdalla, J.; Hawileh, R. Assessment of compressive strength of Ultra-high Performance Concrete using deep machine learning techniques. *Appl. Soft Comput.* **2020**, *95*, 106552. [CrossRef]
43. Dao, D.; Adeli, H.; Ly, H.; Le, L.; Le, V.; Le, T.; Pham, B. A sensitivity and robustness analysis of GPR and ANN for high-performance concrete compressive strength prediction using a monte carlo simulation. *Sustainability* **2020**, *12*, 830. [CrossRef]
44. Han, Q.; Gui, C.; Xu, J.; Lacidogna, G. A generalized method to predict the compressive strength of high-performance concrete by improved Random Forest algorithm. *Constr. Build. Mater.* **2019**, *226*, 734–742. [CrossRef]
45. Nafees, A.; Javed, M.F.; Khan, S.; Nazir, K.; Farooq, F.; Aslam, F.; Musarat, M.; Vatin, N. Predictive Modeling of Mechanical Properties of Silica Fume-Based Green Concrete Using Artificial Intelligence Approaches: MLPNN, ANFIS, and GEP. *Materials* **2021**, *14*, 7531. [CrossRef]

46. Song, H.; Ahmad, A.; Ostrowski, K.A.; Dudek, M. Analyzing the compressive strength of ceramic waste-based concrete using experiment and artificial neural network (ANN) approach. *Materials* **2021**, *14*, 4518. [CrossRef] [PubMed]
47. Chen, N.; Zhao, S.; Gao, Z.; Wang, D.; Liu, P.; Oeser, M.; Hou, Y.; Wang, L. Virtual mix design: Prediction of compressive strength of concrete with industrial wastes using deep data augmentation. *Constr. Build. Mater.* **2022**, *323*, 126580. [CrossRef]
48. Ghanemi, A.; Tarighat, A. Use of Different Hyperparameter Optimization Algorithms in ANN for Predicting the Compressive Strength of Concrete Containing Calcined Clay. *Pract. Period. Struct. Des. Constr. ASCE* **2022**, *27*, 04022002. [CrossRef]
49. Kandiri, A.; Golafshani, E.; Behnood, A. Estimation of the compressive strength of concretes containing ground granulated blast furnace slag using hybridized multi-objective ANN and salp swarm algorithm. *Constr. Build. Mater.* **2020**, *248*, 118676. [CrossRef]
50. Javed, M.; Amin, M.; Shah, M.; Khan, K.; Iftikhar, b.; Farooq, F.; Aslam, F.; Alyousef, R.; Alabduljabbar, H. Applications of gene expression programming and regression techniques for estimating compressive strength of bagasse ash based concrete. *Crystals* **2020**, *10*, 737. [CrossRef]
51. Ahmad, A.; Farooq, F.; Niewiadomski, P.; Ostrowski, K.; Akbar, A.; Aslam, F.; Alyousef, R. Prediction of compressive strength of fly ash based concrete using individual and ensemble algorithm. *Materials* **2021**, *14*, 794. [CrossRef]
52. Zeng, Z.; Zhu, Z.; Yao, W.; Wang, Z.; Wang, C.; Wei, Y.; Wei, Z.; Guan, X. Accurate prediction of concrete compressive strength based on explainable features using deep learning. *Constr. Build. Mater.* **2022**, *329*, 127082. [CrossRef]
53. Ray, S.; Rahman, M.M.; Haque, M.; Hasan, M.W.; Alam, M.M. Performance evaluation of SVM and GBM in predicting compressive and splitting tensile strength of concrete prepared with ceramic waste and nylon fiber. *J. King Saud Univ.—Eng. Sci* **2021**, 1–9. [CrossRef]
54. Moradi, M.; Khaleghi, M.; Salimi, J.; Farhangi, V.; Ramezanianpour, A. Predicting the compressive strength of concrete containing metakaolin with different properties using ANN. *Measurement* **2021**, *183*, 109790. [CrossRef]
55. Kandiri, A.; Sartipi, F.; Kioumarsi, M. Predicting compressive strength of concrete containing recycled aggregate using modified ANN with different optimization algorithms. *Appl. Sci.* **2021**, *11*, 485. [CrossRef]
56. Deng, F.; He, Y.; Zhou, S.; Yu, Y.; Cheng, H.; Wu, X. Compressive strength prediction of recycled concrete based on deep learning. *Constr. Build. Mater.* **2018**, *175*, 562–569. [CrossRef]
57. Shang, M.; Li, H.; Ahmad, A.; Ahmad, W.; Ostrowski, K.; Aslam, F.; Majka, T. Predicting the Mechanical Properties of RCA-Based Concrete Using Supervised Machine Learning Algorithms. *Materials* **2022**, *15*, 647. [CrossRef] [PubMed]
58. Huynh, A.; Nguyen, Q.; Xuan, Q.; Magee, B.; Chung, T.; Tran, K.; Nguyen, K. A Machine Learning-Assisted Numerical Predictor for Compressive Strength of Geopolymer Concrete Based on Experimental Data and Sensitivity Analysis. *Appl. Sci.* **2020**, *10*, 7726. [CrossRef]
59. Ahmad, A.; Ahmad, W.; Chaiyasarn, K.; Ostrowski, K.; Aslam, F.; Zajdel, P.; Joyklad, P. Prediction of geopolymer concrete compressive strength using novel machine learning algorithms. *Polymers* **2021**, *13*, 3389. [CrossRef] [PubMed]
60. Zhu, Y.; Ahmad, A.; Ahmad, W.; Vatin, N.; Mohamed, A.; Fathi, D. Predicting the splitting tensile strength of recycled aggregate concrete using individual and ensemble machine learning approaches. *Crystals* **2022**, *12*, 569. [CrossRef]
61. Han, T.; Siddique, A.; Khayat, K.; Huang, J.; Kumar, A. An ensemble machine learning approach for prediction and optimization of modulus of elasticity of recycled aggregate concrete. *Constr. Build. Mater.* **2020**, *244*, 118271. [CrossRef]
62. Ziolkowski, P.; Niedostatkiewicz, M. Machine learning techniques in concrete mix design. *Materials* **2019**, *12*, 1256. [CrossRef]
63. Abambres, M.; Lantsought, E. ANN-based fatigue strength of concrete under compression. *Materials* **2019**, *12*, 3787. [CrossRef]
64. Grubbs, F. Procedures for detecting outlying observations in samples. *Technometrics* **1969**, *11*, 1–21. [CrossRef]
65. Maddala, G. *Introduction to Econometrics*, 2nd ed.; MacMillan: New York, NY, USA, 1992.
66. Chauvenet, W. *A Manual of Spherical and Practical Astronomy V. II*, 5th ed.; Dover: New York, NY, USA, 1960; pp. 474–566.
67. Peirce, B. Criterion for the Rejection of Doubtful Observations. *Astron. J.* **1852**, *2*, 161–163. [CrossRef]
68. Dixon, W. Analysis of extreme values. *Annals Math. Statis.* **1950**, *21*, 488–506. [CrossRef]
69. Rosner, B. Percentage Points for a Generalized ESD Many-Outlier Procedure. *Technometrics* **1983**, *25*, 165–172. [CrossRef]
70. Thompson, R. A Note on Restricted Maximum Likelihood Estimation with an Alternative Outlier Model. *J. Royal Statis. Sic. Series B* **1985**, *47*, 53–55. [CrossRef]
71. Wheeler, D. Some Outlier Tests, Part 1: Comparisons and Recommendations. *Quality Digest.* **2020**, *378*, 1–10.
72. Wheeler, D. Some Outlier Tests, Part 2: Tests with fixed overall alpha levels. *Quality Digest.* **2021**, *379*, 1–11.
73. Goki, S. *Deep Running Starting from the Bottom*; Hanvit Media: Seoul, Korea, 2019.
74. Grus, J. *Data Science from Scratch: First Principles with Python*; O'Reilly Media: Sebastopol, CA, USA, 2015.
75. Aurélien, G. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*; Hanvit Media: Seoul, Korea, 2018.
76. Islam, S.; Amin, S. Prediction of probable backorder scenarios in the supply chain using Distributed Random Forest and Gradient Boosting Machine learning techniques. *J. Big Data* **2020**, *7*, 1–22. [CrossRef]
77. Ahmad, W.; Ahmad, A.; Ostrowski, K.; Aslam, F.; Joyklad, P.; Zajdel, P. Application of Advanced Machine Learning Approaches to Predict the Compressive Strength of Concrete Containing Supplementary Cementitious Materials. *Materials* **2021**, *14*, 5762. [CrossRef]
78. Ahmad, M.; Kamiński, P.; Olczak, P.; Alam, M.; Iqbal, M. Development of prediction models for shear strength of rockfill material using machine learning techniques. *Appl. Sci.* **2021**, *11*, 6167. [CrossRef]

79. Demsar, J.; Curk, T.; Erjavec, A.; Gorup, C.; Hocevar, T.; Milutinovic, M.; Mozina, M.; Polajnar, M.; Toplak, M.; Staric, A.; et al. Orange: Data Mining Toolbox in Python. *J. Mach. Learn. Res.* **2013**, *14*, 2349–2353.
80. Malinin, A.; Prokhorenkova, L.; Ustimenko, A. Uncertainty in Gradient Boosting via Ensembles, International Conference on Learning Representations. In Proceedings of the Ninth International Conference on Learning Representations, Vienna, Austria, 4 May 2021; pp. 1–17.
81. Boehmke, B.; Greenwell, B. *Hands-On Machine Learning with R*; Chapman and Hall/CRC: London, UK, 2019.
82. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning*, 2nd ed.; Springer: New York, NY, USA, 2009; pp. 337–384.
83. Piryonesi, S.; El-Diraby, T. Data Analytics in Asset Management: Cost-Effective Prediction of the Pavement Condition Index. *J. Infrastruct. Sys.* **2020**, *26*, 04019036. [CrossRef]
84. Benemaran, R.; Esmaeili-Falak, M.; Javadi, A. Predicting resilient modulus of flexible pavement foundation using extreme gradient boosting based optimised models. *I. J. Pavement Eng.* **2022**, 2095385. [CrossRef]
85. A Comparison of the Pearson and Spearman Correlation Methods. Available online: https://support.minitab.com/en-us/minitab/18/help-and-how-to/statistics/basic-statistics/supporting-topics/correlation-and-covariance/a-comparison-of-the-pearson-and-spearman-correlation-methods/ (accessed on 10 June 2020).
86. Clearly explained: Pearson V/S Spearman Correlation Coefficient. Available online: https://towardsdatascience.com/clearly-explained-pearson-v-s-spearman-correlation-coefficient-ada2f473b8 (accessed on 26 June 2020).
87. Permutation Feature Importance. Available online: https://scikit-learn.org/stable/modules/permutation_importance.html (accessed on 10 June 2020).