

Article

Compressive Strength Prediction of Rubber Concrete Based on Artificial Neural Network Model with Hybrid Particle Swarm Optimization Algorithm

Xiao-Yu Huang ¹ , Ke-Yang Wu ², Shuai Wang ², Tong Lu ², Ying-Fa Lu ¹, Wei-Chao Deng ¹ and Hou-Min Li ^{1,*}

¹ School of Civil Engineering, Architecture, and Environment, Hubei University of Technology, Wuhan 430068, China; 102000669@hbut.edu.cn (X.-Y.H.); luf77@126.com (Y.-F.L.); dengweichao@petalmail.com (W.-C.D.)

² Wuhan Construction Engineering Company Limited, Wuhan 430056, China; wukeyang0623@163.com (K.-Y.W.); wangshuai@wceg.com.cn (S.W.); lutong1@wceg.com.cn (T.L.)

* Correspondence: lihoumin2000@163.com

Abstract: Conventional neural networks tend to fall into local extremum on large datasets, while the research on the strength of rubber concrete using intelligent algorithms to optimize artificial neural networks is limited. Therefore, to improve the prediction accuracy of rubber concrete strength, an artificial neural network model with hybrid algorithm optimization was developed in this study. The main strategy is to mix the simulated annealing (SA) algorithm with the particle swarm optimization (PSO) algorithm, using the SA algorithm to compensate for the weak global search capability of the PSO algorithm at a later stage while changing the inertia factor of the PSO algorithm to an adaptive state. For this purpose, data were first collected from the published literature to create a database. Next, ANN and PSO-ANN models are also built for comparison while four evaluation metrics, MSE, RMSE, MAE, and R^2 , were used to assess the model performance. Finally, compared with empirical formulations and other neural network models, the result shows that the proposed optimized artificial neural network model successfully improves the accuracy of predicting the strength of rubber concrete. This provides a new option for predicting the strength of rubber concrete.

Keywords: compressive strength; ANN; concrete; prediction; algorithm



Citation: Huang, X.-Y.; Wu, K.-Y.; Wang, S.; Lu, T.; Lu, Y.-F.; Deng, W.-C.; Li, H.-M. Compressive Strength Prediction of Rubber Concrete Based on Artificial Neural Network Model with Hybrid Particle Swarm Optimization Algorithm. *Materials* **2022**, *15*, 3934. <https://doi.org/10.3390/ma15113934>

Academic Editor: Marijana Hadzima-Nyarko

Received: 29 March 2022

Accepted: 28 May 2022

Published: 31 May 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 is one of the most widely used building materials in the construction industry [1]. The global construction industry is challenged by the high cost and environmental pollution caused by concrete; meanwhile, there are increasing demands on the strength and durability of concrete. Thus, the traditional concrete is being replaced by concrete with better performance. Meanwhile, the rubber tires have vast output from other waste materials due to their excellent strength, low cost, and easy availability. In addition, adding rubber scraps to the cementitious composites can reduce carbon dioxide emissions which is environmentally friendly [2,3]. Therefore, the addition of rubber to concrete (rubberized concrete) becomes a feasible plan [4–10]. However, the compressive strength of rubberized concrete is lower than that of plain concrete [11]. This decrease is caused by variables including the ratio of aggregate (reduced 85% and 65% of compressive strength, when the coarse aggregate and fine aggregate are fully replaced by rubber, respectively) [8,11,12], size, and shape of rubber [12,13].

Machine learning, which can learn from given data and make accurate predictions through complex systems, includes the support vector machine model (SVM), random forest model (RF), fuzzy logic (FL), artificial neural network (ANN) model, etc. [14]. Machine learning techniques have been used extensively in the study of concrete, for example, Nasrollahzadeh, K and Nouhi, E [15] developed a fuzzy inference system (FIS) model that

was used to investigate the compressive strength and strain capacity of axially loaded fiber-reinforced polymer concrete columns, and it was successfully demonstrated to be superior to existing models. Mahmood Ahmad et al. [16]. used three machine learning models, AdaBoost, RF, and decision tree (DT), to predict the compressive strength of concrete at high temperatures and demonstrated the applicability of supervised learning methods to study the compressive strength of concrete at high temperatures. Artificial neural network models are the most widely used of the many machine learning models due to their powerful nonlinear ability to learn from numerous complex situations. Current research on the strength of rubber concrete using artificial neural networks is proving successful, for example, Abdollahzadeh et al. [17] used 20 data to build a neural network model to predict the strength of rubber concrete, with the structure of the network set to 3–1–1, and the final accuracy of the model was calculated to be 96%. El-Khoja A. et al. [9] built a neural network model using 287 data with input variables set to five. The accuracy of the model was calculated to be 91%, while the optimal structure was determined to be 5–10–1. Nyarko et al. [7] used rubber concrete data to build a deep neural network model with the structure 9–3–2, and demonstrated that deep neural networks could be an alternative option for predicting the strength of rubber concrete. From the literature, it is known that there are fewer studies using intelligent algorithms for optimization. Intelligent algorithms achieve optimal results when solving complex nonlinear problems. Conventional ANN models tend to fall into local extremum with large datasets, while the strength of rubber concrete is influenced by a variety of nonlinear factors, so the use of intelligent algorithms is appropriate. Particle swarm optimization (PSO) algorithms were widely used in model optimization due to their fast convergence of iterations and simplicity of operation [18–20]. However, due to the fixed value of the inertia factor of the PSO algorithm, this makes the search space of the particles small and causes the algorithm to easily fall into local extremes late in the iteration. The simulated annealing (SA) algorithm was a global search algorithm with a good ability to search globally and accept poorer values, so it could jump out of local extremum to obtain the global best value [21]. Hybrid algorithms have not been used in current research on the strength of rubber concrete using neural networks. Therefore, an adaptive simulated annealing particle swarm optimization (ASAPSO) algorithm is developed in this study.

In summary, this study developed an ANN model based on ASAPSO algorithm optimized to predict the strength of rubber concrete. This model could achieve higher accuracy than conventional ANN model and ANN model optimized by a single algorithm under a large dataset. Thus, a database was first created and analyzed to filter out 11 variables that sensitivity factor analysis was performed on. Then, three different models, ANN models, ANN models optimized by PSO (PSO-ANN) algorithm, and the ANN models optimized by ASAPSO (ASAPSO-ANN) algorithm, were built. The database data were fed into the three models for training and testing. Finally, the performances of the three models were evaluated statistically, and their computational results were compared and analyzed.

2. Database Description and Analysis of Variables

A reliable database is essential for the successful training of machine learning models. Without reliable data, the training results do not reflect the actual situation and eventually lead to model training failure. This study, therefore, proposes the following treatments in the process of building a database of rubber concrete:

Firstly, an adequate database does not only require a large amount of data, but also a comprehensive range of input and output variables. Therefore, data need to be collected from previously published literature to build the database.

Secondly, concrete is a composite material. As the material has a significant impact on strength, different materials need to be distinguished during data collection to make the network more capable of learning. For example, the compressive strength values and mechanisms of action of concrete mixtures formed with ordinary silicate cement are different from those of concrete mixtures formed with cement replacement materials. It is

also necessary to divide the dimensions of the rubber material; this is because the rubber material replaces different objects.

Thirdly, the size and shape of the concrete may vary. Therefore, the specimen size and shape are considered in database creation. There is a need to standardize the units in which the data are collected. Therefore, except for the specimen dimensions in mm and the compressive strength in MP, the units for all data in this study are kg/m³. In addition, the rubber concrete samples in this study were all on a 28-day curing cycle.

In summary, a database of 307 sets of experimental data has been developed in this paper, where the data were derived from the published literature [6,8,10,22–52]. The database was divided into two parts, of which 277 groups were used for training and the remaining 30 groups were used for testing. Eleven input variables were used in this paper; these include cement content, water content, water–cement ratio, cement replacement material content, coarse aggregate content, fine aggregate content, crumb rubber content, flake rubber content, admixture addition, and specimen shape. The output variable was the compressive strength of the rubber concrete at 28 days. The statistical analysis of the input and output variables of the database is shown in Table 1. It is important to note that the interdependency of the model is an important parameter as it can lead to poor performance of the model [53]. The various parameters used as inputs can be interdependent, and such problems are known as “multicollinearity problems” [54]. It has been suggested in the literature that to develop an accurate model, the correlation between the two input parameters should be less than 0.8 [55]. A heat map of the correlation coefficients of the input and output variables in this study is shown in Figure 1. Table 2 shows that the correlation coefficients between the parameters are all less than 0.8, thus reducing the effect of multiple collinearities.

Table 1. Statistical analysis of input and output variables.

	Max	Min	Average	Median	Standard Deviation	Skewness
Cement (kg/m ³)	629.27	18.80	406.81	400.00	75.22	−0.26
Cement replacement (kg/m ³)	180.00	0.00	12.19	0.00	31.84	3.38
Superplasticizer (kg/m ³)	13.50	0.00	3.23	2.08	3.96	1.45
Water (kg/m ³)	312.00	9.20	192.83	180.00	38.43	−0.24
Water/cement (kg/m ³)	0.83	0.27	0.48	0.45	0.11	0.67
Fine aggregate (kg/m ³)	1364.00	0.00	610.11	631.37	219.09	−0.63
Coarse aggregate (kg/m ³)	1434.60	0.00	911.54	949.00	230.99	−0.44
Crumb rubber [0–5] (kg/m ³)	1160.00	0.00	59.06	36.00	102.16	6.56
Chipped rubber [5–10] (kg/m ³)	227.30	0.00	14.46	0.00	40.29	3.55
Chipped rubber [10–30] (kg/m ³)	630.00	0.00	16.40	0.00	59.47	6.52
fc specimen type	3.00	0.00	0.89	1.00	0.73	0.63
fc (MP)	78.30	2.50	29.41	27.05	15.24	0.65

Table 2. Parameter setting of ANN model.

Parameter	Setting
Input layer node	11
Output layer node	1
Hidden layer node	6
Activation function	Tansig, purelin
Training function	trainlm
Epochs	50
Learning rate	0.01
Performance goal	1.00×10^{-5}
Epochs between display	25
Momentum factor	0.01
Minimum performance gradient	1.00×10^{-6}
Maximum validation failure	6

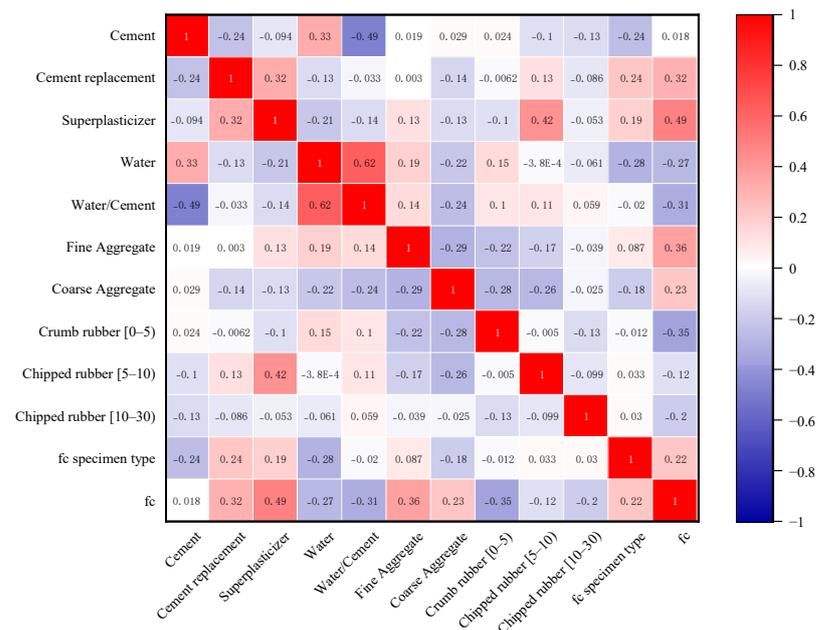


Figure 1. Heat map of the correlation for each variable.

3. Sensitivity Factor Analysis of Input Variables

Sensitivity factor analysis (SA) is an effective method for measuring the influence of model input parameters on output parameters. The sensitivity factor analysis allows the input parameters to be filtered, thus reducing the complexity of the model, and saving time in model calculations. To achieve this, the cosine amplitude method (CAM) can be adopted [56]. The expression for this method is as follows:

$$R_{ij} = \frac{\sum_{k=1}^n x_{ik}x_{jk}}{\sqrt{\sum_{k=1}^n x_{ik}^2 \sum_{k=1}^n x_{jk}^2}} \tag{1}$$

where x_i denotes the input parameter; x_j denotes the output parameter; n indicates the number of data; R_{ij} means the strength of the relationship. The values of R_{ij} between the strength of the rubber concrete and the input parameters are shown in Figure 2. As can be seen from the graph, the most significant influence on the strength of the rubber concrete is the fine aggregate, followed by the coarse aggregate, while the smallest impact is on the largest particle size of the rubber sheet.

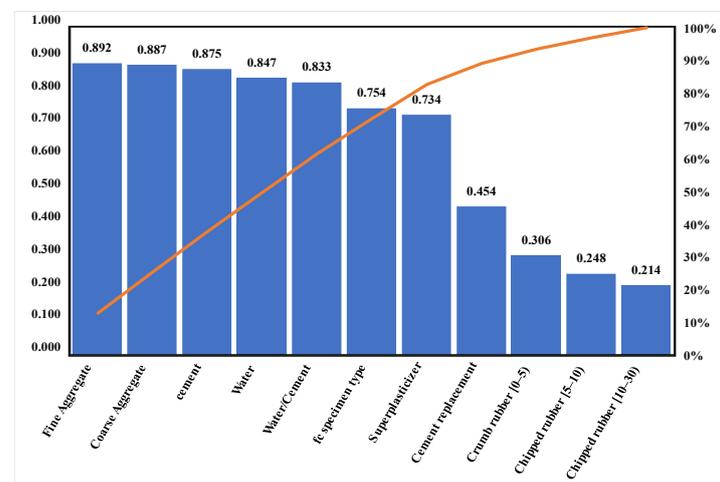


Figure 2. Analysis of sensitivity factors.

4. Methods

Artificial neural networks are used in various fields because of their powerful nonlinear capabilities. The simulated annealing (SA) algorithm is the same as the particle swarm optimization (PSO) algorithm—both are intelligent algorithms, but the SA algorithm is physically inspired and has better global search capabilities. The particle swarm algorithm is a swarm heuristic algorithm which is computationally simple and easy to operate but prone to local extremum. The ASAPSO algorithm and ANN model used are described in detail in this section.

4.1. Artificial Neural Network

The ANN is one of the most used machine learning techniques for predicting the compressive strength of concrete. ANN can be thought of as a data processing operation or as a black box that produces the appropriate output based on the input data [57]. A backpropagation network is the most commonly used network structure and it adjusts weights and biases by backpropagation of errors, eventually reducing the errors to an acceptable state [58]. A backpropagation neural network is composed of an input layer, a hidden layer, and an output layer. In addition to the input layer, the neurons of the hidden layer and output layer are composed of weights, biases, and activation functions. The structure of the backpropagation neural network is shown in Figure 3.

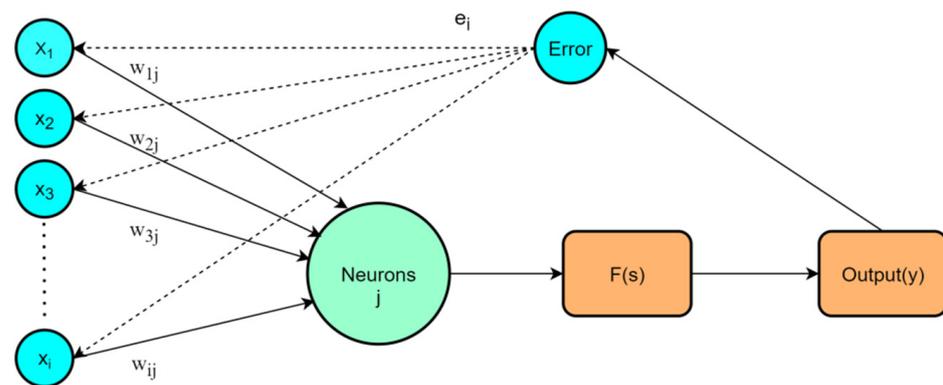


Figure 3. Backpropagation neural network structure.

In Figure 3, x_i represents input variables, w_{ij} represents the connection weight. The expressions of s and neural network output y are shown in Equations (2) and (3) [59].

$$d = \sum_{i=1}^n w_{ij}x_i + b \quad (2)$$

$$Y = F(d) \quad (3)$$

where b is the bias, and d is the sum of weights and biases. The role of the bias is to increase or decrease the influence of the activation function. F denotes the activation function, which allows the neural network to approximate any nonlinear function and thus can be applied to a wide range of nonlinear models. There are various activation functions for neural networks, among which the commonly used transfer functions are sigmoid, softplus, tanh, and ReLU. An image of a commonly used activation function is shown in Figure 4.

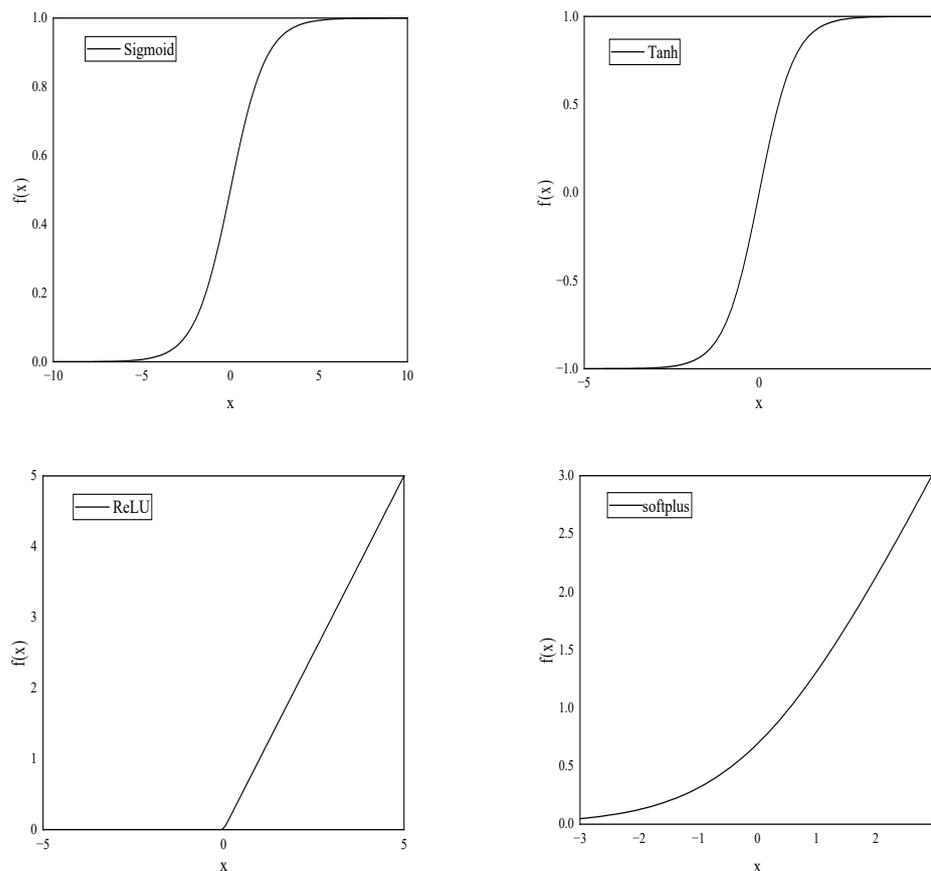


Figure 4. Four activation functions.

Since the backpropagation network can constantly adjust the weights and biases based on the results, the weights and biases are updated by the following formula [59]:

$$w(i + 1) = w(i) - \alpha \frac{\partial E(i)}{\partial w(i)} \tag{4}$$

$$b(i + 1) = b(i) - \alpha \frac{\partial E(i)}{\partial b(i)} \tag{5}$$

where α indicates the learning rate; $w(i)$ and $b(i)$ denote the vector of connection weights and bias vectors between the layers of the i th iteration, respectively; $\frac{\partial E(i)}{\partial w(i)}$ and $\frac{\partial E(i)}{\partial b(i)}$ denote the error gradient of the output error of the i th iteration for each weight and bias, respectively; $E(i)$ is the error in the output of the i th iteration of the network, with the following functional expression [59]:

$$E(k) = \sqrt{\frac{1}{n} \sum_{i=1}^n |e_i|^2} \tag{6}$$

where e_i indicates the error between the actual value and the predicted value; n indicates the total amount of data.

The number of neurons contained in the hidden layer (h) is an important parameter of artificial neural networks. However, there is no specific calculation method for the number of neurons contained in the hidden layer. Usually, the approximate range is determined by an empirical formula, and then the optimal number of neurons is determined according to the trial-and-error method. The expression is given in Equation (7).

$$h = \sqrt{m + n} + a \tag{7}$$

where m is the number of nodes in the input layer; n is the number of nodes in the output layer, $a \in (1, 10)$.

To reduce the prediction error and improve the training efficiency, it is necessary to normalize the training data and the verification data. The specific expression for the normalization function is as follows:

$$y = \frac{(y_{max} - y_{min})(x - x_{min})}{x_{max} - x_{min}} + y_{min} \quad (8)$$

where y is the normalized value; y_{max} and y_{min} are the maximum and minimum values of the normalized range, respectively, usually taking the values [0,1]; x is the value of the input variable; x_{max} and x_{min} are the maximum and minimum values of the variables, respectively.

There are several classic algorithms in the training phase of ANN, namely, adaptive learning rate gradient descent algorithm, gradient descent algorithm, momentum gradient descent algorithm, conjugate gradient algorithm, and Levenberg–Marquardt (LM) algorithm [60]. The choice of the algorithm needs to depend on the situation. The LM algorithm is a classic backpropagation algorithm with fast convergence speed and high precision. The LM algorithm was chosen for this study.

4.2. Particle Swarm Optimization Algorithm

Similar to the genetic algorithm (GA), the PSO algorithm is also a new type of iterative algorithm based on the swarm evolution algorithm [61]. The PSO algorithm was proposed by Kennedy and Eberhar in 1995 [62]. Particles can be used to simulate individual birds, and the optimal path to find food is the search process of each particle, so each particle may be a candidate solution to a problem. Particles have only two properties: velocity and position, which are constantly updated through information interactions within the population. When any particle in the population finds the global optimum position, the other particles in the population update their velocity and position according to the optimum position and move closer to the optimum position [63]. The particle swarm optimization algorithm is widely used in various fields due to its simple operation and fast convergence [64]. The particle swarm algorithm first initializes a group of random particles. Assuming that the total number of particles is n , the position and velocity of the i th particle in d -dimensional space are expressed as follows:

$$v_i^d(t) = \omega v_i^d(t-1) + c_1 r_1 (p_{best}^d - x_i^d(t)) + c_2 r_2 (G_{best}^d - x_i^d(t)) \quad (9)$$

$$x_i^d(t) = x_i^d(t-1) + v_i^d(t) \quad (10)$$

where $v_i^d(t)$ is the speed of the particle at time t ; $v_i^d(t-1)$ represents the velocity of the particle at time $t-1$; $x_i^d(t)$ is the position of the particle at time t ; $x_i^d(t-1)$ is the position of the particle at time $t-1$; p_{best}^d denotes the local optimum and G_{best}^d denotes the global optimum; ω means inertia factor. The more significant the inertia factor, the better the global search, and the smaller the inertia factor, the better the local search [65]. r_1 and r_2 are random numbers between 0 and 1. c_1 is the individual learning factor, which represents the ability of the individual to search for the optimal solution, while c_2 is the social learning factor, which means the ability of the group to search for the optimal solution. Learning factors c_1 and c_2 are usually set as 2. If $c_1 = 0$, then the particle is considered to have only social learning ability; at this time, the particle has the ability of extended search and faster convergence speed but lacks the ability of local search and is prone to fall into the problem of local optimal solution on complex problems. If $c_2 = 0$, then the particle is considered to have the only cognitive ability; at this point, the particle behaves similar to a blind random search and converges slowly, which makes it less likely to eventually obtain the optimal solution.

4.3. Adaptive Particle Swarm Optimization

The inertia factor is an essential parameter in the PSO algorithm, which represents the effect of the velocity of the previous generation of particles on the speed of this generation of particles. For ordinary PSO algorithms, the inertia factor is static, but the static inertia factor is often not well adapted to the current environment and balances local search and global search. To improve the optimization ability of the PSO algorithm and reduce the probability of falling into the local extremum value, this study adopts a nonlinear decreasing method to optimize the inertia factor, and the specific expression is as follows [66]:

$$\omega = (\omega_{max} - \omega_{min} - d_1)e^{\frac{1}{1 + \frac{d_2 * t}{T}}} \quad (11)$$

where ω_{max} and ω_{min} are the maximum and minimum values of the initial inertia factor, respectively. When $\omega_{max} = 0.95$ and $\omega_{min} = 0.4$, the algorithm's performance will be significantly improved [67]. The control factors are d_1 and d_2 , and the primary function is to control the inertia factor ω between ω_{max} and ω_{min} ; t is the current number of iterations, and T is the maximum number of iterations.

The improved algorithm allows the particles to have a larger inertia factor at the beginning of the iteration. This enhances the global search ability of the particle and reduces the probability of falling into a local optimum solution. As the number of iterations increases, the inertia factor gradually decreases. At the end of the iteration, the particle has a small inertia factor, which allows the particle to converge quickly to the global optimal solution, increasing the probability of obtaining the global optimal solution.

4.4. Adaptive Simulated Annealing Particle Swarm Optimization

The SA algorithm is a global optimization algorithm inspired by the metal annealing mechanism [56]. The SA algorithm consists of two processes, the Metropolis algorithm, and the annealing. The Metropolis algorithm is the basis for simulated annealing, where the objective function is allowed to degenerate over a range of values in the search for the optimum, allowing it to jump out of the local extremes and find the globally optimal solution. The primary process of the SA algorithm is as follows:

- (1) Initialize the annealing temperature T , generate the initial solution x_0 , and calculate the corresponding objective function value $F(x_0)$.
- (2) Set $T = KT$, where K is the temperature drop rate, $K \in (0, 1)$.
- (3) Apply random perturbation to the current solution x_0 to generate a new solution x_1 and calculate the corresponding objective function value $F(x_1)$, then the difference between the two objective functions is $\Delta F = F(x_1) - F(x_0)$.
- (4) If $\Delta F < 0$, then accept the new solution as the current solution, otherwise, obtain the new solution as the current solution according to probability $\exp(-\Delta F / KT)$.
- (5) After the solution is obtained, whether the number of iterations is reached is judged. If the number of iterations is not reached, go back to steps 3 and 4. If it is reached, it is judged whether the termination condition ($\Delta F < 0$) is met. If the condition is met, output the result; otherwise, go back to step 2.

The SA algorithm was introduced into the adaptive particle swarm optimization algorithm because the PSO algorithm is prone to local extremes late in the iteration, while the global search capability of the SA algorithm is stronger and can effectively compensate for the shortcomings of the PSO algorithm. The workflow of the ASAPSO-ANN model developed in this study is shown in Figure 5.

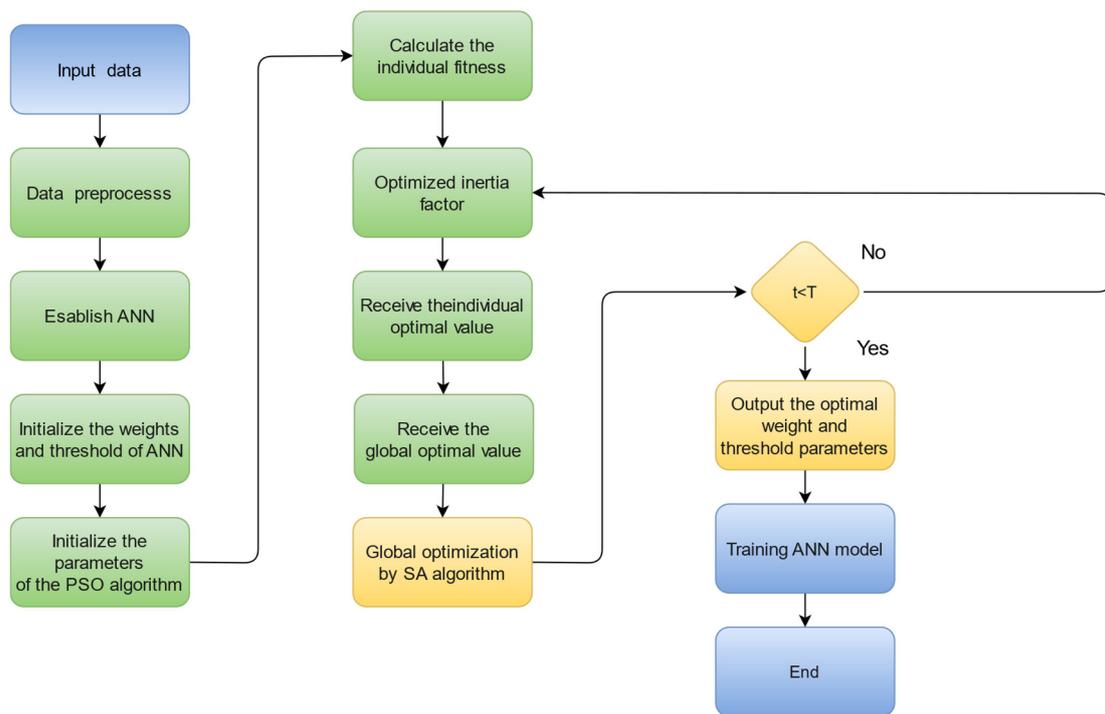


Figure 5. Workflow of the ASAPSO-ANN model.

5. Evaluation of the Model

In the present work, four statistical criteria are used to evaluate the accuracy of neural network models, namely, mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), and coefficient of determination (R^2). The coefficient of determination (R^2) is widely used in regression problems [68]. Its main function is to evaluate the correlation between the actual target and the output target [69]. RMSE and MAE are used to evaluate the average error between the real target and the output target [70–72]. The accuracy of the model gradually improves as R^2 approaches 1, while MSE and MAE approach 0. The expressions of these four evaluation indicators are as follows [73,74]:

$$R^2 = \frac{\sum_{k=1}^N (q_{0,k} - \bar{q}_0)(q_{t,k} - \bar{q}_t)}{\sqrt{\sum_{k=1}^N (q_{0,k} - \bar{q}_0)^2 \sum_{k=1}^N (q_{t,k} - \bar{q}_t)^2}} \tag{12}$$

$$MAE = \frac{\sum_{k=1}^N |q_{0,k} - q_{t,k}|}{N} \tag{13}$$

$$MSE = \frac{1}{N} \sum_{k=1}^N (q_{0,k} - q_{t,k})^2 \tag{14}$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{k=1}^N (q_{0,k} - q_{t,k})^2} \tag{15}$$

where N is the number of samples; q_0 is the actual target value; \bar{q}_0 is the real target average value; q_t represents the output target value; and \bar{q}_t represents the output target average value, $k = 1:N$.

6. Results of the Three Models

The objective of the computational analysis is to build three neural network models (ANN, PSO-ANN, ASAPSO-ANN) to predict the compressive strength of rubber concrete at 28 days using a database containing 307 sample data. The model constructed and the results of the calculations are described in detail in the following subsections. Since each calculation of the artificial neural network is an approximate solution, the results for each

model are averaged to avoid chance in the results. The best results from each model are selected for graphical analysis.

6.1. ANN Model

A three-layer feedforward neural network model was built, and the number of neurons in the hidden layer was obtained between 4 and 13 by Equation (7), and then the trial-and-error method was used to obtain the optimal number of neurons as 6. Thus, the neural network structure was 11–6–1 (Figure 6). The parameters of the neural network model in this study are shown in Table 2.

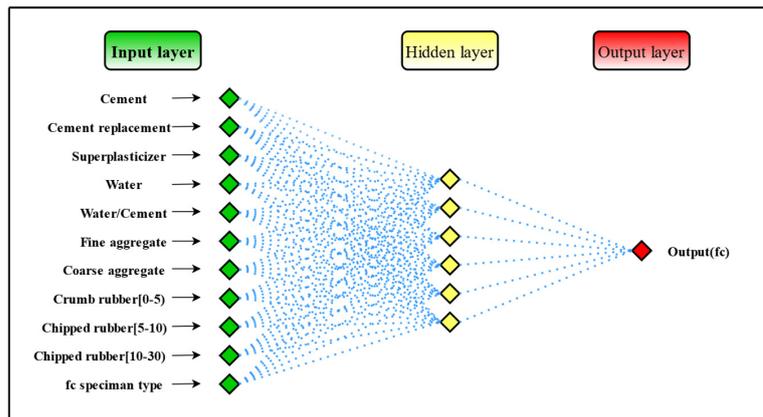


Figure 6. ANN structure used in this study.

The ANN model performance evaluation results are presented in Table 4. The average value of the training set R^2 of the ordinary ANN model was 0.8990, and the average value of the testing set R^2 was 0.8385. From these data, the R^2 for both the training and testing sets of the standard ANN model were relatively good (published literature suggests that models were high-precision when $R > 0.8$ and RMSE, MAE are low [75]). The mean value of the RMSE for the training set of the ordinary ANN model was 5.0237, and the testing set was 4.9673. The mean value of the MAE was 3.7363 for the training set and 4.2117 for the testing set. From these data, it could be seen that the common ANN model has a high error, but the RMSE and MAE values are closer. It could be concluded that the ordinary ANN model has some generalization ability and could predict the unknown data to some extent, but the prediction accuracy of the model was not high.

6.2. PSO-ANN Model

Weights and biases have a significant impact on the prediction results of the ordinary ANN model. Therefore, PSO algorithms were needed to find the optimal weight and biases to achieve the prediction of the target. For the PSO-ANN model, in addition to the usual parameters, the population size, number of iterations, social learning factor, individual learning factor, particle position constraint, particle velocity constraint, and inertia factor also needed to be set. The number of neurons in the hidden layer of the PSO-ANN model is also set to six. The parameter settings for the PSO-ANN in this study are shown in Table 3.

Table 3. Parameter settings for the PSO-ANN model.

Parameter	Setting
Popsiz	15
Maxgen	500
c_1	2
c_2	2
ω	0.95
Position constraint	[−3,3]
Velocity constraint	[−3,3]

The performance evaluation results of the PSO-ANN model are shown in Table 4. The mean value of the R^2 for the training set of the PSO-ANN model was 0.9516, and the mean value of R^2 for the testing set was 0.8732. The mean value of the RMSE for the training set of the PSO-ANN model was 3.4573, and the testing set was 3.6340. The mean value of the MAE for the training set was 2.5493, and the testing set was 2.7260. From these data, the R^2 of both the training and testing sets of the PSO-ANN model improved compared to the normal ANN model (R^2 was 0.8990 for the training set and 0.8385 for the testing set of the ANN model). The RMSE for both the training and testing sets of the PSO-ANN model was reduced to varying degrees compared to the standard ANN model (RMSE was 5.0237 for the training set and 4.9673 for the testing set of the ANN model). The MAE of the training and testing sets was also lower than the ANN model (MAE was 3.7363 for the training set and 4.2117 for the testing set for the ANN model). Thus, the PSO algorithm improved the predictive ability of the ANN model.

Table 4. Results of the performance evaluation of the three models.

		ANN	PSO-ANN	ASAPSO-ANN
R^2	Train	0.8990	0.9516	0.9554
	Test	0.8385	0.8732	0.9240
MSE	Train	26.8847	12.0370	11.0969
	Test	25.1023	13.3453	7.4011
RMSE	Train	5.0237	3.4573	3.3238
	Test	4.9673	3.6340	2.7805
MAE	Train	3.7363	2.5493	2.4016
	Test	4.2117	2.7260	2.1088

6.3. ASAPSO-ANN Model

Introduction of the SA algorithm into the PSO algorithm improved the global search capability of the algorithm and the probability of jumping out of the local extremum. The inertia factor was optimized by Equation (11). For ASAPSO-ANN, the initial temperature and the temperature decay coefficient needed to be set. The initial temperature used in this study was 200 degrees, while the temperature decay coefficient was 0.95.

The results of the performance evaluation of the ASAPSO-ANN model are presented in Table 4. The mean value of R^2 for the training set of the ASAPSO-ANN model was 0.9554, and the testing set was 0.9240. The mean value of the RMSE for the training set was 3.3238, and the testing set was 2.7805. The mean value of the training MAE was 2.4016, and the mean of the testing set was 2.1088. From these data, the performance of the neural network model was further improved by adding the SA algorithm. Compared to the PSO-ANN model, the ASAPSO-ANN model has a higher R^2 on both the training and testing sets (R^2 was 0.9516 for the PSO-ANN model training set and 0.8732 for the PSO-ANN testing set), and a lower RMSE (RMSE was 3.4573 for the training set and 3.6340 for the testing set of the PSO-ANN model). The mean value of MAE decreased from 2.5493 to 2.4016 for the training set and from 2.7260 to 2.1088 for the testing. The R^2 of the testing set of the ASAPSO-ANN model was significantly higher than that of the PSO-ANN model, while other error metrics were also lower than that of the PSO-ANN model. This demonstrated that the ASAPSO algorithm further improved the predictive ability of the ANN model.

6.4. Weights and Biases of Neural Networks for the Three Models

This section gives the weight matrices for the three neural network models so that the neural network models can be applied. The weight matrices for the ANN model, the PSO-ANN model, and the ASAPSO-ANN model are shown in Appendix A.

7. Discussion

This study focused on the prediction of the compressive strength of rubber concrete using a neural network optimized by the ASAPSO algorithm. Since a single optimization

algorithm has its shortcomings, it was necessary to mix the algorithms to compensate for their shortcomings between them. In the previous section, the results of the three models ANN, PSO-ANN, and ASAPSO-ANN are presented in Table 4, but it was necessary to discuss the results from a more intuitive point of view.

The regression analysis for the three models are presented in Figure 7. Panels a, c, and e denote training set results, and b, d, and f denote testing set results. The figure shows that most of the data points in the training set of the three models were distributed on both sides of the fitted line and the R^2 were good. This indicates that all three neural network models had a good fitting ability. Among these, the ASAPSO-ANN model had the highest R^2 , which is the indication of a stronger fitting ability. The results from Figure 7b,d,f showed a significant increase in R^2 in the testing set compared to the training set, which increased from 0.8564 for the ANN model to 0.9396 for the ASAPSO-ANN model. Meanwhile, the ASAPSO-ANN model had the lowest error in each. Since the testing set did not participate in the training of the model, it could be shown that the ANN model with the introduction of the ASAP SO algorithm had a substantial improvement in predictive ability. This conclusion could also be drawn from Figures 8 and 9. As can be seen in Figures 8f and 9, the predicted value of the ASAPSO-ANN model was closer to the actual value compared to the ANN and PSO-ANN models. The error analysis for the three models is shown in Figure 10.

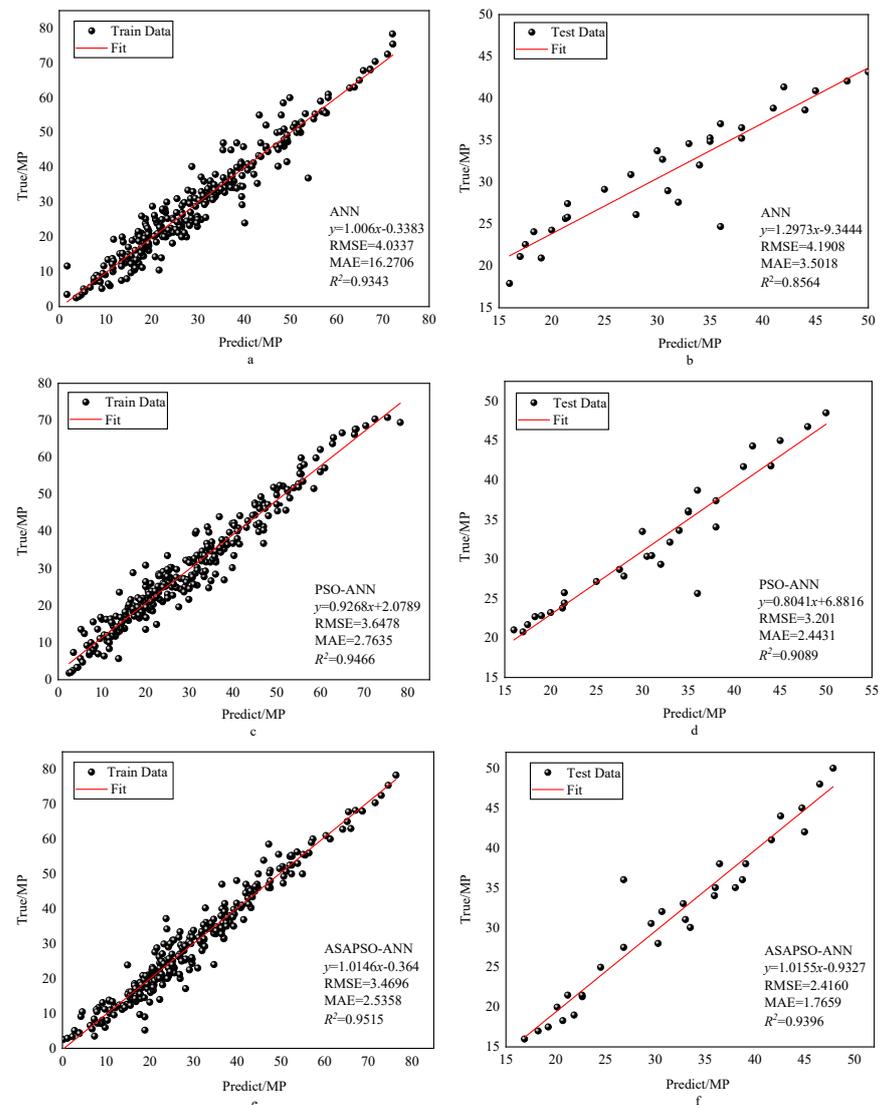


Figure 7. Regression analysis of the three models: (a,b) for the ANN model; (c,d) for the PSO-ANN model; (e,f) for ASAPSO-ANN model, for training and testing phases, respectively.

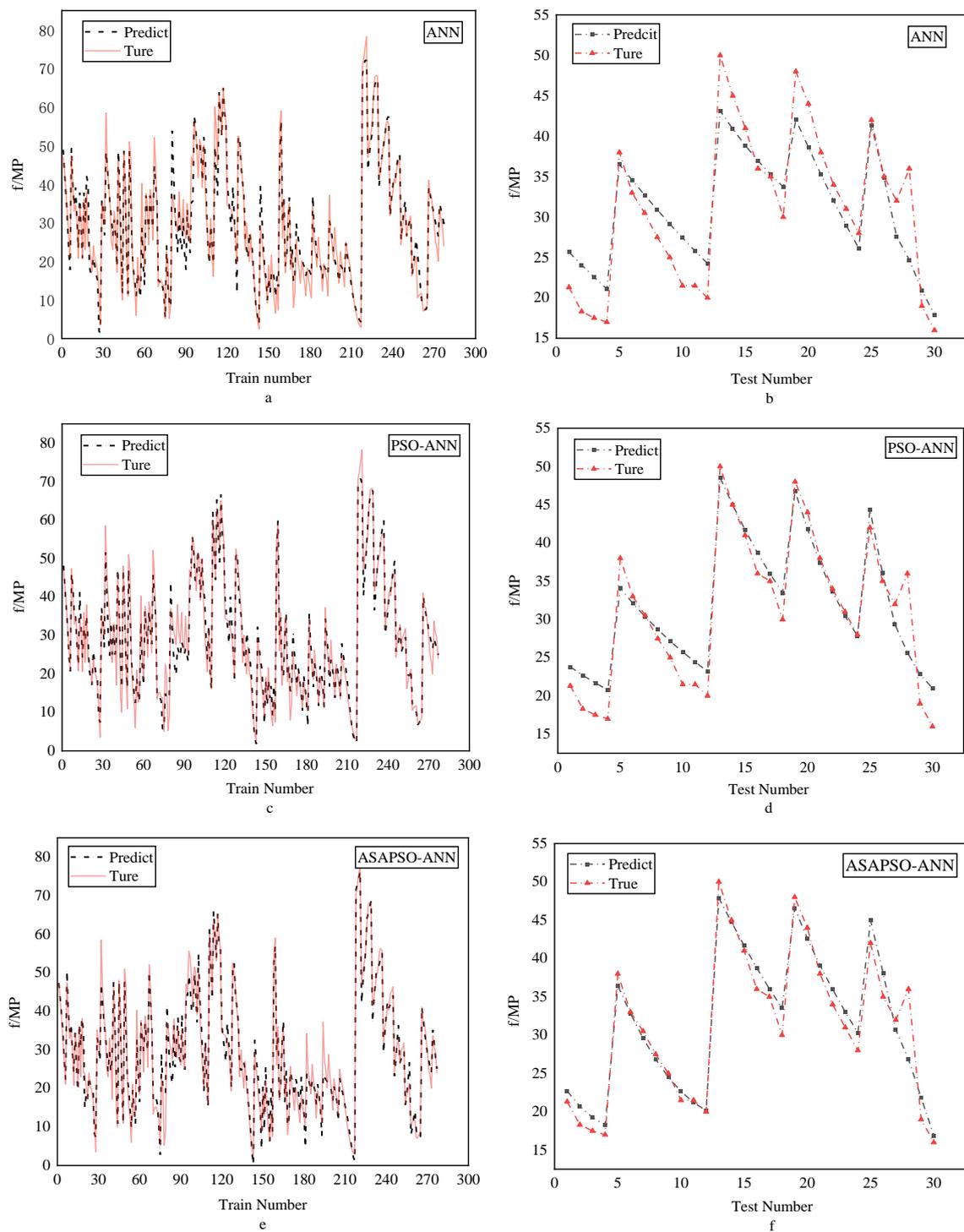


Figure 8. Comparison of actual and predicted values for the three models: (a,b) for ANN model; (c,d) for PSO-ANN model; (e,f) for ASAPSO-ANN model, for training and testing phases, respectively.

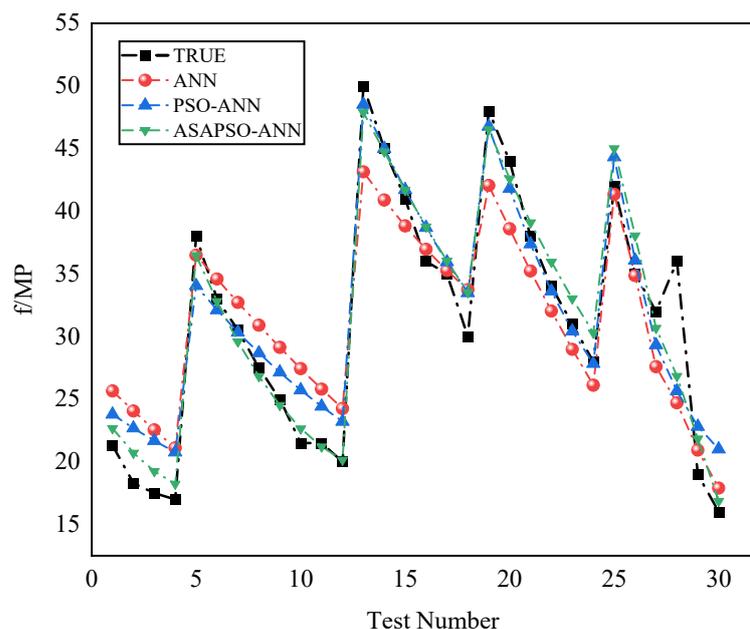


Figure 9. Comparison of data from three model testing sets.

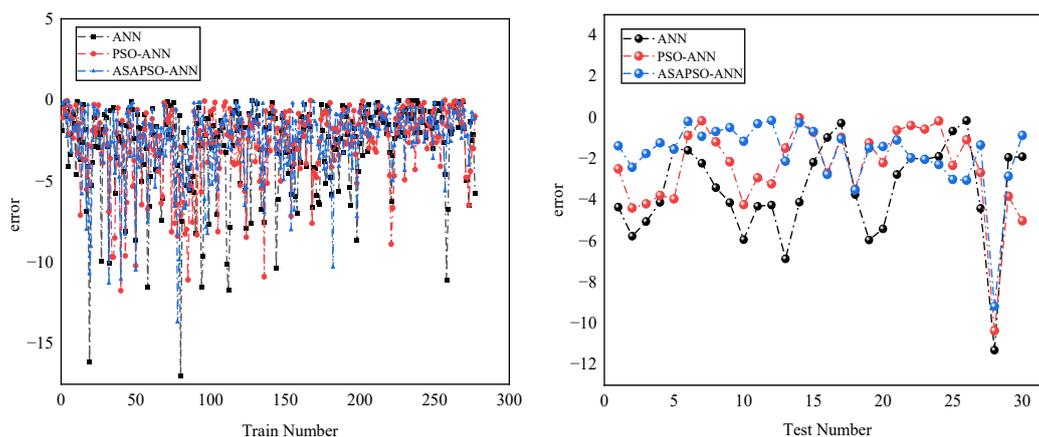


Figure 10. Comparison of errors in the training (right) and testing (left) sets of three models.

In summary, the introduction of the algorithm proved to be successful and effective. The PSO algorithm optimized the weights and biases of the ANN model using its optimization capabilities, resulting in improved model performance and reduced prediction errors. The introduction of the SA algorithm in the PSO algorithm successfully compensated for the shortcomings of the PSO algorithm in terms of fast convergence at the end of iteration and the tendency to fall into local extremum, so that the probability of obtaining the globally best model was greatly increased. It can also be seen that the adaptive treatment of inertia factor successfully helped the PSO algorithm to perform sufficient optimization search in the early stage and converge to the global best quickly in the later stage. This shows that the hybrid algorithm improved the prediction accuracy of the model more than the single optimization algorithm. It also shows that the hybrid algorithm could be used to study the strength of rubber concrete.

To further validate the reliability of the proposed model, the traditional multiple linear regression model was introduced here [17]. The multiple linear regression (MLR) model was similar to the neural network model in that both studied the effects of multiple variables; therefore, the two models could be used for comparison. The results of the regression analysis of the two models are shown in Figure 11. A comparison of the indicators is shown in Table 5. As can be seen from Figure 11 and Table 5, the ASAPSO-ANN model

outperforms the MLR model. Comparison with empirical formulae was also necessary. M. Reda Taha modeled the empirical formula using polynomials [26]. Since the empirical formula only used the percentage of rubber content as an input variable, the input variable for the ASAPSO-ANN model was also changed to the percentage of rubber content. To avoid complex calculations, a randomly selected cited study [39] was used for calculations using both models. Figure 12 shows the results of the regression analysis for both models. A comparison of the indicators is shown in Table 6. From Figure 12a, it can be seen that the empirical equation model had a high R^2 , but its horizontal and vertical values were very different. This indicates that there was a large error between the predicted and actual values. This is also reflected in the other indicator values in Table 6. This indicates the poor predictive ability of the empirical formula model. In contrast, the ASAPSO-ANN model represented in Figure 12b did not suffer from such problems and could predict the strength values of the concrete with accuracy. For these types of empirical formulas, the influence of multiple factors on strength was usually not taken into account, and therefore the ASAPSO-ANN model had a wider application capability than the empirical formulas. Comparison of other neural network models were also necessary. The comparison results are shown in Table 7. As can be seen from Table 6, ASAPSO-ANN performed better than the conventional ANN model when faced with large datasets. Therefore, the proposed model is feasible for strength studies of rubber concrete.

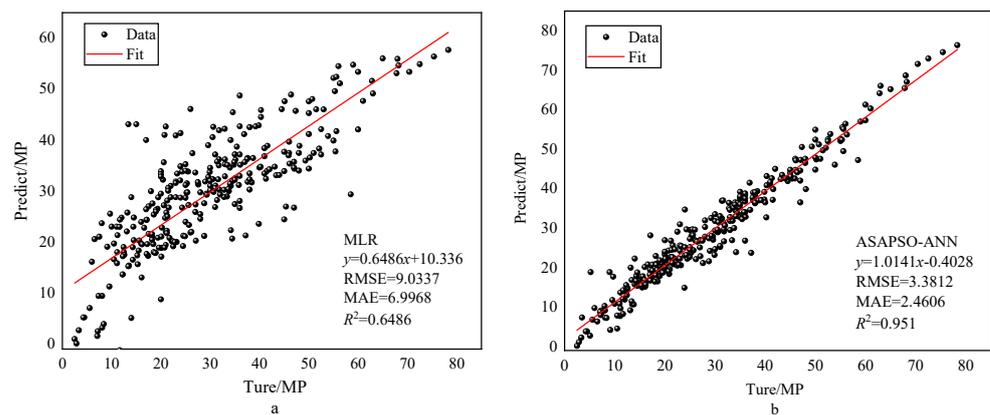


Figure 11. Results of regression analysis of two models: (a) for MRL and (b) for ASAPSO-ANN.

Table 5. Metrics for MLR and ASAPSO-ANN models.

	R^2	MSE	RMSE	MAE
ASAPSO-ANN	0.951	11.4323	3.3812	2.4606
MLR	0.6486	81.6069	9.0337	6.9968

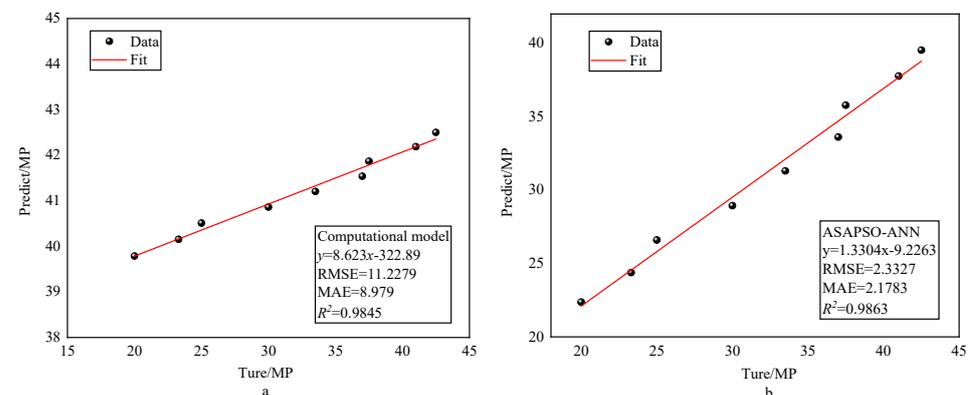


Figure 12. Results of regression analysis of two models: (a) for empirical formula and (b) for ASAPSO-ANN.

Table 6. Metrics for empirical formula and ASAPSO-ANN models.

	R^2	MSE	RMSE	MAE
ASAPSO-ANN	0.9863	5.4414	2.3327	2.1783
M. Reda Taha [26]	0.9845	126.0657	11.2279	8.979

Table 7. Comparison with models in the literature.

	ML Algorithm	Structure	Dataset	Performance
This study	ASAPSO algorithm with ANN	11–6–1	307	R = 0.9774(train) R = 0.9612(test) R = 0.9752(all)
Khoja [9]	ANN with Levenberg–Marquardt algorithm	5–10–1	287	R = 0.954(all)
Abdollahzadeh [17]	ANN multi-layered perceptron (BP)	3–1–1	20	R = 0.9885(train) R = 0.9824test

8. Conclusions and Future Prospect

In this study, an ASAPSO-ANN model was developed for predicting the compressive strength of rubber concrete. The data were obtained from the published literature. Both ANN and PSO-ANN artificial neural network models were developed for comparative analysis. RMS, R^2 , MSE, and MAE were used to evaluate the model performance. According to the results of the testing phase, all three artificial neural networks established in this study have predictive capabilities, but the proposed ASAPSO-ANN model has the highest prediction accuracy compared to the ANN model and the PSO-ANN model ($R^2 = 0.9240$, RMS = 2.7805, MSE = 7.8028, and MAE = 2.1088). By comparing the prediction curves of the testing set of the three models, it can also be obtained that the prediction curves of the ASAPSO-ANN model are closer to the actual curves. This led to the conclusion that the SA algorithm successfully compensates for the shortcomings of the PSO algorithm, proving the validity of the proposed model.

In summary, this study successfully demonstrated that the proposed ASAPSO algorithm improved the performance and accuracy of the network. It also showed that hybrid algorithm could be used to optimize neural networks for predicting the strength of rubber concrete. This provided a new option for predicting the strength of rubber concrete using neural network models. However, observation of the results revealed that the proposed method could be further optimized in the future to obtain better results, which included making the dataset more comprehensive, considering the issue of outliers in the model, and comparing it with other machine learning models.

Author Contributions: Conceptualization, X.-Y.H. and K.-Y.W.; methodology, X.-Y.H.; software, X.-Y.H.; testing, X.-Y.H. and K.-Y.W.; formal analysis, X.-Y.H.; investigation, X.-Y.H. and W.-C.D.; resources, X.-Y.H.; data curation, X.-Y.H. and W.-C.D.; writing—original draft preparation, X.-Y.H.; writing—review and editing, H.-M.L.; supervision, S.W. and T.L.; project administration, S.W., T.L., Y.-F.L. and H.-M.L.; funding acquisition, Y.-F.L., S.W. and T.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the 2021 Hubei Construction Science and Technology Plan Project, Grant number 43, and by the National Natural Science Foundation of China under Grant No. 42071264.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in the article can be obtained from the author here.

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

Appendix A

ANN:

$$w1 = \begin{bmatrix} 0.130 & -0.636 & -0.410 & 0.336 & -0.860 & -0.00 & 0.439 & -0.130 & -0.700 & -0.096 & -0.751 \\ -0.102 & 0.470 & 0.171 & 0.605 & 0.757 & -0.362 & 0.373 & 0.725 & 0.673 & -0.466 & -0.220 \\ 0.421 & 0.603 & -0.459 & -0.307 & -0.613 & -0.540 & 0.411 & 0.298 & -0.704 & 0.571 & 0.350 \\ -0.231 & -0.037 & -0.460 & 0.668 & -0.428 & 0.637 & -0.545 & -0.640 & -0.581 & 0.588 & -0.155 \\ -0.835 & -0.370 & 0.098 & 0.018 & 0.328 & 0.153 & -0.630 & -0.874 & -0.022 & 0.660 & -0.378 \\ -0.042 & 0.474 & -0.618 & 0.360 & 0.467 & -0.439 & 0.198 & -0.601 & -0.621 & -0.612 & -0.636 \end{bmatrix}$$

$$w2 = [-0.235 \ -0.249 \ -0.768 \ -0.799 \ 0.014 \ -0.885]$$

$$b1 = [-1.648 \ 0.989 \ -0.330 \ -0.330 \ -0.986 \ -1.648]^T$$

$$b2 = [-0.227]^T$$

PSO-ANN:

$$w3 = \begin{bmatrix} -2.428 & -0.857 & -0.303 & 2.257 & -1.039 & 0.364 & -0.159 & 5.346 & 0.870 & 2.751 & -0.311 \\ 3.711 & 1.442 & 4.023 & 2.092 & 0.645 & 3.596 & -1.585 & -0.736 & 0.133 & 2.603 & 2.975 \\ 2.762 & 1.015 & -0.196 & -1.843 & 0.933 & -1.459 & -0.184 & -4.372 & -5.070 & -2.021 & 0.706 \\ 0.970 & 1.115 & 1.016 & 5.295 & -0.281 & 3.921 & 7.953 & -2.226 & -0.617 & 1.082 & 9.331 \\ 0.871 & 3.674 & 3.767 & 3.616 & -1.162 & -2.082 & 4.125 & -3.288 & -2.313 & -3.038 & 0.561 \\ 7.484 & 0.425 & 6.608 & 13.598 & -3.780 & 5.058 & -3.747 & 1.662 & -2.279 & -2.991 & -0.761 \end{bmatrix}$$

$$w4 = [-1.380 \ -2.93 \ -1.557 \ 0.209 \ -0.040 \ 0.064]$$

$$b3 = [8.195 \ -1.706 \ -12.078 \ 6.403 \ 2.066 \ -2.192]^T$$

$$b4 = [-3.921]^T$$

ASAPSO-ANN:

$$w5 = \begin{bmatrix} -5.804 & -2.772 & -1.222 & 0.163 & -0.356 & -2.316 & 1.700 & 3.703 & 0.884 & 3.381 & 3.198 \\ -4.767 & -0.563 & -0.436 & -0.586 & 2.726 & 2.958 & 6.314 & -6.992 & 3.005 & 6.417 & -0.152 \\ -2.509 & 1.148 & -1.844 & -3.255 & 1.088 & 1.858 & -0.074 & -1.452 & 1.547 & 1.895 & 2.816 \\ 1.370 & 1.195 & -2.008 & -4.159 & 1.528 & -4.568 & -2.848 & -2.623 & -1.290 & 1.446 & -0.018 \\ 0.445 & 0.224 & 0.339 & -0.882 & 0.135 & 0.082 & 0.316 & -3.514 & -0.743 & -2.287 & 0.008 \\ 3.672 & 0.538 & -1.742 & -2.114 & 0.364 & -4.714 & 2.701 & 0.646 & -0.439 & 2.871 & 6.128 \end{bmatrix}$$

$$w6 = [-0.250 \ 0.290 \ -0.091 \ -0.283 \ 1.944 \ 0.288]$$

$$b5 = [7.154 \ -1.294 \ -3.143 \ -2.614 \ -7.033 \ 4.742]^T$$

$$b6 = [1.240]^T$$

References

1. Sun, Y.; Li, G.; Zhang, J.; Qian, D. Prediction of the Strength of Rubberized Concrete by an Evolved Random Forest Model. *Adv. Civ. Eng.* **2019**, *2019*. [[CrossRef](#)]
2. Azevedo, F.; Pacheco-Torgal, F.; Jesus, C.; Barroso de Aguiar, J.L.; Camoes, A.F. Properties and durability of HPC with tyre rubber wastes. *Constr. Build. Mater.* **2012**, *34*, 186–191. [[CrossRef](#)]
3. Zhang, Z.; Ma, H.; Qian, S. Investigation on Properties of ECC Incorporating Crumb Rubber of Different Sizes. *J. Adv. Concr. Technol.* **2015**, *13*, 241–251. [[CrossRef](#)]
4. Toutanji, H.A. The use of rubber tire particles in concrete to replace mineral aggregates. *Cem. Concr. Compos.* **1996**, *18*, 135–139. [[CrossRef](#)]

5. Skripiūnas, G.; Grinys, A.; Černius, B. Deformation properties of concrete with rubber waste additives. *Mater. Sci.* **2007**, *13*, 219–223.
6. Mohammed, B.S.; Azmi, N. Strength reduction factors for structural rubbercrete. *Front. Struct. Civ. Eng.* **2014**, *8*, 270–281. [[CrossRef](#)]
7. Hadzima-Nyarko, M.; Nyarko, E.K.; Ademović, N.; Miličević, I.; Kalman Šipoš, T. Modelling the influence of waste rubber on compressive strength of concrete by artificial neural networks. *Materials* **2019**, *12*, 561. [[CrossRef](#)]
8. Ganjian, E.; Khorami, M.; Maghsoudi, A.A. Scrap-tyre-rubber replacement for aggregate and filler in concrete. *Constr. Build. Mater.* **2009**, *23*, 1828–1836. [[CrossRef](#)]
9. El-Khoja, A.; Ashour, A.; Abdalhmud, J.; Dai, X.; Khan, A. Prediction of Rubberised Concrete Strength by Using Artificial Neural Networks. *Training* **2018**, *30*, 35.
10. Batayneh, M.K.; Marie, I.; Asi, I. Promoting the use of crumb rubber concrete in developing countries. *Waste Manag.* **2008**, *28*, 2171–2176. [[CrossRef](#)]
11. Aslani, F. Mechanical properties of waste tire rubber concrete. *J. Mater. Civ. Eng.* **2016**, *28*, 04015152. [[CrossRef](#)]
12. Khatib, Z.K.; Bayomy, F.M. Rubberized Portland cement concrete. *J. Mater. Civ. Eng.* **1999**, *11*, 206–213. [[CrossRef](#)]
13. Muyen, Z.; Mahmud, F.; Hoque, M. Application of waste tyre rubber chips as coarse aggregate in concrete. *Prog. Agric.* **2019**, *30*, 328–334. [[CrossRef](#)]
14. Marshal, S. *Machine Learning an Algorithm Perspective*; CRC Press: Boca Raton, FL, USA, 2015.
15. Nasrollahzadeh, K.; Nouhi, E. Fuzzy inference system to formulate compressive strength and ultimate strain of square concrete columns wrapped with fiber-reinforced polymer. *Neural Comput. Appl.* **2018**, *30*, 69–86. [[CrossRef](#)]
16. Ahmad, M.; Hu, J.-L.; Ahmad, F.; Tang, X.-W.; Amjad, M.; Iqbal, M.J.; Asim, M.; Farooq, A. Supervised learning methods for modeling concrete compressive strength prediction at high temperature. *Materials* **2021**, *14*, 1983. [[CrossRef](#)] [[PubMed](#)]
17. Abdollahzadeh, A.; Masoudnia, R.; Aghababaei, S. Predict strength of rubberized concrete using artificial neural network. *WSEAS Trans. Comput.* **2011**, *10*, 31–40.
18. Kumarappan, N.; Suresh, K. Combined SA PSO method for transmission constrained maintenance scheduling using leveled risk method. *Int. J. Electr. Power Energy Syst.* **2015**, *73*, 1025–1034. [[CrossRef](#)]
19. Cheng, M.; Liu, B. Application of an extended VES production function model based on improved PSO algorithm. *Soft Comput.* **2021**, *25*, 7937–7945. [[CrossRef](#)]
20. Chang, J.; Li, Z.; Huang, Y.; Yu, X.; Jiang, R.; Huang, R.; Yu, X. Multi-objective optimization of a novel combined cooling, dehumidification and power system using improved M-PSO algorithm. *Energy* **2022**, *239*, 122487. [[CrossRef](#)]
21. Liang, J.; Suganthan, P.; Chan, C.; Huang, V. Wavelength detection in FBG sensor network using tree search DMS-PSO. *IEEE Photonics Technol. Lett.* **2006**, *18*, 1305–1307. [[CrossRef](#)]
22. Paine, K.A.; Dhir, R.; Moroney, R.; Kopasakis, K. Use of crumb rubber to achieve freeze thaw resisting concrete. In Proceedings of the International Conference on Concrete for Extreme Conditions; University of Dundee: Scotland, UK, 2002; pp. 486–498.
23. Güneyisi, E.; Gesoğlu, M.; Özturan, T. Properties of rubberized concretes containing silica fume. *Cem. Concr. Res.* **2004**, *34*, 2309–2317. [[CrossRef](#)]
24. Albano, C.; Camacho, N.; Reyes, J.; Feliu, J.; Hernández, M. Influence of scrap rubber addition to Portland I concrete composites: Destructive and non-destructive testing. *Compos. Struct.* **2005**, *71*, 439–446. [[CrossRef](#)]
25. Gesoğlu, M.; Güneyisi, E. Strength development and chloride penetration in rubberized concretes with and without silica fume. *Mater. Struct.* **2007**, *40*, 953–964. [[CrossRef](#)]
26. Reda Taha, M.M.; El-Dieb, A.S.; Abd El-Wahab, M.; Abdel-Hameed, M. Mechanical, fracture, and microstructural investigations of rubber concrete. *J. Mater. Civ. Eng.* **2008**, *20*, 640–649. [[CrossRef](#)]
27. Zheng, L.; Huo, X.S.; Yuan, Y. Strength, modulus of elasticity, and brittleness index of rubberized concrete. *J. Mater. Civ. Eng.* **2008**, *20*, 692–699. [[CrossRef](#)]
28. Aiello, M.A.; Leuzzi, F. Waste tyre rubberized concrete: Properties at fresh and hardened state. *Waste Manag.* **2010**, *30*, 1696–1704. [[CrossRef](#)]
29. Gesoğlu, M.; Güneyisi, E.; Özturan, T.; Özbay, E. Modeling the mechanical properties of rubberized concretes by neural network and genetic programming. *Mater. Struct.* **2010**, *43*, 31–45. [[CrossRef](#)]
30. Ghedan, R.H.; Hamza, D.M. Effect of rubber treatment on compressive strength and thermal conductivity of modified rubberized concrete. *J. Eng. Dev* **2011**, *15*, 21–29.
31. Ozbay, E.; Lachemi, M.; Sevim, U.K. Compressive strength, abrasion resistance and energy absorption capacity of rubberized concretes with and without slag. *Mater. Struct.* **2011**, *44*, 1297–1307. [[CrossRef](#)]
32. Grinys, A.; Sivilevičius, H.; Daukšys, M. Tyre rubber additive effect on concrete mixture strength. *J. Civ. Eng. Manag.* **2012**, *18*, 393–401. [[CrossRef](#)]
33. Rahman, M.; Usman, M.; Al-Ghalib, A.A. Fundamental properties of rubber modified self-compacting concrete (RMSCC). *Constr. Build. Mater.* **2012**, *36*, 630–637. [[CrossRef](#)]
34. Al-Tayeb, M.; Abu Bakar, B.; Akil, H.; Ismail, H. Performance of rubberized and hybrid rubberized concrete structures under static and impact load conditions. *Exp. Mech.* **2013**, *53*, 377–384. [[CrossRef](#)]
35. Dong, Q.; Huang, B.; Shu, X. Rubber modified concrete improved by chemically active coating and silane coupling agent. *Constr. Build. Mater.* **2013**, *48*, 116–123. [[CrossRef](#)]

36. Gesoğlu, M.; Güneyisi, E.; Khoshnaw, G.; İpek, S. Investigating properties of pervious concretes containing waste tire rubbers. *Constr. Build. Mater.* **2014**, *63*, 206–213. [[CrossRef](#)]
37. Grdić, Z.; Topličić-Curčić, G.; Ristić, N.; Grdić, D.; Mitković, P. Hydro-abrasive resistance and mechanical properties of rubberized concrete. *Građevinar* **2014**, *66*, 11–20.
38. Onuaguluchi, O.; Panesar, D.K. Hardened properties of concrete mixtures containing pre-coated crumb rubber and silica fume. *J. Clean. Prod.* **2014**, *82*, 125–131. [[CrossRef](#)]
39. Thomas, B.S.; Gupta, R.C.; Kalla, P.; Csetenyi, L. Strength, abrasion and permeation characteristics of cement concrete containing discarded rubber fine aggregates. *Constr. Build. Mater.* **2014**, *59*, 204–212. [[CrossRef](#)]
40. Wang, L.; Huang, Y.H. Study on rubber particles modified concrete. In *Applied Mechanics and Materials*; Trans Tech Publications Ltd.: Bäch, Switzerland, 2014; pp. 953–958. [[CrossRef](#)]
41. Youssf, O.; ElGawady, M.A.; Mills, J.E.; Ma, X. An experimental investigation of crumb rubber concrete confined by fibre reinforced polymer tubes. *Constr. Build. Mater.* **2014**, *53*, 522–532. [[CrossRef](#)]
42. Abusharar, S.W. Effect of particle sizes on mechanical properties of concrete containing crumb rubber. *Innov. Syst. Des. Eng* **2015**, *6*, 114–125.
43. Gesoğlu, M.; Güneyisi, E.; Hansu, O.; İpek, S.; Asaad, D.S. Influence of waste rubber utilization on the fracture and steel–concrete bond strength properties of concrete. *Constr. Build. Mater.* **2015**, *101*, 1113–1121. [[CrossRef](#)]
44. Herrera-Sosa, E.S.; Martínez-Barrera, G.; Barrera-Díaz, C.; Cruz-Zaragoza, E.; Ureña-Núñez, F. Recovery and modification of waste tire particles and their use as reinforcements of concrete. *Int. J. Polym. Sci.* **2015**, *2015*. [[CrossRef](#)]
45. Ismail, M.K.; De Grazia, M.T.; Hassan, A.A. Mechanical properties of self-consolidating rubberized concrete with different supplementary cementing materials. In Proceedings of the International Conference on Transportation and Civil Engineering (ICTCE'15), London, UK, 21–22 March 2015; pp. 21–22.
46. Mishra, M.; Panda, K. An experimental study on fresh and hardened properties of self compacting rubberized concrete. *Indian J. Sci. Technol.* **2015**, *8*, 1–10. [[CrossRef](#)]
47. Selvakumar, S.; Venkatakrishnaiah, R. Strength properties of concrete using crumb rubber with partial replacement of fine aggregate. *Int. J. Innov. Res. Sci. Eng. Technol.* **2015**, *4*, 1171–1175.
48. Liu, H.; Wang, X.; Jiao, Y.; Sha, T. Experimental investigation of the mechanical and durability properties of crumb rubber concrete. *Materials* **2016**, *9*, 172. [[CrossRef](#)]
49. Marie, I. Zones of weakness of rubberized concrete behavior using the UPV. *J. Clean. Prod.* **2016**, *116*, 217–222. [[CrossRef](#)]
50. Zaoiai, S.; Makani, A.; Tafraoui, A.; Benmerioul, F. Optimization and Mechanical Characterization of Self-Compacting Concrete Incorporating Rubber Aggregates. *Asian J. Civ. Eng. (Build. Hous.)* **2016**, *17*, 817–829.
51. Asutkar, P.; Shinde, S.; Patel, R. Study on the behaviour of rubber aggregates concrete beams using analytical approach. *Eng. Sci. Technol. Int. J.* **2017**, *20*, 151–159. [[CrossRef](#)]
52. Murugan, R.B.; Sai, E.R.; Natarajan, C.; Chen, S.E. Flexural fatigue performance and mechanical properties of rubberized concrete. *Građevinar* **2017**, *69*, 983–990.
53. Azim, I.; Yang, J.; Javed, M.F.; Iqbal, M.F.; Mahmood, Z.; Wang, F.; Liu, Q.F. Prediction Model for Compressive Arch Action Capacity of RC Frame Structures under Column Removal Scenario Using Gene Expression Programming. *Structures* **2020**, *25*, 212–228. [[CrossRef](#)]
54. Dunlop, P.; Smith, S. Estimating key characteristics of the concrete delivery and placement process using linear regression analysis. *Civ. Eng. Environ. Syst.* **2003**, *20*, 273–290. [[CrossRef](#)]
55. Smith, G.N. *Probability and Statistics in Civil Engineering*; Collins Professional Technical Books: London, UK, 1986; Volume 244.
56. Jahed Armaghani, D.; Hajihassani, M.; Sohaei, H.; Tonnizam Mohamad, E.; Marto, A.; Motaghedhi, H.; Moghaddam, M.R. Neuro-fuzzy technique to predict air-overpressure induced by blasting. *Arab. J. Geosci.* **2015**, *8*, 10937–10950. [[CrossRef](#)]
57. Topçu, İ.B.; Sarıdemir, M. Prediction of rubberized concrete properties using artificial neural network and fuzzy logic. *Constr. Build. Mater.* **2008**, *22*, 532–540. [[CrossRef](#)]
58. Duan, Z.H.; Kou, S.C.; Poon, C.S. Prediction of compressive strength of recycled aggregate concrete using artificial neural networks. *Constr. Build. Mater.* **2013**, *40*, 1200–1206. [[CrossRef](#)]
59. Dahou, Z.; Sbartai, Z.M.; Castel, A.; Ghomari, F. Artificial neural network model for steel–concrete bond prediction. *Eng. Struct.* **2009**, *31*, 1724–1733. [[CrossRef](#)]
60. Liu, Q.F.; Iqbal, M.F.; Yang, J.; Lu, X.Y.; Zhang, P.; Rauf, M. Prediction of chloride diffusivity in concrete using artificial neural network: Modelling and performance evaluation. *Constr. Build. Mater.* **2021**, *268*, 121082. [[CrossRef](#)]
61. Huang, C.-L.; Dun, J.-F. A distributed PSO–SVM hybrid system with feature selection and parameter optimization. *Appl. Soft Comput.* **2008**, *8*, 1381–1391. [[CrossRef](#)]
62. Eberhart-Phillips, D.; Chadwick, M. Three-dimensional attenuation model of the shallow Hikurangi subduction zone in the Raukumara Peninsula, New Zealand. *J. Geophys. Res. Solid Earth* **2002**, *107*, ESE 3-1–ESE 3-15. [[CrossRef](#)]
63. Ghorbani, N.; Kasaeian, A.; Toopshekan, A.; Bahrami, L.; Maghami, A. Optimizing a hybrid wind-PV-battery system using GA-PSO and MOPSO for reducing cost and increasing reliability. *Energy* **2018**, *154*, 581–591. [[CrossRef](#)]
64. Moayedi, H.; Mehrabi, M.; Mosallanezhad, M.; Rashid, A.S.A.; Pradhan, B. Modification of landslide susceptibility mapping using optimized PSO-ANN technique. *Eng. Comput.* **2019**, *35*, 967–984. [[CrossRef](#)]

65. Ge, H.-W.; Qian, F.; Liang, Y.-C.; Du, W.-L.; Wang, L. Identification and control of nonlinear systems by a dissimilation particle swarm optimization-based Elman neural network. *Nonlinear Anal. Real World Appl.* **2008**, *9*, 1345–1360. [[CrossRef](#)]
66. Li, H.R.; Gao, Y.L.; Li, J.M. A Particle Swarm Optimization Algorithm with the Strategy of Nonlinear Decreasing Inertia Weight. *J. Shangluo Univ.* **2007**, *21*, 16–20. [[CrossRef](#)]
67. Shi, Y.; Eberhart, R. A modified particle swarm optimizer. In Proceedings of the 1998 IEEE International Conference on Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence (Cat. No.98TH8360), Anchorage, AK, USA, 4–9 May 1998; pp. 69–73.
68. Menard, S. Coefficients of determination for multiple logistic regression analysis. *Am. Stat.* **2000**, *54*, 17–24. [[CrossRef](#)]
69. Le, L.M.; Ly, H.-B.; Pham, B.T.; Le, V.M.; Pham, T.A.; Nguyen, D.-H.; Tran, X.-T.; Le, T.-T. Hybrid artificial intelligence approaches for predicting buckling damage of steel columns under axial compression. *Materials* **2019**, *12*, 1670. [[CrossRef](#)] [[PubMed](#)]
70. Dao, D.V.; Trinh, S.H.; Ly, H.-B.; Pham, B.T. Prediction of compressive strength of geopolymer concrete using entirely steel slag aggregates: Novel hybrid artificial intelligence approaches. *Appl. Sci.* **2019**, *9*, 1113. [[CrossRef](#)]
71. Ly, H.-B.; Le, L.M.; Duong, H.T.; Nguyen, T.C.; Pham, T.A.; Le, T.-T.; Le, V.M.; Nguyen-Ngoc, L.; Pham, B.T. Hybrid artificial intelligence approaches for predicting critical buckling load of structural members under compression considering the influence of initial geometric imperfections. *Appl. Sci.* **2019**, *9*, 2258. [[CrossRef](#)]
72. Ly, H.-B.; Monteiro, E.; Le, T.-T.; Le, V.M.; Dal, M.; Regnier, G.; Pham, B.T. Prediction and sensitivity analysis of bubble dissolution time in 3D selective laser sintering using ensemble decision trees. *Materials* **2019**, *12*, 1544. [[CrossRef](#)]
73. Ly, H.-B.; Le, L.M.; Phi, L.V.; Phan, V.-H.; Tran, V.Q.; Pham, B.T.; Le, T.-T.; Derrible, S. Development of an AI model to measure traffic air pollution from multisensor and weather data. *Sensors* **2019**, *19*, 4941. [[CrossRef](#)]
74. Pham, B.T.; Jaafari, A.; Prakash, I.; Bui, D.T. A novel hybrid intelligent model of support vector machines and the MultiBoost ensemble for landslide susceptibility modeling. *Bull. Eng. Geol. Environ.* **2019**, *78*, 2865–2886. [[CrossRef](#)]
75. Iqbal, M.F.; Liu, Q.-f.; Azim, I.; Zhu, X.; Yang, J.; Javed, M.F.; Rauf, M. Prediction of mechanical properties of green concrete incorporating waste foundry sand based on gene expression programming. *J. Hazard. Mater.* **2020**, *384*, 121322. [[CrossRef](#)]