


Article

Artificial Intelligence Models for the Mass Loss of Copper-Based Alloys under Cavitation

Cristian Ștefan Dumitriu¹ and Alina Bărbulescu^{2,*} 

¹ Doctoral School, Technical University of Civil Engineering Bucharest, 124, Lacul Tei Bd., 020396 Bucharest, Romania

² Department of Civil Engineering, Transilvania University of Brașov, 5, Turnului Street, 900152 Brașov, Romania

* Correspondence: alina.barbulescu@unitbv.ro

Abstract: Cavitation is a physical process that produces different negative effects on the components working in conditions where it acts. One is the materials' mass loss by corrosion–erosion when it is introduced into fluids under cavitation. This research aims at modeling the mass variation of three samples (copper, brass, and bronze) in a cavitation field produced by ultrasound in water, using four artificial intelligence methods—SVR, GRNN, GEP, and RBF networks. Utilizing six goodness-of-fit indicators (R^2 , MAE, RMSE, MAPE, CV, correlation between the recorded and computed values), it is shown that the best results are provided by GRNN, followed by SVR. The novelty of the approach resides in the experimental data collection and analysis.

Keywords: mass loss; cavitation; ultrasound; corrosion–erosion; artificial intelligence (AI)



Citation: Dumitriu, C.Ș.; Bărbulescu, A. Artificial Intelligence Models for the Mass Loss of Copper-Based Alloys under Cavitation. *Materials* **2022**, *15*, 6695. <https://doi.org/10.3390/ma15196695>

Academic Editor: Mikhail Zheludkevich

Received: 11 August 2022

Accepted: 23 September 2022

Published: 27 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

An ultrasound signal passing through a liquid gives birth to cavitation, a phenomenon defined by the cyclic apparition, increase, and collapse of the bubbles of vapors formed in the liquid [1]. During these cycles, a voltage is induced near the cavitation zone's boundaries [2–4]. Cavitation is an exogenous process accompanied by discontinuities in the liquid's state, which appear when the pressure drops under critical limits [5]. Emulsification, noise, vibrations, sonoluminescence, unpassivation, and corrosion are some of its multiple effects [6,7].

Corrosion is a complex process affecting most of the metallic materials used in industry. Electrochemical corrosion occurs at the contact of a metal with an electrolyte (for example, seawater) by charge transfer through the interface [8]. Oliphant [9] classified the copper alloys' corrosion types into pitting, rosace corrosion, induced microbiological corrosion, erosion–corrosion, and corrosion induced by welding flux. Regardless of the corrosion type, the cavitation effects are determined by the material and environmental characteristics [10].

Analyzing and monitoring the process apparition and the damages produced on different installations and components (such as ballast installation, bilge systems, propellers) working in cavitation conditions are essential for ensuring their correct functioning and the workers' safety. Therefore, erosion–corrosion and materials' mass loss as effects produced by cavitation have become topics of interest for many scientists [11–15].

Several investigations on cavitation corrosion, mainly from the viewpoint of the driving mechanism, have been performed by different authors [16–18]. Wharton and Stokes [15] studied the corrosion process of bronzes with Ni and Al. Basumatary et al. [11] found that the Ni–Al bronze has a significantly better corrosion resistance than the duplex stainless steel in cavitation (both materials are usually used for building propellers). It was also shown [12,19] that the presence of Sn in different bronzes increases their resistance to high-pressure vapors of seawater. The alloys' mechanical properties were improved by

adding Ni in bronzes with Fe and Al. The resulting materials had an increased resistance to corrosion in cavitation media.

The results of Kumar et al. [20] show that the increase in the lead content of brass from 1 to 3.4 wt.% increases the corrosion resistance and improves the stability of the alloy in both chloride and sulfate media due to the passive film formation. The literature [21,22] specifies that finely processed surfaces are more resistant than those with high roughness. It was also shown that the damage of the materials introduced in a cavitation field is more intense than that in the same medium without cavitation [23].

It is known that there is a direct relationship between the material's resistance and its mass loss. The lower the resistance is, the higher the mass loss is, since the links between the atoms and molecules in the material can be easily broken when the resistance is low. Nevertheless, despite the extensive study of the corrosion mechanism, only a few researchers (Schüssler and Exner [13], Fortes-Patella et al. [24], Dumitriu [25], Dumitriu and Barbulescu [26]) have analyzed the copper-based alloy samples' mass loss in liquids under cavitation from a quantitative viewpoint. None of them [13,24–26] approached the subject with artificial intelligence tools (AI).

AI algorithms provide solutions to a large variety of problems. Approaches such as Support Vector Machine (SVR) [27], Artificial Neural Networks (ANNs) [28], General Regression Neural Networks (GRNN) [29], Gene Expression Programming (GEP) [30–32], adaptive neuro-fuzzy interference (ANFIS) [33], and decision tree (DT) [34] have been utilized in engineering and finance for solving optimization problems. Radial basis networks (RBFs), multilayer perceptrons (MLPs), artificial regression trees, and Random Forest were successfully employed to predict the mass loss in Electro-Discharge Machining [35]. The reliability of some components from different industrial case studies has been forecast by four artificial intelligence methods [36]. GEP, AdaBoost, and XGBoost were applied to evaluate the high-strength concrete properties in [37,38].

In the above context, the present article aims at modeling the mass loss of samples of brass, bronze, and copper (for comparison) in a cavitation field produced in an experimental setup built by our team for such purposes. Four AI algorithms are employed for this aim (SVM, GRNN, RBF, and GEP) and the best results (with respect to six indicators) are presented. It is shown that all algorithms performed well on the studied series, but the best was GRNN. In comparison with the classical parametric regression, this approach has the advantage that it does not rely on restrictive hypotheses on the data sets, and any condition does not constrain the residuals.

The novelty of the research resides in the following:

- The experimental setup was designed by our team.
- The experiments on materials mass loss have been performed in the cavitation field produced by ultrasound; this approach is important for knowing the mass-loss behavior of some materials used in naval construction.
- Based on our knowledge, the analysis of the mass loss of copper-based alloys in the ultrasound cavitation field was not extensively performed.
- Modeling the mass loss of such alloys in the cavitation field has not been performed using AI methods.
- Knowing the mass loss is important for predicting the behavior of different components built using such materials; a good model can be used to obtain a forecast that can be utilized for predicting the replacement periods in an integrated reliability study.

2. Materials and Methods

2.1. Experiments

The experimental plant built for performing the study of ultrasound cavitation and the materials' mass loss is presented in Figure 1.

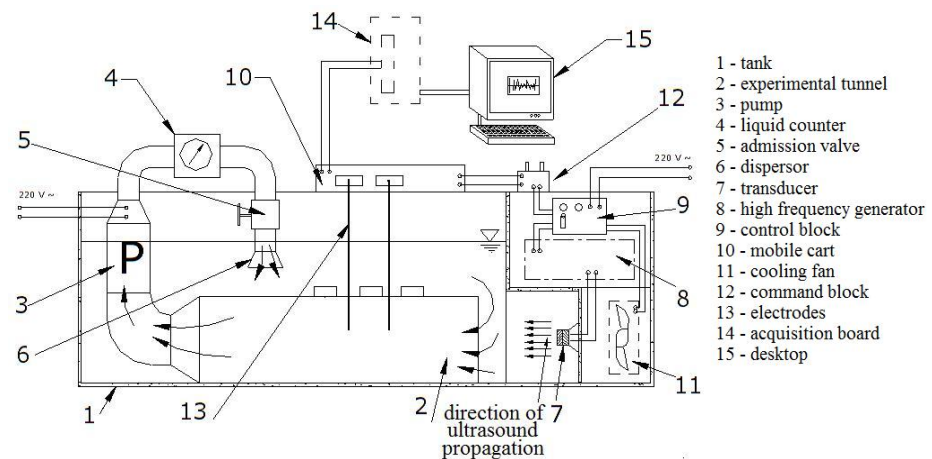


Figure 1. Experimental plant for the cavitation study [39].

Its main parts are as follows [3,39]:

- The tank (1) containing the liquid subject to the ultrasound cavitation;
- The high-frequency ultrasound generator (8), designed to work at 220 V, 18 kHz, and three power levels;
- The piezoceramic transducer (7) that produces cavitation entering into oscillation as a response to the high-frequency signal received from the generator;
- The control panel (command block) (12) from where the ultrasound generator's working power is selected;
- The cooler (11) that is used to maintain a constant temperature of the liquid;
- The measurement electrodes (13), which are utilized only in the experiments related to capturing the signal induced in the cavitation field;
- The acquisition data unit (14), used only in the experiments related to electrical signals induced by ultrasound cavitation to collect the signals.

For experiments in the circulating liquid medium—not discussed in this article—the pump (3) is switched on. The following conditions have been met in the experiments whose results are presented here.

The studied samples had the following compositions:

- Cu containing small percentages of Fe, Sn, and Zn (0.0395%, 0.0446%, and 0.0747%, respectively);
- A brass with 2.75% Pb, 38.45% Zn besides Cu (57.95 %);
- A bronze containing Zn, Pb, and Sn (4.07%, 4.40%, and 6.4%, respectively) besides Cu.

The samples had a hexagonal shape, with a side of about 1.5 cm. They were suspended by rigid plastic wires inside the tank at a distance of about 20 cm from the transducer.

The ultrasound generator worked at 180 W, and the water temperature was maintained at 20 °C. The samples were kept in saline water under cavitation produced by ultrasound for 1320 min, cleaned, and weighted every 20 min. The composition of the seawater used for all the experiments is the following: pH = 7, 22.17 g/L NaCl, 0.051 mg/L Fe, 0.0033 mg/L Ni, 0.31 g/L SO_4^{2-} , total water hardness—6.27 meq/L.

The experiments were performed in triplicate.

To estimate the ultrasound effect on the samples from a quantitative point of view, the mass variation (computed by the difference between the sample's mass at the experiment's beginning and the mass at the moment t) on the surface (S) was recorded for the modeling stage. The data series are represented in Figure 2.

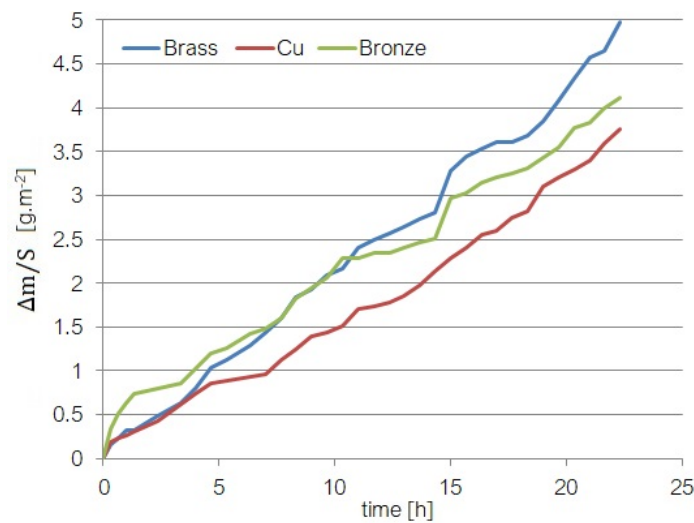


Figure 2. Data series.

The mass loss variation can be explained by a complex mechanism (electrochemical and erosion–corrosion). Since the material is introduced in seawater with a high concentration of NaCl, two opposite processes appear: (a) passivation leading to the formation of Cu oxides deposited on the sample surface; (b) removal of these oxides due to the ultrasound action. In the first phase, the sample mass increases, while in the second one, it decreases, explaining the oscillations observed in the chart.

Moreover, as a cavitation effect, the material’s microcrystalline structure is weakened by the erosion–corrosion that breaks the atomic links. Consequently, the material’s resistance decreases, resulting in microfissures and cracks. These fissures become new sites where the material is broken and detached from the sample. Therefore, the material mass loss is more accelerated when it is no longer maintained in the cavitation field, increasing the mass loss rate. The material loss is not uniform because it is composed of different elements with various structures and resistances to cavitation.

2.2. Modeling Methodology for the Weight Loss

The AI methods utilized in this work for modeling the mass loss of the materials in the described conditions are SVR, GRNN, GEP, and RBF.

SVR [40–42] belongs to the class of supervised learning algorithms characterized by high performances correlated with a low computational cost in solving regression and classification problems. Using training sets composed of n pairs (x_k, y_k) , where x_k ($k = \overline{1, n}$) are the vectors of known features from a domain D in the m -dimensional space, \mathbf{R}^m ($x_k \in D \subset \mathbf{R}^m$) and $y_k \in \mathbf{R}$ are the target values, the algorithm output is a function f based on which unknown \widehat{y}_k are provided when x_k are given. The model function f , in the linear case, is

$$f(x) = (u, x) + b, \quad u \in \mathbf{R}^m, \quad x \in D, \quad (1)$$

where (\cdot, \cdot) is the inner product in D , and b is a real constant.

SVR produces prediction functions mapped on a set $S \subset D$ of vectors support [27] and f should minimize the objective function

$$\frac{1}{2} \|u\|^2 + C \sum_{k=1}^n L(y_k, f(x_k)), \quad (2)$$

where $\| \cdot \|$ is the norm in L^2 , $C \in (0, \infty)$, $L(\cdot, \cdot)$ is a ε -insensitive loss function [40,41].

The maximum deviation between the target and the f function’s values must be less than ε on the training set. The number of support vectors is controlled by ε , whereas C ensures a balance between the model function’s flatness and the bias from the ε – tube [27,41].

The nonlinear problems are transformed into linear through some mappings, Ψ , defined using different kernels $K : D \times D \rightarrow \mathbf{R}$, where $K(u, v)$ is the inner product of $\Psi(u)$ and $\Psi(v)$ in $\Psi(D)$.

One of the most important aspects of SVR modeling is choosing the kernel type. The most used kernels are RBF, linear, sigmoid, and polynomial. Their choice significantly influences the model quality. Therefore, for our study, we performed experiments with all these types of kernels. Since the best results were obtained utilizing a linear kernel, we report them in the article.

Another aspect is related to the time necessary to perform the experiments. SVR is the worst compared with GEP, GRNN, and RBF. It takes a few minutes to run the algorithm, compared with a few seconds for the other three methods. The time consumed increases when the number of the kernel's parameters to be estimated increases and the step size in the grid search for each parameter decreases.

After choosing the linear kernel, the step size in the grid search was selected to be 0.1 (to maintain a balance between the time and parameters' quality) and the parameter C was searched in the interval $[0, 50,000]$.

The number of predictors should also be selected for performing the algorithm. Theoretically, the user should choose the number of predictors based on his experience or the series characteristics. After performing experiments with different predictors, and given that the mass loss depends on the mass value before running the experiment, the number of predictors was chosen to be one, which is the lag 1 variable (the sample's absolute mass loss at the previous moment).

For modeling, the data series was divided into two parts—the first for training and the second for testing. Different ratios were tested. Here, we report the best results obtained for the ratio training:test = 70:30 (70% of the data series for training the model and the rest for the testing). A 4-fold cross-validation was used in the training–prediction process. The optimization criterion was to minimize the total error.

For details of SVR, the reader may see the articles of Vapnik [40] and Smola [41].

GRNNs (Figure 3) are ANNs with four layers—input, hidden, summation, and output. These feedforward networks process the information in successive order from one layer to another without feedback [43].

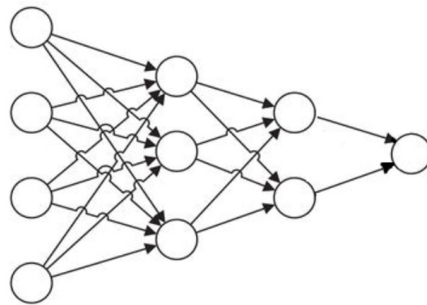


Figure 3. The scheme of a GRNN.

The neurons in the hidden layers symbolize different patterns. They calculate the distance (usually Euclidean) between each input data and a center point and apply an activation function to these distances.

Each training sample, X_i , is utilized as the mean of a Gaussian distribution

$$Y(X) = \frac{\sum_{i=1}^n Y_i \exp(-D_i^2 / (2\sigma^2))}{\sum_{i=1}^n \exp(-D_i^2 / (2\sigma^2))}, \quad (3)$$

where

$$D_i^2 = (X - X_i)^T (X - X_i) \quad (4)$$

and D_i is the distance between the training sample and the prediction point.

D_i is a measure of how well each training sample can represent the position of prediction, X . When D_i is small, $\exp(-D_i^2/(2\sigma^2))$ is big. When $D_i = 0$, $\exp(-D_i^2/(2\sigma^2)) = 1$, and the evaluation point is best represented by the training sample. A high D_i produces a small $\exp(-D_i^2/(2\sigma^2))$; as a consequence, the contribution of the other training samples to the prediction is relatively small. If σ is big, the possible representation of the evaluation point by the training sample is possible for a wider range of X , whereas when σ is small, the representation is limited to a narrow range of X [44]. Since the influence radius of each neuron is controlled by σ , this parameter must be found in the training process to optimize the network's performance.

The third layer is formed by the S- and D-summation neurons, which sum up the information received from the previous layer (weighted by the first one).

The GRNN is described by the equation [45]

$$Y = W_{ho}\varphi(XW_{ih} + b_I) + b_o \quad (5)$$

where the symbols have the following meanings:

X —the vector containing the network input;

Y —the output vector;

φ —the activation function;

W_{ih} —the matrix of the weights of the input in the hidden layer;

W_{ho} —the matrix containing the weights of the results of the hidden layer;

$b_I(b_o)$ —the error vector corresponding to the first (hidden layer).

In this study, to select the optimum $\sigma - s$, the employed algorithm was the conjugate gradient, and the search was performed in the interval [0.0001; 10]. The convergence tolerances were 10^{-8} —absolute and 10^{-4} —relative, and the upper limit of the number of iterations (without improvement, respectively) was 5000 (1000, respectively). As for SVR, the best results are reported here. They were obtained when running the algorithm with a ratio of 70:30 between the training and test sets.

RBF belongs to the feedforward ANNs and is built of three layers—input, hidden, and output. The first layer's output is obtained by computing the distance between its input and the second layer's centers. The hidden layer has as output the weighted values of the output of the first layer. The center is a vector of the hidden layer's neurons. An activation function (usually Gaussian) is associated with the neurons from the second layer. This function has a spread parameter that controls its behavior.

The output of an RBF network can be described by [46]

$$\hat{y}_k = \sum_{j=1}^J w_{jk}\varphi(\|\mathbf{x} - \mathbf{c}_j\|) + \beta_k, \quad j = \overline{1, J}, \quad k = \overline{1, K}, \quad (6)$$

with

\hat{y}_k —the k th neuron's output;

\mathbf{x} —the input data vector;

\mathbf{c}_j —the j th neuron center vector;

$J (K)$ —the number of neurons belonging to the second (third) layer;

w_{jk} —the weight corresponding to the j th and k th neuron and output;

$\|\cdot\|$ —the Euclidean norm;

β_k —the bias corresponding to the k th neuron's output;

φ —the RBF function:

$$\varphi(\|\mathbf{x} - \mathbf{c}_j\|) = \exp(-\alpha_j\|\mathbf{x} - \mathbf{c}_j\|^2); \quad (7)$$

α_j —the spread parameter (radius) that controls the j th neurons' spread.

For the network's performance evaluation, MSE or RMSE are usually utilized.

The RBF training is performed to minimize the objective function, which is the sum of square errors, defined by

$$SSE = \sum_{k=1}^K (y_k - \hat{y}_k)^2 \quad (8)$$

where y_k is the recorded value and \hat{y}_k is the result after running RBF [47].

In RBF networks, choosing the neurons' number in the hidden layer affects the network's complexity and its generalizing capability. If this number is insufficient, the network cannot learn the data adequately. If this number is very high, it may result in overfitting or limited generalization capacity [48,49].

It was also shown that the centers' detection (in the second layer) significantly influences the performance of the RBF network [49]. Therefore, one of the main tasks when training the network is determining the best center positions.

The RBF network training must also include the optimization of the spread parameters of each neuron and the selection of the weights between the second and third layers. Therefore, in the training process, the number of neurons, the centers in the second layer, the spread parameters, and the weights must be appropriately selected. To achieve this goal, the network's training was performed by an orthogonal forward algorithm proposed by Chen et al. [50] that employs tunable center vector nodes for building the RBF function and minimizes the leave-one-out procedure. The algorithm does not need to impose a stop criterion. The ridge regression was utilized for the weights' computation.

For tuning the neurons' parameters, the size of the population was fixed to 200, maximum number of generations—20, maximum generations flat—5, and maximum boosting tolerance— 10^{-4} . The used networks' parameters were maximum number of neurons—100, minimum (maximum) radius = 0.01 (400), and absolute tolerance— 10^{-6} .

The reader may refer to [47,51–53] for more details on RBF networks.

Genetic Algorithms (GAs) are evolutionary techniques based on the principle of Darwinian selection, operating on populations of individuals, successively created by choosing the best individuals based on their fitness. The selected individuals are combined using specific operations to give birth to new generations. The steps of this procedure are (a) random initialization of the population, (b) fitness assignment, (c) individuals selection, (d) crossover and mutation. The algorithm is run until the stop criterion is met [54]. Random mutations do not permit GAs to remain captured in a locally optimal region.

Introduced in 1988 by Koza [55], Genetic Programming (GP) belongs to the same class of evolutionary techniques [56] and differs from others by the information's representation as programs inspired by nature mechanisms [57]. GP provides the solution to the problem at hand, enabling the computer to search for it and deliver it in the form of parse trees whose nodes but the terminal ones contain operators. The terminal nodes contain constants and variables so the expressions can be quickly evaluated and evolved. For example, the tree in Figure 4 presents the parse tree that encodes the expression $2 + (7 * X) - (11 / \cos(Y))$.

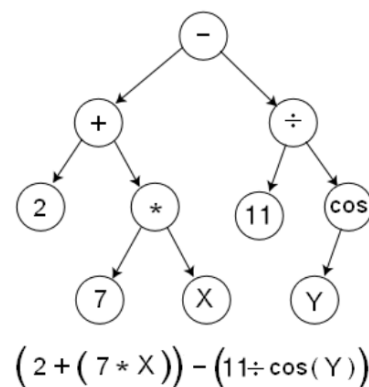


Figure 4. The parse tree and the encoded expression.

Gene Expression Programming (GEP), proposed by Ferreira in 2001 [58], incorporates features from GAs (such as linear chromosomes of fixed length) and GP (the structure of a tree with different shapes and sizes).

A chromosome is a string consisting of elements belonging to (a) and (b) that is mapped into a tree that has, as a correspondent, a unique mathematical formula [59]. It contains at least one gene, which is formed, at its turn, by several symbols (fixed) and has a head (that contains constants, variables, and functions) and a tail (containing terminal symbols—constants and variables). When a minimum of two genes are present, they are connected by a linking function to generate the solution to the problem at hand.

For solving a problem using GEP, one must specify (a) the set of functions, (b) the terminal set (that contains constants and variables), (c) the fitness function, (d) the control parameters, and (e) the stop condition.

To start, GEP randomly generates a population of chromosomes evaluated with respect to a fitness function defined by the user. Then, the best individuals are chosen, using the roulette-wheel selection criterion, for producing the next generation using genetic operations (mutation, transposition, crossover). Still, elitism is also applied: the individual with the highest fitness in each generation goes without modifications to the next generation. The number of genes and the linking function must be specified before running the algorithm. In all cases, the main point is that the solution to a problem is represented by individuals evolving and improving from one generation to the next. The process continues until the stop criterion is met.

It should be mentioned that the expression with the simplest form is preferred among the expressions with the same performances.

For details on GEP, the reader may see [58,60].

In the experiments, we performed 50 independent runs for each setup.

In this article, the following settings have been used to run the algorithm:

- Number of genes per chromosome—4.
- Length of the head of a gene—8.
- Number of constant per gene—10.
- The size of the population—50 individuals.
- The maximum number of generations (and without improvement)—2000 (1000).
- The fitness function—MSE, and the hit tolerance—0.01.
- The functions used to build the final expression—{+, −, *, /, sqrt}.
- The linking function was the addition.
- The algorithm was allowed to do algebraic simplification.
- The mutation (and inversion) rate—0.44 (0.1).
- The transposition rate—0.1.
- The one-point (two-point and gene) recombination rate—0.3 (0.3 and 0.1).
- For the experimental reasons explained above, the regressor in the model was the lag 1 variable, X_{t-1} , to predict X_t .
- The obtained models are compared with respect to the MSE values obtained on the original data set; in the following, we report only the best model (i.e., the model with the smallest mean standard error) found in all 50 runs of the algorithm.
- Experiments were performed using different ratios between the Training and Test sets, such as 80:20 or 90:10; the best results are presented in this article, which were obtained for the Training set formed by 70% of the data series, and the Test set formed by the rest of the series' values.

To compare the algorithms' performances, the following indicators were used: the proportion of variance explained by the model (R^2), the coefficient of variation (CV), the correlation between actual and predicted (r_{ap}), the mean absolute and mean absolute percentage error (MAE and MAPE), and the root mean squared error (RMSE).

The DTREG software [61] was employed for running the algorithms.

The flowchart of the study is presented in Figure 5.

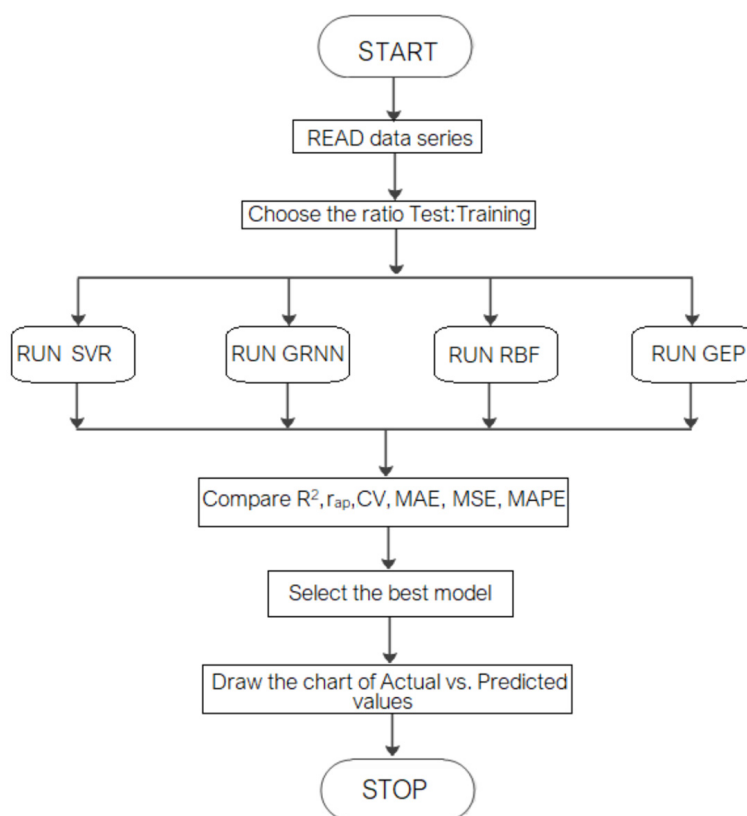


Figure 5. The flowchart of the modeling and estimation process.

3. Results

3.1. Results on the Cu Sample Mass Loss

The goodness of fit indicators for modeling the copper mass loss by the methods presented in Methodology are given in Table 1—on the Training (columns 2–5) and Test (columns 6–10) sets. The R^2 and r_{ap} are close to 1, indicating a very good concordance between the computed and recorded values on both sets. The highest values correspond to the GRNN algorithm on both sets.

Table 1. The goodness of fit test for the Cu sample mass loss in the cavitation field.

Set	Training				Test			
	SVR	GRNN	RBF	GEP	SVR	GRNN	RBF	GEP
R^2 (%)	99.290	99.943	99.315	99.339	98.923	99.206	99.166	99.206
CV	0.0500	0.0141	0.0491	0.0482	0.0161	0.0137	0.0141	0.0138
r_{ap}	0.9967	0.9997	0.9967	0.9967	0.9963	0.9962	0.9960	0.9962
RMSE	0.0528	0.0149	0.0519	0.0510	0.0479	0.0410	0.0422	0.0412
MAE	0.0422	0.0031	0.0453	0.0448	0.0402	0.0345	0.0350	0.0353
MAPE	6.5800	77×10^{-7}	7.3097	7.0305	1.3421	1.1612	1.1957	1.1935

The SVR with a linear kernel gave the best results. During search, the number of evaluated points was 191, the minim error = 0.002853, and the parameters' values—the tube-width $\epsilon = 0.00003787$ and $C = 0.15714469$. Twenty-two support vectors were utilized during the process. In GRNN, the optimal sigma corresponding to the regressor was 0.0006164, and 1156 evaluations were performed. In the RBF network, the number of neurons was found 2, and the radius was between 0.01 and 5.8808.

In the GEP model, the complexity of the model after simplification was 9, and the fitness function was evaluated 110,150 times. The expression generated was

$$y_t = 0.5094507y_{t-1}/(1 - y_{t-1}) + 0.0004684, t \geq 2 \quad (9)$$

Since the lower the coefficient of variation is, the better the fit is, the best algorithm is GRNN (CV = 0.0141 and CV = 0.0137 on the Training and Test set, respectively). The lowest MAPE (77×10^{-7} and 1.1612, respectively), RMSE, and MAE correspond to GRNN as well. The r_{ap} values are almost equal in all methods on the Test set, while on the Training one, the one corresponding to GRNN (0.9997) is higher than the others (equal to 0.9967).

The second best algorithm on the Training set is GEP, considering the first four indicators, and SVR, in terms of MAE and MAPE. On the Test set, the second-best algorithm is GEP, with respect to all indicators but MAE.

Table 2 contains the actual, computed values; the errors; and the absolute percentage error (denoted by % error) from the Test procedure. The % error on the Test set in GRNN is between 0.056 and 2.729 (the amplitude 2.673); in SVR, it is in the interval [0.085, 2.869] (amplitude = 2.784); in GEP, it is between 0.179 and 2.669 (amplitude = 2.490); and in RBF, in the range 0.098–3.154 (amplitude = 2.956).

Table 2. Actual and computed values in GRNN on the Test set on the Cu sample.

Actual	Computed	Error	% Error
2.2840	2.2734	0.0106	0.465
2.4002	2.4015	−0.0013	0.056
2.5550	2.5258	0.0292	1.145
2.5938	2.6500	−0.0562	2.169
2.7486	2.7756	−0.0270	0.982
2.8260	2.9031	−0.0771	2.729
3.0970	3.0329	0.0641	2.070
3.2131	3.1650	0.0481	1.500
3.2906	3.2994	−0.0088	0.269
3.4067	3.4363	−0.0296	0.869
3.6003	3.5757	0.0246	0.683
3.7551	3.7176	0.0375	0.999

Figure 6 displays the chart of the computed values in GRNN (named “Predicted target values”) vs. the recorded ones (named “Actual target values”). Since the dots are aligned along the line representing the perfect fit between the computed and recorded values, the chart confirms that the GRNN is the most appropriate to model the mass loss of the Cu sample.

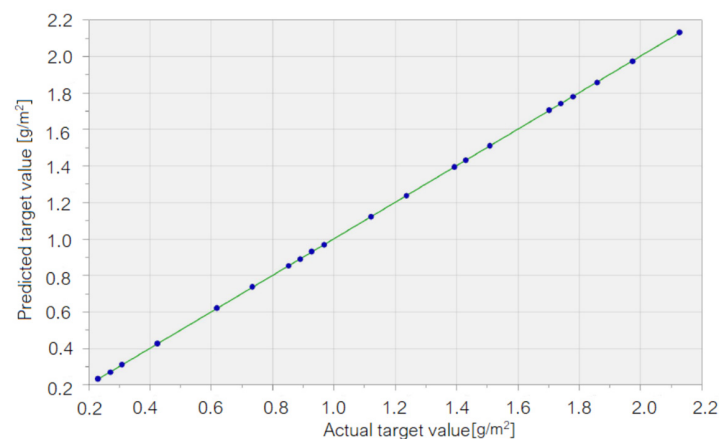


Figure 6. The computed values (called “predicted target values”) vs. the recorded ones (called “actual target values”) in the GRNN model for the Cu sample.

3.2. Results on the Brass Sample Mass Loss

The results of modeling the brass sample's mass loss by different AI methods are presented in Table 3. Columns 2–5 contain the results on the Training set, while the last four columns are filled in with the results on the Test set.

Table 3. The goodness of fit test for the brass sample mass loss in the cavitation field.

Set Indicator	Training				Test			
	SVR	GRNN	RBF	GEP	SVR	GRNN	RBF	GEP
R ² (%)	99.231	99.969	99.619	99.555	94.501	94.197	93.574	90.628
CV	0.0563	0.0112	0.0397	0.0429	0.0291	0.0299	0.0315	0.0380
r _{ap}	0.9964	0.9999	0.9981	0.9978	0.9773	0.9724	0.9715	0.9662
RMSE	0.0854	0.0170	0.0602	0.0651	0.1173	0.1205	0.1268	0.1532
MAE	0.0580	0.0048	0.0502	0.0546	0.1004	0.0981	0.1022	0.1240
MAPE	6.2502	0.2009	6.9911	6.9481	2.463	2.4912	2.6025	3.1965

When running the SVR algorithm, the best results were obtained with a linear kernel, and the parameters $C = 2.26564144$ and $\epsilon = 0.00001459$. During the search, 196 points were checked, 23 support vectors were utilized, and the minimum error computed was 0.007092. In GRNN, the optimum sigma was 0.00217, and 2652 evaluations were performed. In the RBF network, the number of neurons was found to be 4, and the radius was between 0.01 and 5.8804.

In the GEP model, the complexity of the model after simplification was 30, whereas the fitness function was estimated at 118,300 times. The expression generated was

$$y_t = -46.262975y_{t-1}^3 + y_{t-1}/(y_{t-1} + 2.9923668) + y_{t-1}, t \geq 2. \quad (10)$$

On the Training set, GRNN performed the best, having the lowest RMSE, MAE, MAPE, and CV. On the Test set, SVR performed the best based on all indicators but MAE. In terms of MAE, the best algorithm was GRNN.

A comparison between the output of the Training and Test sets show the following:

- Point of view of R², r_{ap}, RMSE, and MAE for all algorithms provided the best results on the Training sets;
- Point of view of CV, all but GRNN gave the best results on the Test sets;
- From the MAPE viewpoint, the best results were provided by GRNN on the Test set and by SVR on the Test set.

Table 4 contains the absolute percentage error in the AI models on the Test sets.

Table 4. Absolute % error in AI models on the Test set.

SVR	GRNN	RBF	GEP
2.368	3.458	2.621	3.012
1.676	1.757	1.119	0.745
0.573	0.134	0.522	1.394
3.012	3.831	4.461	5.676
4.474	5.537	6.106	7.506
3.797	4.986	5.482	6.951
1.234	2.463	2.884	4.348
0.986	0.244	0.600	2.040
2.932	1.723	1.420	0.013
1.366	0.149	0.117	1.535
4.676	3.520	3.296	1.942

The amplitudes (difference between the maximum and minimum % errors) are 4.103 (SVR), 5.403 (GRNN), 7.493 (GEP), and 5.506 (RBF), confirming the SVR and GRNN's performances.

Figure 7 presents the chart of computed values (“predicted target values”) vs. the “actual target values” (recorded) in the SVR and GRNN models on the Training set. Since the closer the points (whose coordinates are the pairs (recorded values, computed values)) are to the first bisector of the axes of coordinates (the green line), the better the fit is, one may remark that the best model is GRNN (Figure 7b).

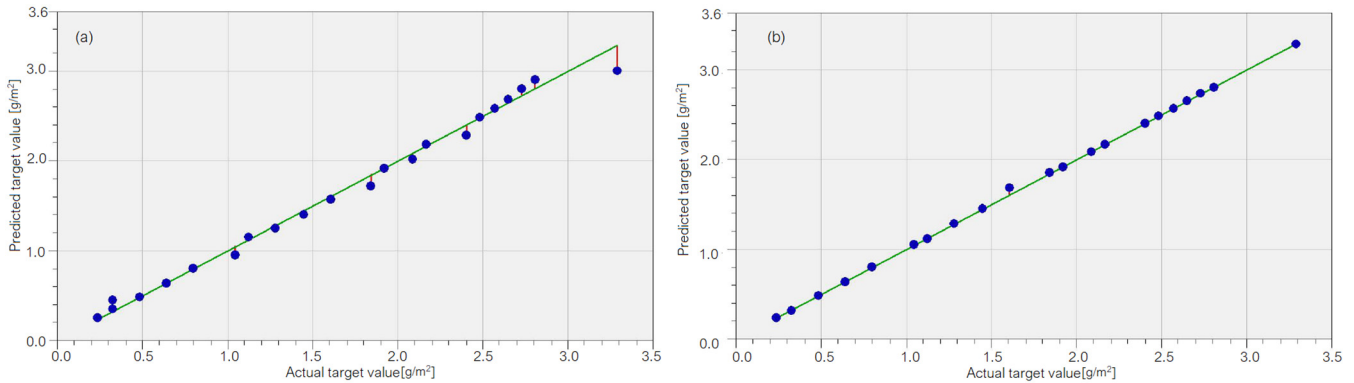


Figure 7. The computed values (called “predicted target values”) vs. the recorded ones (called “actual target values”) in (a) the SVR model and (b) the GRNN model for the brass sample.

Overall, the best model is GRNN because there are no significant mismatches between the parameters’ values on the Test set for both methods, but on the Training set, MAPE is much lower for GRNN than for SVR; overall, the best model is GRNN.

3.3. Results on the Bronze Sample Mass Loss

When running the SVR algorithm, the best results were obtained with a linear kernel, and the parameters $C = 1.1930123$ and $\epsilon = 0.04402791$. One hundred seventy points were checked when running the algorithm, the minimum error found was 0.005817, and thirteen support vectors were employed.

In GRNN, the optimum sigma was 0.1143176, and 850 evaluations were performed. In the RBF network, the number of neurons was found to be 2, and the radius was between 6.7531 and 9.0718.

In the GEP model, the complexity of the model after simplification was 13, and the number of fitness function evaluations was 114,400. The expression generated was

$$y_t = 6.028799y_{t-1}^4 + y_{t-1} - 9.6586234y_{t-1}^3, \quad t \geq 2. \tag{11}$$

Table 5 shows the indicators’ values after running the algorithms for the bronze.

Table 5. The goodness of fit test for the bronze sample mass loss in the cavitation field.

Set	Training				Test			
	SVR	GRNN	RBF	GEP	SVR	GRNN	RBF	GEP
R ² (%)	98.629	99.066	98.730	98.849	92.271	98.686	92.029	91.312
CV	0.0594	0.0490	0.0572	0.5746	0.0294	0.0220	0.0299	0.0312
r _{ap}	0.9932	0.9957	0.9938	0.9944	0.9768	0.9834	0.9795	0.9836
RMSE	0.0887	0.0732	0.0854	0.0813	0.1020	0.0762	0.1036	0.1081
MAE	0.0676	0.0426	0.0665	0.0613	0.0789	0.0601	0.0794	0.0873
MAPE	5.0594	2.4270	5.0261	4.0533	2.4427	1.8253	2.4556	2.6488

On both sets—Training and Test—the highest R² and r_{ap} and the lowest CV, RMSE, MAE, and MAPE are issued from the GRNN model, which proved to have the highest performances among the four algorithms. One may notice that the MAPEs for the GRNN are at least 1.670 (1.338) times higher than the corresponding MAPEs of the competitors on the Training (Test) set.

The absolute % errors on the Test set are also small—between 0.146 and 0.510. Table 6 contains the values of the errors and the absolute % errors in the GRNN model on the Test set. The values of the last parameter vary between 0.501 and 5.150.

Table 6. Recorded and computed values in the GRNN on the Test set for the bronze sample.

Recorded	Computed	Error	Absolute % Error
2.9724	2.8194	0.1530	5.150
3.0296	2.9639	0.0657	2.168
3.1439	3.1282	0.0157	0.501
3.2011	3.1495	0.0516	1.613
3.2582	3.2993	−0.0409	1.260
3.3154	3.4634	−0.1470	4.463
3.4297	3.4944	−0.0647	1.886
3.5441	3.6393	−0.0952	2.688
3.7727	3.8036	−0.0309	0.818
3.8298	3.8248	0.0050	0.133
4.0013	3.9745	0.0268	0.671
4.1156	4.1385	−0.0229	0.556

Figure 8 displays the recorded values (called “Actual”), those computed during the Training procedure (called “Predicted”), and those in the Test procedure (called “Validation”) in the GRNN algorithm. The shapes of the “Predicted” and “Validation” series are close to the “Actual” series, confirming the previous findings on the model quality.

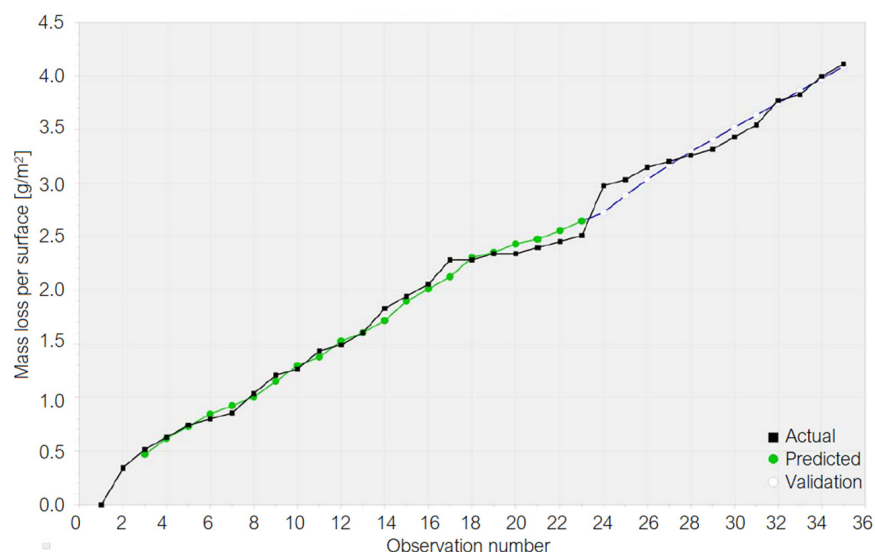


Figure 8. The series of the recorded values (called “Actual”), values computed in the Training procedure (called “Predicted”), and those calculated in the Test procedure (called “Validation”) in the GRNN model for the Cu sample.

4. Discussion

In [23,25,26], the authors proved that the power at which the ultrasound generator works is another factor affecting the mass loss, given that the cavitation effect is intensified at a higher power level compared with a lower one. Still, the best AI algorithm (with the same regressors) for modeling the mass loss in the experiments, performed when the generator worked at 80 W and 120 W, was GRNN. For the sake of brevity, the results are not reproduced here.

A complementary approach is to find the absolute mass loss variations as a function of time. The best nonlinear regression models obtained for the same data series are provided in the following.

The equation of the mass loss of the Cu sample is

$$\frac{\Delta m_t}{S} = 0.1684 + 0.1079t + 0.0023t^2 \quad (R^2 = 0.9968) \quad (12)$$

that for the brass sample is

$$\frac{\Delta m_t}{S} = 0.0732 + 0.1922t + 0.0008t^2 \quad (R^2 = 0.9957) \quad (13)$$

and that for the bronze sample is

$$\frac{\Delta m_t}{S} = 0.3384 + 0.1741t - 0.0004t^2 \quad (R^2 = 0.9910) \quad (14)$$

Despite that these models are easier to be obtained, and R^2 is close to one in all cases, their main drawback is that the residuals do not satisfy all the conditions (normality, homoscedasticity, the correlation absence) necessary to validate the model from a statistical viewpoint. Moreover, if a model is not validated, it cannot be used for forecasting for different reasons. For example, the residuals' autocorrelation leads to the errors' propagation and the increase in their amplitude. The heteroskedasticity may involve the existence of many subseries with different behaviors, making the forecast based on the entire series inconsistent.

An essential difference between the AI models and those from Equations (12)–(14) is that the regressor is the lag 1 variable in the first case and the time in last ones. Knowing that the mass loss at a certain moment depends on the sample mass and the experimental time, a natural idea is to extend the research by considering both regressors. Since this is an extensive work to be presented in detail in a future article, we introduce only the GRNN model for copper. Figure 9 shows the output of modeling the absolute mass loss of the copper sample, where "Actual", "Predicted", and "Validation" have the same significance as in Figure 8. "Forecast" represents the series of the future values computed based on the trained model.

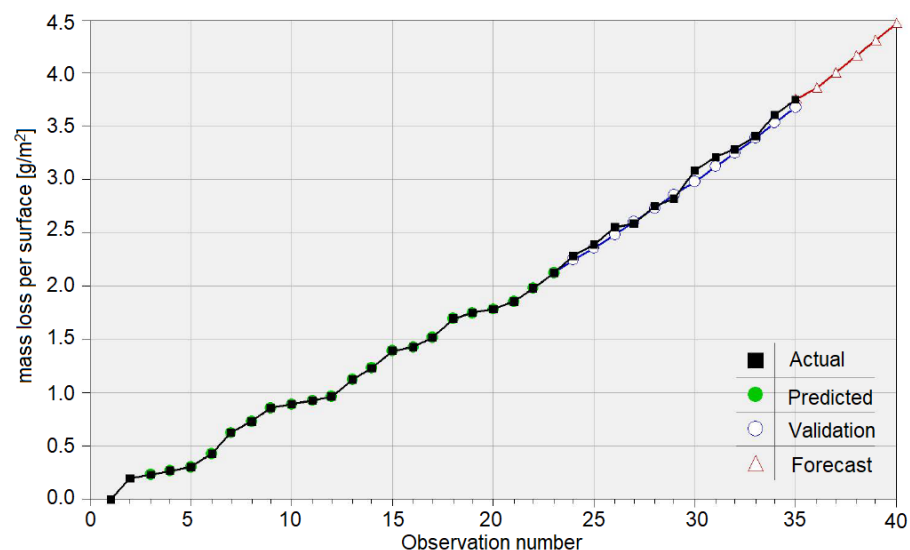


Figure 9. GRNN model with two regressors (lag 1 variable and time) for the mass loss per surface of the Cu sample.

The goodness-of-fit indicators in the model are as follows, respectively:

- On the Training set: $R^2 = 99.935\%$, $CV = 0.0151$, $MAE = 0.0033$, $RMSE = 0.016$, $r_{ap} = 0.99972$, and $MAPE = 10^{-8}$,

- On the Test set: $R^2 = 98.369\%$, $CV = 0.0198$, $MAE=0.0504$, $RMSE= 0.0590$, $r_{ap} = 0.9964$, and $MAPE = 1.6719$.

The model with two regressors is better than that with one regressor on the Training (Test) set point of view of all indicators (but R^2). The absolute % error in the Test set varies between 0.403 and 3.468; so, the amplitude is smaller than in the model with one regressor.

The gain of using both regressors should be also evaluated against the use of only one regressor (lag 1 variable) to determine the essential influence factor.

5. Conclusions

The paper presents the finding on the absolute mass loss per surface of Cu, brass, and bronze samples in a cavitation field produced by a high-frequency generator in seawater. It was shown that all four artificial intelligence approaches provide good modeling results for all the samples, but the best in terms of the majority of the goodness of fit indicators proved to be GRNN for Cu and bronze, followed by SVR for brass.

For the future research, four directions are considered. The first one is modeling the mass loss in time and per surface in the same liquid (and other liquids) in the cavitation absence and to compare the results of the present study with the new ones. The second direction is to model the mass loss of the material using multiple regressors, such as, for example, the concentrations of the elements in the samples, time, and the environmental conditions (temperature, seawater concentration). The third direction is to develop a model that relates the material mass loss to the voltage induced by cavitation at different generator's power level. The fourth one is to model the dependence between the material resistance and the mass loss during the experimental stages. The results of these studies will contribute to a better understanding of the corrosion–erosion process of the studied materials in cavitation field.

Author Contributions: Conceptualization, C.Ş.D. and A.B.; methodology, C.Ş.D. and A.B.; software, C.Ş.D.; validation, A.B.; formal analysis, C.Ş.D. and A.B.; investigation, C.Ş.D. and A.B.; resources, C.Ş.D. and A.B.; data curation, A.B.; writing—original draft preparation, A.B.; writing—review and editing, C.Ş.D.; visualization, C.Ş.D.; supervision, A.B.; project administration, A.B.; funding acquisition, A.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Flynn, H.G. Physics of acoustic cavitation in liquids. In *Physical Acoustics*; Mason, W.P., Ed.; Academic Press: New York, NY, USA, 1964; Volume 1, Part B, pp. 57–172.
2. Bărbulescu, A. Models of the voltage induced by cavitation in hydrocarbons. *Acta Phys. Pol. B* **2006**, *37*, 2919–2931.
3. Bărbulescu, A.; Dumitriu, C.Ş. Modeling the Voltage Produced by Ultrasound in Seawater by Stochastic and Artificial Intelligence Methods. *Sensors* **2022**, *22*, 1089. [[CrossRef](#)] [[PubMed](#)]
4. Bărbulescu, A.; Dumitriu, C.S. ARIMA and Wavelet-ARIMA Models for the Signal Produced by Ultrasound in Diesel. In Proceedings of the 25th International Conference on System Theory, Control and Computing (ICSTCC 2021), Iasi, Romania, 20–23 October 2021. [[CrossRef](#)]
5. Bai, L.; Yan, J.; Zeng, Z.; Ma, Y. Cavitation in thin liquid layer: A review. *Ultrason. Sonochem.* **2020**, *66*, 105092. [[CrossRef](#)] [[PubMed](#)]
6. Young, F.E. *Cavitation*; Mac Graw-Hill: Maidenhead, UK, 1989.
7. Rooney, J.A. *Ultrasound: Its Chemical, Physical and Biological Effects*, Suslick; VCH: New York, NY, USA, 1988.
8. Dumitriu, C.Ş. On the copper-based materials corrosion. In *Physics Studies*; Emek, M., Ed.; IKSAD Publishing House: Ankara, Turkey, 2021; pp. 67–100.
9. Oliphant, R.J. *Causes of Copper Corrosion in Plumbing Systems. Foundation for Water Research*; Allen House: Marlow, UK, 2003.
10. Simionov, M. Studies and Research on the Cavitation Destruction of Cylinder Liners from Diesel Engines. Ph.D. Thesis, Dunarea de Jos University of Galati, Galati, Romania, 1997.

11. Basumatary, J.; Nie, M.; Wood, J.K. The synergistic effects of cavitation erosion-corrosion in ship propeller materials. *J. Bio-Tribo-Corros.* **2015**, *1*, 12. [[CrossRef](#)]
12. Basumatary, J.; Wood, R.J.K. Synergistic effects of cavitation erosion and corrosion for nickel aluminium bronze with oxide film in 3.5% NaCl solution. *Wear* **2017**, *376–377*, 1286–1297. [[CrossRef](#)]
13. Schüssler, A.; Exner, H.E. The corrosion of nickel-aluminium bronzes in seawater—I. Protective layer formation and the passivation mechanism. *Corros. Sci.* **1993**, *3*, 1793–1802. [[CrossRef](#)]
14. Wharton, J.A.; Barik, R.C.; Kear, G.; Wood, R.J.K.; Stokes, K.R.; Walsh, F.C. The corrosion of nickel-aluminium bronze in seawater. *Corros. Sci.* **2005**, *47*, 3336–3367. [[CrossRef](#)]
15. Wharton, J.A.; Stokes, K.R. The influence of nickel–aluminium bronze microstructure and crevice solution on the initiation of crevice corrosion. *Electrochim. Acta* **2008**, *53*, 2463–2473. [[CrossRef](#)]
16. Bakhshandeh, H.R.; Allahkaram, S.R.; Zabihi, A.H. An investigation on cavitation-corrosion behavior of Ni/β-SiC nanocomposite coatings under ultrasonic field. *Ultrason. Sonochem.* **2019**, *56*, 229–239. [[CrossRef](#)]
17. Bărbulescu, A.; Orac, L. Corrosion analysis and models for some composites behavior in saline media. *Int. J. Energy Environ.* **2008**, *1*, 35–44.
18. Peng, S.; Xu, J.; Li, Z.; Jiang, S.; Xie, Z.-H.; Munroe, P. Electrochemical noise analysis of cavitation erosion corrosion resistance of NbC nanocrystalline coating in a 3.5 wt% NaCl solution. *Surf. Coat. Technol.* **2021**, *415*, 127133. [[CrossRef](#)]
19. Ivanov, I.V. *Corrosion Resistant Materials in Food Industry*; Editura Agro-Silvica: Bucharest, Romania, 1959. (In Romanian)
20. Kumar, S.; Narayanan, T.S.N.S.; Manimaran, A.; Kumar, M.S. Effect of lead content on the dezincification behaviour of leaded brass in neutral and acidified 3.5% NaCl solution. *Mater. Chem. Phys.* **2007**, *10*, 134–141. [[CrossRef](#)]
21. Hagen, C.M.H.; Hognestad, A.; Knudsen, O.Ø.; Sørby, K. The effect of surface roughness on corrosion resistance of machined and epoxy coated steel. *Prog. Org. Coat.* **2019**, *130*, 17–23. [[CrossRef](#)]
22. Okada, T. *Corrosive Liquid Effects on Cavitation Erosion*, Reprint UMICH No. 014456-52-1; University of Michigan: Ann Arbor, MI, USA, 1979.
23. Bărbulescu, A.; Dumitriu, C.S. Models of the mass loss of some copper alloys. *Chem. Bull. Politehnica Univ. (Timisoara)* **2007**, *52*, 120–123.
24. Fortes-Patella, R.; Choffat, T.; Reboud, J.L.; Archer, A. Mass loss simulation in cavitation erosion: Fatigue criterion approach. *Wear* **2013**, *300*, 205–215. [[CrossRef](#)]
25. Dumitriu, C.S. On the corrosion of two types of bronzes under cavitation. *Ann. Dunarea Jos Univ. of Galati Fasc. IX Metall. Mater. Sci.* **2021**, *4*, 12–16. [[CrossRef](#)]
26. Dumitriu, C.S.; Bărbulescu, A. Copper corrosion in ultrasound cavitation field. *Ann. Dunarea Jos Univ. of Galati Fasc. IX Metall. Mater. Sci.* **2021**, *3*, 31–35. [[CrossRef](#)]
27. Simian, D.; Stoica, F.; Bărbulescu, A. Automatic Optimized Support Vector Regression for Financial Data Prediction. *Neural Comput. Appl.* **2020**, *32*, 2383–2396. [[CrossRef](#)]
28. Uysal, M.; Tanyildizi, H. Estimation of compressive strength of self compacting concrete containing polypropylene fiber and mineral additives exposed to high temperature using artificial neural network. *Constr. Build. Mater.* **2012**, *27*, 404–414. [[CrossRef](#)]
29. Bărbulescu, A.; Barbes, L. Modeling the outlet temperature in heat exchangers. Case study. *Thermal Sci.* **2021**, *25*, 591–602. [[CrossRef](#)]
30. Javed, M.F.; Amin, M.N.; Shah, M.I.; 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](#)]
31. Farooq, F.; Akbar, A.; Khushnood, R.A.; Muhammad, W.L.B.; Rehman, S.K.U.; Javed, M.F. Experimental investigation of hybrid carbon nanotubes and graphite nanoplatelets on rheology, shrinkage, mechanical, and microstructure of SCCM. *Materials* **2020**, *13*, 230. [[CrossRef](#)] [[PubMed](#)]
32. Bărbulescu, A.; Șerban, C.; Caramihai, S. Assessing the soil pollution using a genetic algorithm. *Rom. J. Phys.* **2021**, *66*, 806.
33. Vakhshouri, B.; Nejadi, S. Prediction of compressive strength of self-compacting concrete by ANFIS models. *Neurocomputing* **2018**, *280*, 13–22. [[CrossRef](#)]
34. Bărbulescu, A.; Dani, A. Statistical analysis and classification of the water parameters of Beas River (India). *Rom. Rep. Phys.* **2019**, *71*, 716.
35. Bustillo, A.; Pimenov, D.Y.; Matuszewski, M.; Mikolajczyk, T. Using artificial intelligence models for the prediction of surface wear based on surface isotropy levels. *Robot. Comput.-Integr. Manuf.* **2018**, *53*, 215–227. [[CrossRef](#)]
36. Alsina, E.F.; Chica, M.; Trawiński, K.; Regattieri, A. On the use of machine learning methods to predict component reliability from data-driven industrial case studies. *Int. J. Adv. Manuf. Technol.* **2018**, *94*, 2419–2433. [[CrossRef](#)]
37. Aslam, F.; Furqan, F.; Amin, M.N.; Khan, K.; Waheed, A.; Akbar, A.; Javed, M.F.; Alyousef, R.; Alabduljabbar, H. Applications of Gene Expression Programming for Estimating Compressive Strength of High-Strength Concrete. *Adv. Civil Eng.* **2020**, *2020*, 8850535. [[CrossRef](#)]
38. Shen, Z.; Deifalla, A.F.; Kaminski, P.; Dyczko, A. Compressive Strength Evaluation of Ultra-High-Strength Concrete by Machine Learning. *Materials* **2022**, *15*, 3523. [[CrossRef](#)]
39. Bărbulescu, A.; Mârza, V.; Dumitriu, C.S. Installation and Method for Measuring and Determining the Effects Produced by Cavitation in Ultrasound Field in Stationary and Circulating Media. Romanian Patent No. RO 123086-B1, 30 April 2010.
40. Vapnik, V. *The Nature of Statistical Learning Theory*; Springer: Berlin, Germany, 1995.

41. Smola, A.J.; Scholkopf, B. A tutorial on support vector regression. *Stat. Comput.* **2004**, *14*, 199–222. [[CrossRef](#)]
42. Basak, D.; Pal, S.; Patranabis, D.C. Support vector regression. *Neural Inf. Process. Lett. Rev.* **2007**, *11*, 203–224.
43. Specht, D.F. A General Regression Neural Network. *IEEE Trans. Neural Netw.* **1991**, *2*, 568–578. [[CrossRef](#)] [[PubMed](#)]
44. Bauer, M.M. 2 General Regression Neural Network (GRNN). Available online: <https://minds.wisconsin.edu/bitstream/handle/1793/7779/ch2.pdf?sequence%3D14> (accessed on 26 July 2022).
45. Al-Mahasneh, A.J.; Anavatti, S.; Pratama, M.G.M. Applications of General Regression Neural Networks in Dynamic Systems. In *Digital Systems*; Asadpour, V., Ed.; IntechOpen: London, UK, 2018.
46. Howlett, R.J.; Jain, L.C. *Radial Basis Function Networks 2: New Advances in Design*; Physica-Verlag: Heidelberg, Germany, 2001.
47. Kurban, T.; Beşdok, E. A Comparison of RBF Neural Network Training Algorithms for Inertial Sensor Based Terrain Classification. *Sensors* **2009**, *9*, 6312–6329. [[CrossRef](#)] [[PubMed](#)]
48. Liu, Y.; Zheng, Q.; Shi, Z.; Chen, J. Training radial basis function networks with particle swarms. *Lect. Note Comput. Sci.* **2004**, *3173*, 317–322.
49. Simon, D. Training radial basis neural networks with the extended Kalman filter. *Neurocomputing* **2002**, *48*, 455–475. [[CrossRef](#)]
50. Chen, S.; Hong, X.; Harris, C.J. Orthogonal Forward Selection for Constructing the Radial Basis Function Network with Tunable Nodes. In *Advances in Intelligent Computing. ICIC 2005. Lecture Notes in Computer Science*; Huang, D.S., Zhang, X.P., Huang, G.B., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3644, pp. 777–786.
51. Fernández-Redondo, M.; Hernández-Espinosa, C.; Ortiz-Gómez, M.; Torres-Sospedra, J. Training Radial Basis Functions by Gradient Descent. In *Artificial Intelligence and Soft Computing—ICAISC 2004. Lecture Notes in Computer Science*; Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; Volume 3070, pp. 184–189.
52. Karayiannis, N.B. Reformulated radial basis neural networks trained by gradient descent. *IEEE Trans. Neural Netw.* **1999**, *3*, 2230–2235. [[CrossRef](#)] [[PubMed](#)]
53. Orr, M.J.L. Introduction to Radial Basis Function Networks. 1966. Available online: <https://faculty.cc.gatech.edu/~jbell/tutorials/rbf-intro.pdf> (accessed on 6 December 2021).
54. Genetic Algorithms for Feature Selection. Available online: https://www.neuraldesigner.com/blog/genetic_algorithms_for_feature_selection (accessed on 26 July 2022).
55. Koza, J.R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*; MIT Press: Cambridge, MA, USA, 1992.
56. Cheng, R. *Genetic Algorithms and Engineering Design*; Wiley: Hoboken, NJ, USA, 2007.
57. Banzhaf, W.; Nordin, P.; Keller, R.; Francone, F.D. *Genetic Programming—An Introduction; On the Automatic Evolution of Computer Programs and Its Applications*; Morgan Kaufmann: San Francisco, CA, USA, 1998.
58. Ferreira, C. Gene Expression Programming: A New Adaptive Algorithm for Solving Problems. *Complex Syst.* **2001**, *13*, 85–129.
59. Zhang, Q.; Zhou, C.; Xiao, W.; Nelson, P.C. Improving Gene Expression Programming Performance by Using Differential Evolution. Available online: <https://www.cs.uic.edu/~qzhang/Zhang-GEP.pdf> (accessed on 26 July 2022).
60. Ferreira, C. *Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2006.
61. DTREG. Available online: <https://www.dtreg.com/> (accessed on 6 August 2022).