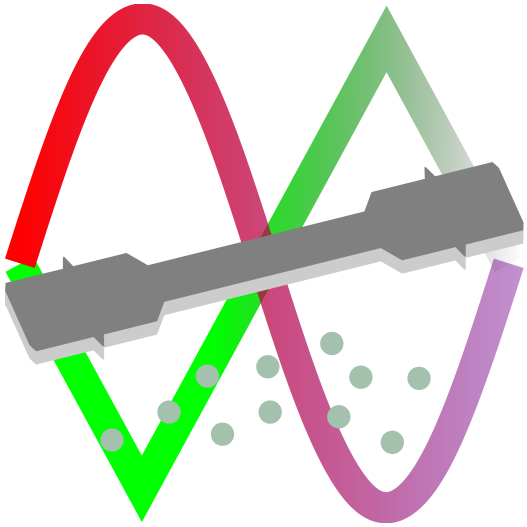
The Fatigue Life Predictor for SLIMMER (FLiPS) User Guide



Chris Hulme-Smith

University of Cambridge

March 30, 2017

Contents

1 How the Program Works

The Fatigue Life Predictor for SLIMMER (FLiPS) is a graphical user interface (GUI) for three independent *committees* of *artificial neural networks* (ANNs). Artificial neural networks are a powerful and highly adaptable form of model that may be fitted to source data to allow predictions to be made. FLiPS was made using over 15,000 datasets from the JApanese NAtional Research Institute for MEtals (NRIM).

Artificial neural networks are so named as they mimic (crudely and qualitatively) the human brain. All inputs are repeatedly combined to form a series of *hidden units*, which are then combined to produce the output. This is shown schematically in figure 2. Algebraically, the ANN can be summarised by equations 1–3, where x_i is the ith input variable, w_{ij} is the weighting parameter that relates all n inputs to the jth hidden unit, h_j; θ_j is an offset associated with the jth hidden unit; w_i is the weighting factor that relates the jth hidden unit to the output, y; p is the number of hidden units and $theta_y$ is an offset related to the output parameter.

$$h_j = \theta_j + \sum_{i=1}^{i=n} w_{ij} x_i \tag{1}$$

$$y = \theta_y + \sum_{j=1}^{j=p} w_j \tanh\left(h_j\right) \tag{2}$$

$$y = \theta_y + \sum_{j=1}^{j=p} w_j \tanh\left(\theta_j + \sum_{i=1}^{i=n} w_{ij} x_i\right) \tag{3}$$

The weighting parameters and offsets are fitted to source data. The use of the hyperbolic tangent (tanh function allows a very flexible fit and so virtually any function may be approximated using an artificial neural network. Since there are a large number of fitting parameters, it is very possible that the fit is erroneous. On the other hand, the fact that the fit is non-physical means that no knowledge of the physical relationship between inputs and outputs is required. To mitigate the possibility of an erroneous fit, FLiPS utilises a *committee* approach: multiple networks are produced from different starting points and fitted independently. This gives several independent predictions. The mean of all these predictions is presented as the prediction.

The predictive power of the network depends strongly on both the number of variables used to produce a fit, how well the model is able to reproduce data (i.e. how good the model is) and how many data there are with similar

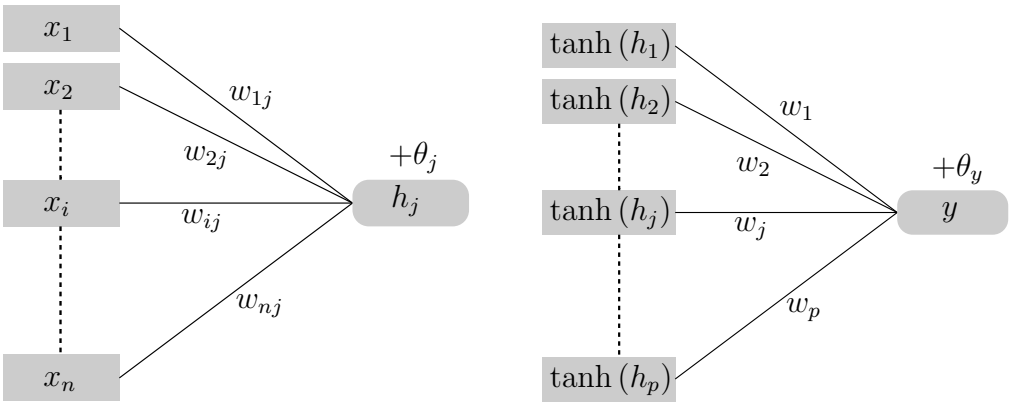


Figure 2: Schematic representation of an artificial neural network. The ith input is multiplied by a weighting factor, w_{ij}. All n products are then summed together with an offset, θ_j, to produce the jth hidden unit. All p hidden units are then multiplied by new weighting faactors, w_i and all products are added together with a final offset, θ_y to produce the output, y.

values to the input variables. To avoid overfitting, only a fraction of the available data is used (FLiPS uses 50%, called the *training* data). Another subset of the data is used to test the fit and is not used to refine the variables. This subset (30% in FLiPS) is the *test* data. The remaining 20% of data, termed the *validation* dataset, is held back to assess the predictive ability of the whole model.

The disagreement between the actual values, z_i and the predicted values, y_i of the *test* data is used to provide a measurement of the model's fit, called the *test error*, E_{test} (equation 4). The lower the test error, the more closely the model fits the observed data. However, this does not necessarily imply predictive ability, as the good performance of the model may only be valid for a very limited set of inputs. Test error also does not account for the certainty of predictions of the model. FLiPS uses Bayes' theorem to assess the confidence of the prediction. In general, the more source data there are, the greater the confidence in the prediction. Discrepancies between predictions and actual values are less significant if the model is less able to make predictions for the given input data. In other words, when there are few source data, the model is penalised less for any disagreement between calculated and observed values, since it is not reasonable to expect as strong a prediction than if there are many source data. The confidence in predictions may be accounted for using a quantity called the *variance* of the model, σ, which is dependent on the inputs used to produce a prediction. This is not

a true variance, but is analogous to variance in statistical distribution as it represents the range within which the true prediction will lie 66.7% of the time. Doubling this uncertainty gives the range in which the prediction will lie 95% of the time, tripling it 99% of the time, etc. A measurement of the model's predictive ability which includes this variance is the *log predictive error*, E_{lp} (equation 5).

$$E_{\mathrm{test}} = \sum^{m} (z_m - y_m)^2 \tag{4}$$

$$E_{\mathrm{lp}} = \sum^{m} \left(\frac{(z_m - y_m)^2}{2\sigma_m^2} + \log\left(\sigma_m \sqrt{2\pi}\right) \right) \tag{5}$$

As inferred previously, FLiPS uses a *committee* approach to increase the confidence and accuracy of predictions. Each model that is made is assessed using both E_{test} and E_{lp}. Combinations (or *committees* of the best performing models are tried and E_{test} and E_{lp} are calculated for the combinations. The best performing *committee* is selected to form the final model and all weighting parameters and offsets are re-refined for this committee.

FLiPS was produced using the Neuromat Model Manager [1] software package, which in turn uses the Bigback5 ANN refinement procedure [2]. Bigback5 uses a technique called *backpropagation* to derive the uncertainty in the prediction. It allows up to 25 models to be created, with up to 9 different starting values. This gives a maximum of $25 \times 9 = 255$ predictions. The 25 best performing individual models were then averaged in various combinations to find the best-performing committee.

In order to account for the varying magnitudes in input values which are all treated in same manner during (e.g. yield and ultimate tensile strength are of the order of 10^3 MPa, but stress concentration factors are of the order of 1), all inputs and outputs are normalised according to equation 6, where x_{min} and x_{max} are the minimum and maximum values of the variable in the input data, such that all values lie in the range $-\frac{1}{2} \leq x_{\mathrm{norm}} \leq \frac{1}{2}$. The refinement is carried out using normalised values and the prediction provided by each model is itself normalised using the fatigue lives in the input data. Normalised values may be unnormalised using equation 7, which is rearranged from equation 6.

Since fatigue lives range from hundreds of cycles to billions (10^2–10^9), it is extremely difficult for a model to predict small values with any accuracy, as an uncertainty of 10^8 cycles is relatively small for a prediction of 10^9, but astronomical for a prediction of, say, one million (10^6) cycles. To mitigate this, all fatigue lives in the source data were converted to (decadic)

logarithms before training the models. Predictions are, correspondingly, the logarithms of fatigue life, although FLiPS converts these to actual fatigue lives before displaying them to the user. One drawback of this approach is that uncertainties, which are $\pm$ for the logarithm of fatigue life generated by the model are effectively uncertainty factors, i.e. the fatigue life prediction must be multiplied and divided by the uncertainty to determine the range in which the true prediction could lie.

$$x_{\text{norm}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} - \frac{1}{2} \tag{6}$$

$$x = \left(x_{\text{norm}} + \frac{1}{2} \right) \times (x_{\text{max}} - x_{\text{min}}) + x_{\text{min}} \tag{7}$$

2 Starting the Program

FLiPS was developed using Qt versions 5.7 and 5.8 for MacOS. Once built, it was deployed for Mac with static libraries, meaning that all required resources should be included with the FLiPS application (CTRL+click and select "Show Package Contents" to see the resources included with the application). FLiPS will also be deployed for Windows, Linux/Unix, Android and iOS.

The FLiPS application is provided ready-to-use. In other words, no installation is needed and any special dependencies are bundled within the app itself.

2.1 Mac

Assuming that the deployment has been completed properly, the application you receive should open simply by either double-clicking on the executable (figure 3) or, equivalently, by moving it into Applications (or a subdirectory therein) and clicking on it. All required libraries are contained within the *framework* of the app, so no special environment is required.

2.2 Windows

FLiPS is not yet deployed on Windows, so no testing has been done. However, once deployed, the application should work by double-clicking on the executable, or by adding it to Program Files (or a subfolder therein) and clicking on it in the Start Menu or tiled desktop. All dependencies and resources are bundled within the application.

2.3 Linux/Unix

FLiPS is not yet deployed on Linux or Unix, so no testing has been done. However, once deployed, the application should work by running the executable from the terminal, by double-clicking on the executable in the file browser (e.g. Nautilus) or by selecting it from the application launcher in the particular Linux flavour you are running. All dependencies and resources are bundled within the application.

2.4 iOS (iPhone)

FLiPS is not yet deployed on iOS, so no testing has been done. However, once deployed, the application should work by clicking on its icon. All required dependencies and resources are bundles within the application.

2.5 Android

FLiPS is not yet deployed on Android, so no testing has been done. However, once deployed, the application should work by clicking on its icon. All required dependencies and resources are bundles within the application.

3 Making a Prediction

FLiPS has been created to allow a prediction to be made very easily.

3.1 How to make a prediction

FLiPS has two tables: one for the mechanical properties of the material and another for the test conditions (figure 4). Simple entering values for each input and pressing the "Predict" button gives predictions for all three loading geometries (uniaxial, bending and torsional). Pressing "Reset" empties the tables and predictions. "Quit" safely and efficiently closes the FLiPS application.

3.2 Radio buttons

The radio buttons at the top of the window allow the user to specify different units for some inputs.

3.2.1 Temperature

Temperature may be expressed using the Celsius, Kelvin or Farenheit scales, with the appropriate well-known conversion performed when the user request a prediction to convert the temperature to Celsius.

3.2.2 Stress type

The stress may be specified as one of the following:

- Stress amplitude ($\sigma_{\max} - \sigma_{\text{mean}}$)
- Stress range ($\sigma_{\max} - \sigma_{\min}$)
- Any of the minimum, mean or maximum stress — together with the R factor ($\frac{\sigma_{\min}}{\sigma_{\max}}$), the stress state is completely defined

3.2.3 Toughness

Toughness may be expressed as a Charpy impact energy in $\mathrm{J\,cm^{-2}}$ or as a fracture toughness in $\mathrm{MPa\,m^{1/2}}$. If fracture toughness, C_{fract}, is selected, the Ralf-Novak-Barsom correlation (equation 8) is used to convert to Charpy toughness, C_{Charpy}.

$$C_{\text{Charpy}} = \sigma_{\text{y}} \left(0.2 \times \left(\frac{C_{\text{fract.}}}{\sigma_{\text{y}}} \right)^2 + 0.05 \right) \tag{8}$$

3.3 Source Data

Predictions made using FLiPS will be much more accurate if made using input data that is similar to many source data. The best uncertainty achieved when predicting fatigue life during testing was a factor of approximately four. Using input data which do not correspond to any source data can easily lead to uncertainties of the order of a factor of one thousand. It is therefore essential that the availability of source data is considered when deciding upon input for predictions. The source data used are presented in the following subsections. It should be noted that while uniaxial loading had different values of all variables, torsional and 4-point bending fatigue — for which there were fewer source data — do not account for some variables. These will be detailed in the relevant subsections.

3.3.1 Uniaxial Loading

The ANNs that predict fatigue life under uniaxial loading are able to accept nine out of the ten inputs, as the source data contained variations in each variable except stress concentration factor, K_t. All of the source data uniaxial fatigue was taken from experiments where the sample was not notched, i.e. the stress concentration factor, $K_\mathrm{t} = 1.0$, so no account is taken of stress concentration when predicting the uniaxial fatigue life.

A statistical breakdown of the source data for the uniaxial fatigue ANN is given in table 1. Histograms for each variable are given in figures 5–14.

3.3.2 Torsional loading

For the ANN to predict fatigue life under torsional loading, all source data were taken from experiments performed at ambient temperature and without any notching in the samples, i.e. a stress concentration factor, $K_\mathrm{t} = 1.0$ and an $R = -1.0$. Therefore, none of temperature, stress concentration factor and r are accounted for when making predictions.

A statistical breakdown of the source data for the torsional fatigue ANN is given in table 2. Histograms for each variable are given in figures 15–22.

3.3.3 4-point Bending

For the ANN to predict fatigue life under four point bending, all source data were taken from experiments performed at ambient temperature and without any notching in the samples, i.e. a stress concentration factor, $K_\mathrm{t} = 1.0$ and an $R = -1.0$. Therefore, none of temperature, stress concentration factor and r are accounted for when making predictions.

A statistical breakdown of the source data for the torsional fatigue ANN is given in table 3. Histograms for each variable are given in figures 23–31.

4 Proposed Future Developments

As of March 30, 2017, FLiPS v1.1 has only been deployed on Mac OS. The next planned deployments are for Windows and Android, followed by Linux and, eventually, iOS.

In parallel to deployment, uncertainties in predictions will be calculated by extracting the necessary data from the original ANNs and writing them into the code, along with the necessary calculations.

Variable	Mean	St. dev.	Mode	Lower bound	Lower Quartile	Median	Upper quartile	Upper bound
Yield stress / MPa	983.3	328.8	824	320	824	908	1094	1968
Charpy toughness / J cm^{-2}	93.8	71.5	32.0	4.6	35.8	56.0	150.0	305.0
Elongation (%)	17.2	7.3	18.0	0.5	13.0	18.0	21.0	41.0
Reduction of area (%)	51.5	16.6	64	0	45	54	64	74
UTS / MPa	1124.2	378.4	906	457	906	978	1353	2128
Temperature / °C	21.6	15.3	20	20	20	20	20	180
Loading frequency / Hz	3333.6	7316.3	140	1	120	140	140	20000
Stress amplitude / MPa	535.0	187.0	560	90	400	550	640	1260
Effective R factor	84.7	178.3	-1.0	-1.0	-1.0	-1.0	0.3	730.0
$\log_{10}$ (fatigue life)	6.0	1.1	5.3	3.8	5.2	5.8	6.8	9.9

Table 1: Summary of training data for the uniaxial fatigue ANN.

Variable	Mean	St. dev.	Mode	Lower bound	Lower Quartile	Median	Upper quartile	Upper bound
Yield stress / MPa	786.3	210.7	718	320	633	789	914	1636
Charpy toughness / J cm^{-2}	160.7	53.2	143	15	126	163	195	311
Elongation (%)	21.3	4.8	20.0	10	18	21	24	40
Reduction of area (%)	63.4	9.0	64	5	61	64	68	74
UTS / MPa	903.6	191.0	837	473	790	897	988	1756
Loading frequency / Hz	36.3	6.7	33	33	33	33	33	50
Stress amplitude / MPa	349.1	69.3	350	140	310	350	390	700
$\log_{10}$ (fatigue life)	5.7	0.7	6.1	4.0	5.2	5.6	6.1	8.0

Table 2: Summary of training data for the torsional fatigue ANN.

Variable	Mean	St. dev.	Mode	Lower bound	Lower Quartile	Median	Upper quartile	Upper bound
Yield stress / MPa	861.1	295.4	894	242	654	827	996	2466.7
Charpy toughness / J cm^{-2}	139.1	70.2	134	3.4	100	149	190	311
Elongation (%)	19.2	8.5	21	0.2	17.0	20.0	23.0	68.0
UTS / MPa	979.7	279.8	1088	455	801	924	1088	2168
Reduction of area (%)	57.0	20.7	0	0	59	64	67	93
Loading frequency / Hz	188.3	1640.8	50	30	50	50	50	20000
Stress amplitude / MPa	584.3	184.6	600	40	470	560	640	1410
Stress concentration factor	1.0	0.2	1.0	1.0	1.0	1.0	1.0	3.0
$\log_{10}$ (fatigue life)	5.7	0.8	5.0	3.5	5.1	5.5	6.0	10.0

Table 3: Summary of training data for the four-point bending fatigue ANN.

References

[1] Neuromat Model Manager, `http://thomas-sourmail.net/neural_networks_model_manager.html`, Thomas Sourmail (1999).

[2] Bigback5, David Mackay (1994).

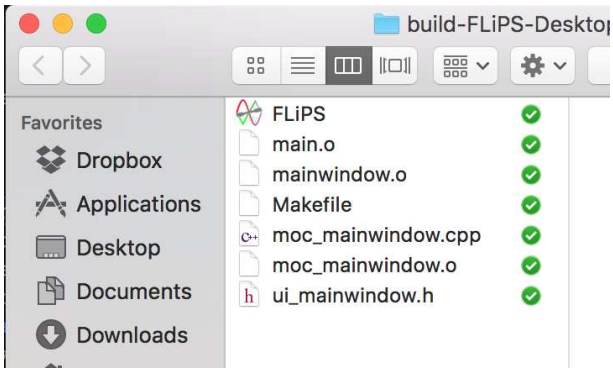


Figure 3: The FLiPS executable in a directory. Double-clicking on it will launch FLiPS.

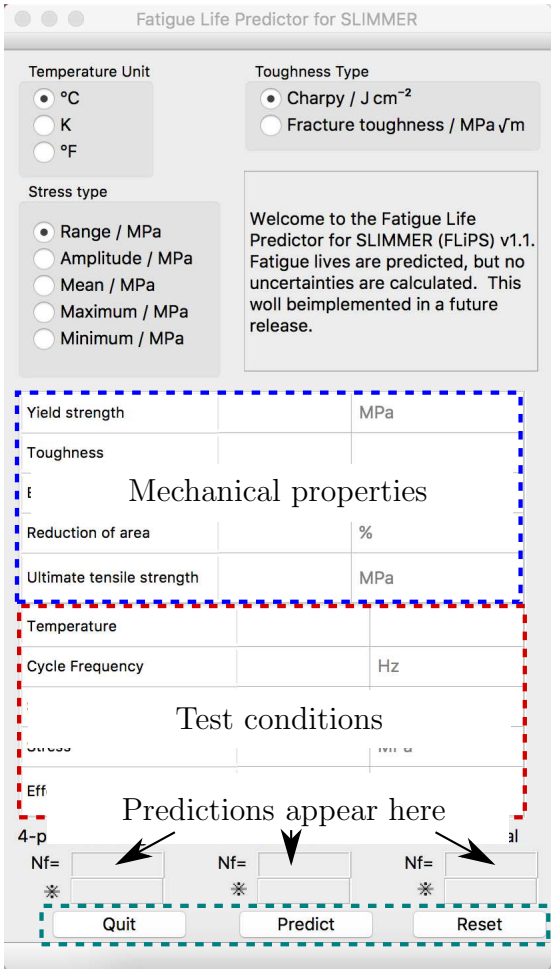


Figure 4: FLiPS window. The top table is for the mechanical properties of the materials being considered, the bottom table contains the test conditions. The three buttons at the bottom control the program.

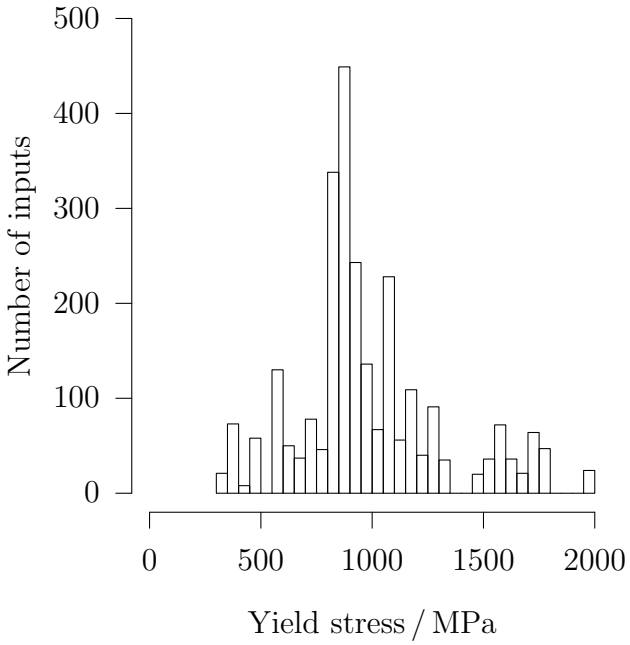


Figure 5: Distribution of yield stresses in the input data for training of the uniaxial fatigue life artificial neural network.

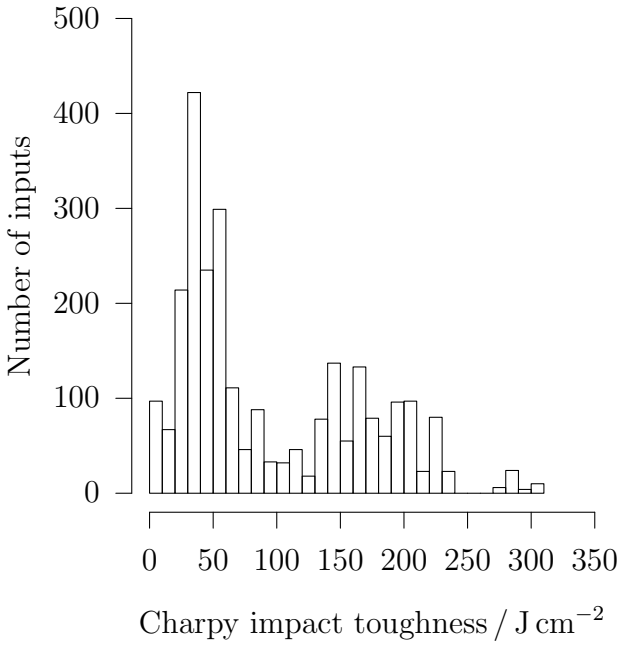


Figure 6: Distribution of Charpy impact toughness in the input data for training of the uniaxial fatigue life artificial neural network.

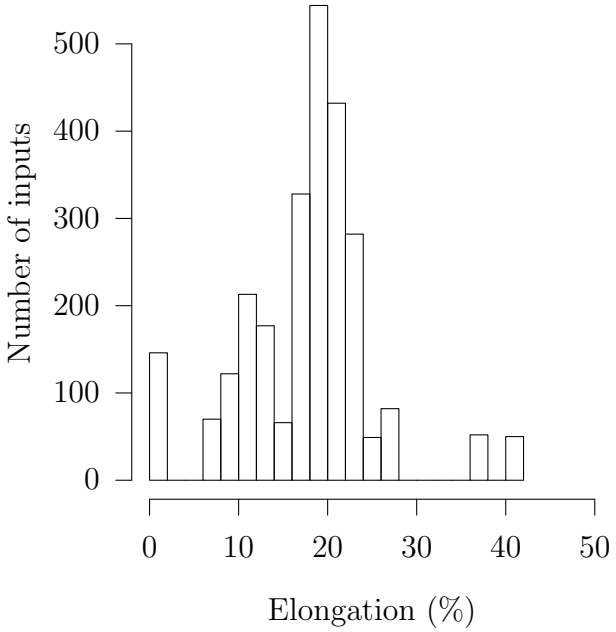


Figure 7: Distribution of elongation in the input data for training of the uniaxial fatigue life artificial neural network.

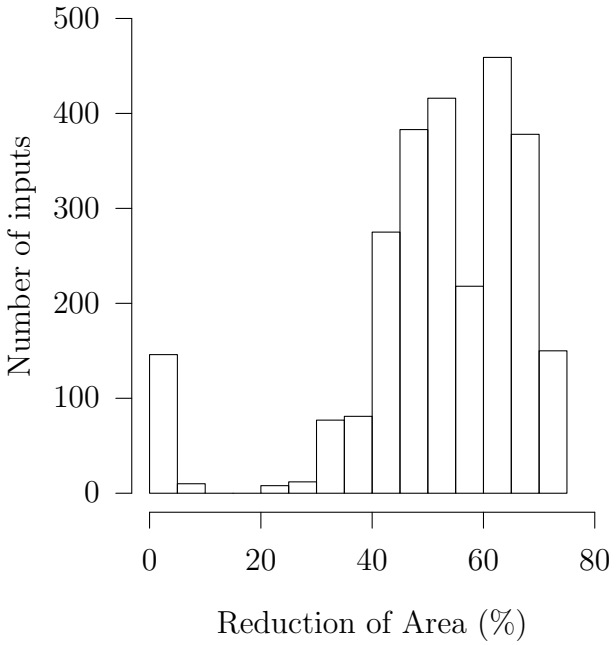


Figure 8: Distribution of reduction of areas in the input data for training of the uniaxial fatigue life artificial neural network.

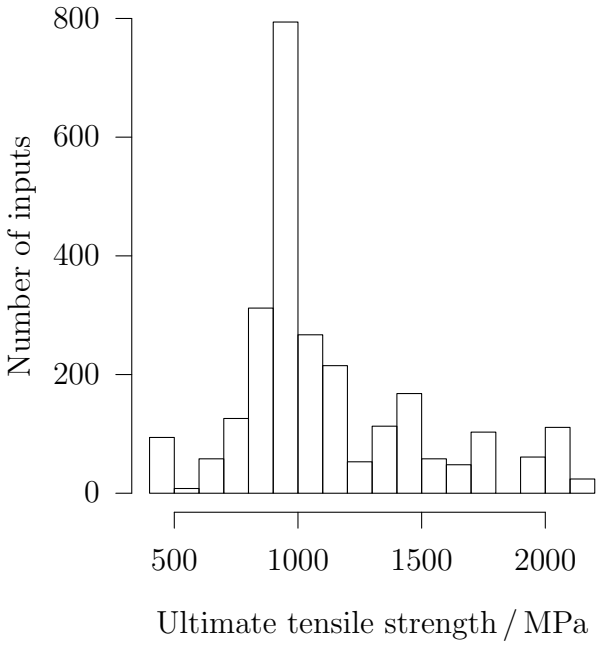


Figure 9: Distribution of ultimate tensile stresses in the input data for training of the uniaxial fatigue life artificial neural network.

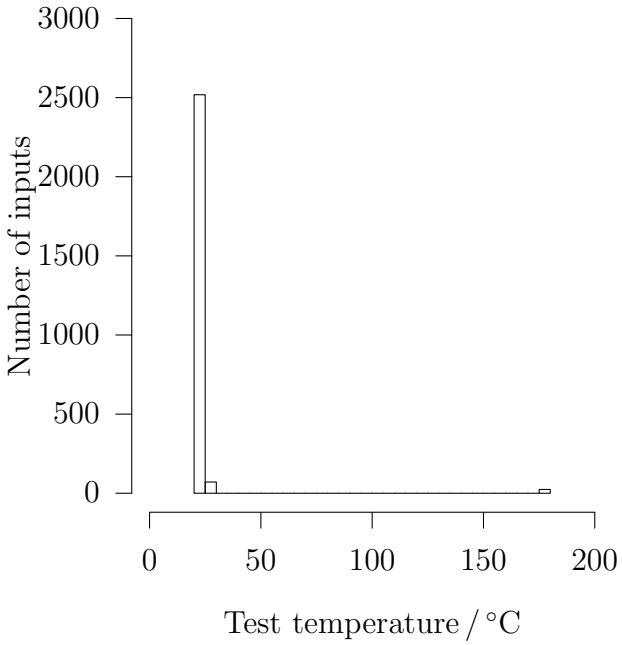


Figure 10: Distribution of test temperatures in the input data for training of the uniaxial fatigue life artificial neural network.

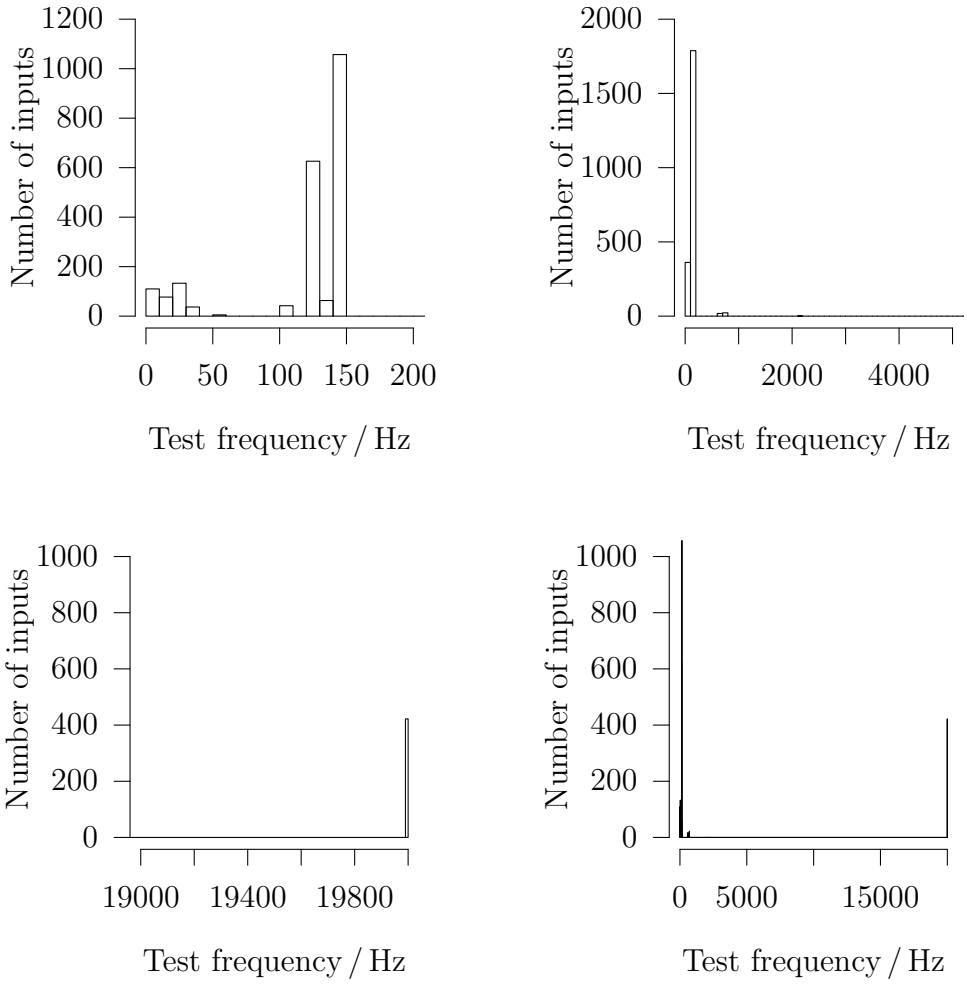


Figure 11: Distribution of test frequencies used in the training of the artificial neural network of fatigue life under uniaxial loading. The various plots show different parts of the range of frequencies used.

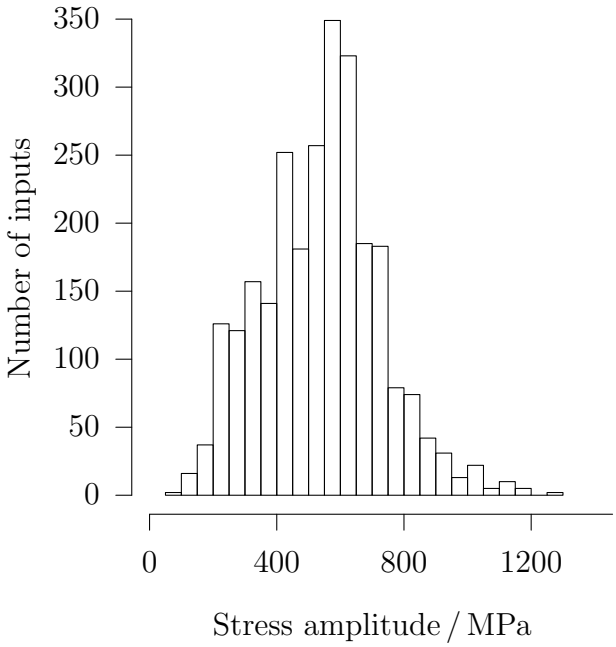


Figure 12: Distribution of stress amplitudes in the input data for training of the uniaxial fatigue life artificial neural network.

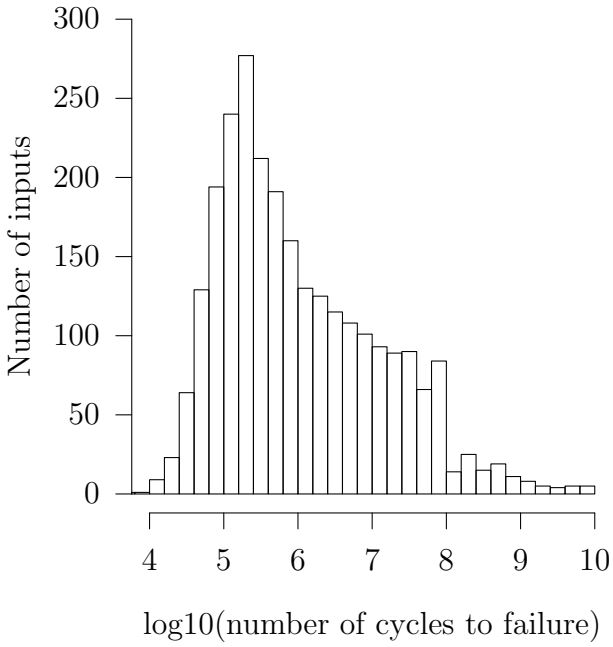


Figure 13: Distribution of the effective R factor of the assessed fatigue life, $\log_{10}(N_{\mathrm{f}})$, in the input data for training of the uniaxial fatigue life artificial neural network.

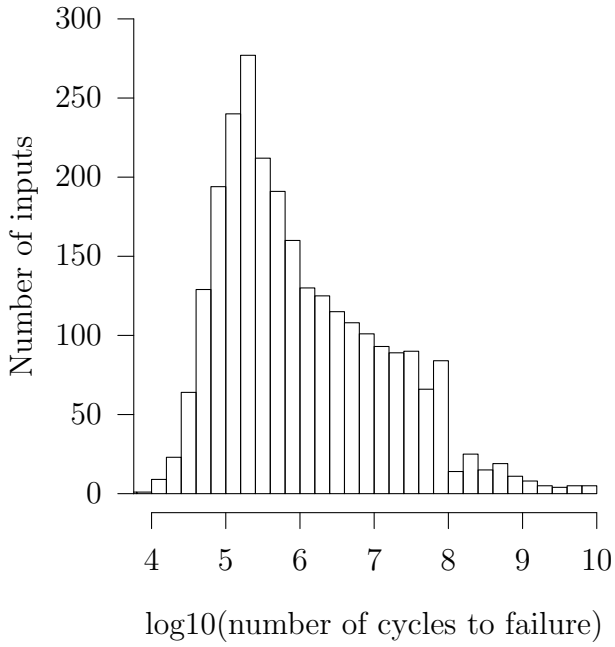


Figure 14: Distribution of the (decadic) logarithm of the assessed fatigue life, $\log_{10}(N_\mathrm{f})$, in the input data for training of the uniaxial fatigue life artificial neural network.

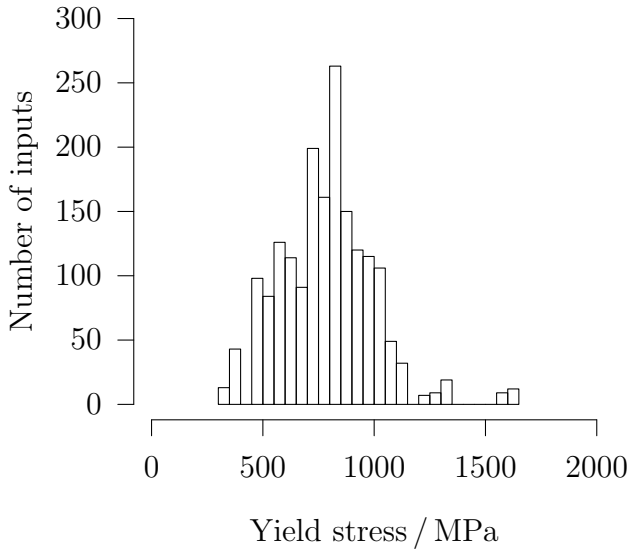


Figure 15: Distribution of yield stresses in the input data for training of the torsional fatigue life artificial neural network.

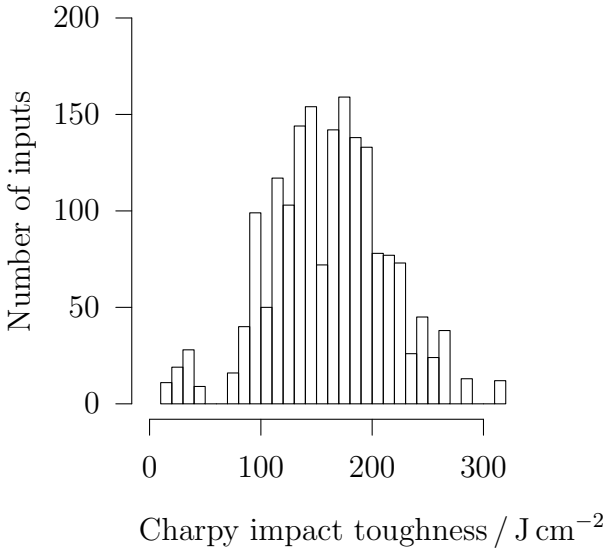


Figure 16: Distribution of Charpy impact toughness in the input data for training of the torsional fatigue life artificial neural network.

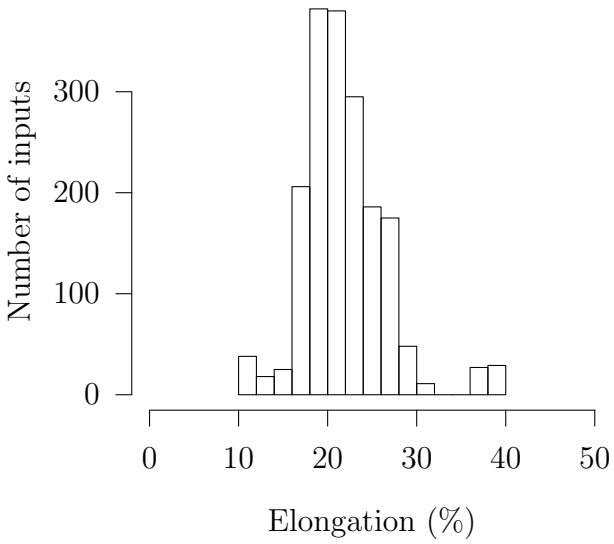


Figure 17: Distribution of elongation in the input data for training of the torsional fatigue life artificial neural network.

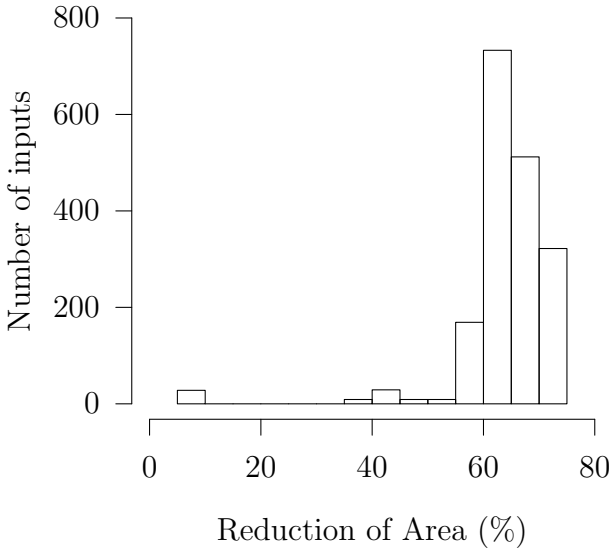


Figure 18: Distribution of reduction of areas in the input data for training of the torsional fatigue life artificial neural network.

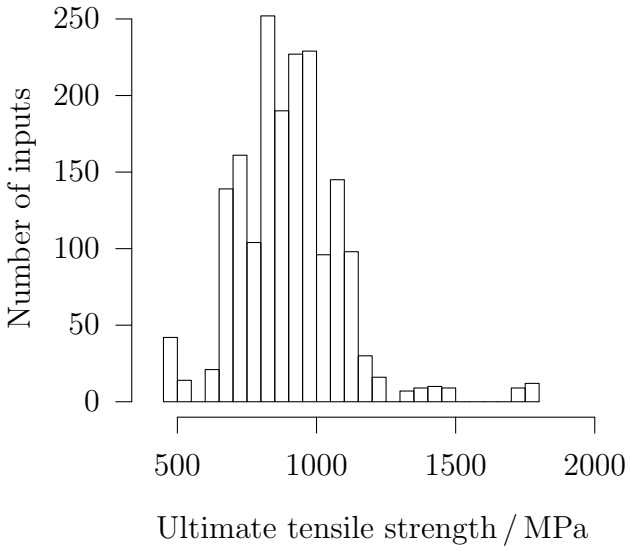


Figure 19: Distribution of ultimate tensile stresses in the input data for training of the torsional fatigue life artificial neural network.

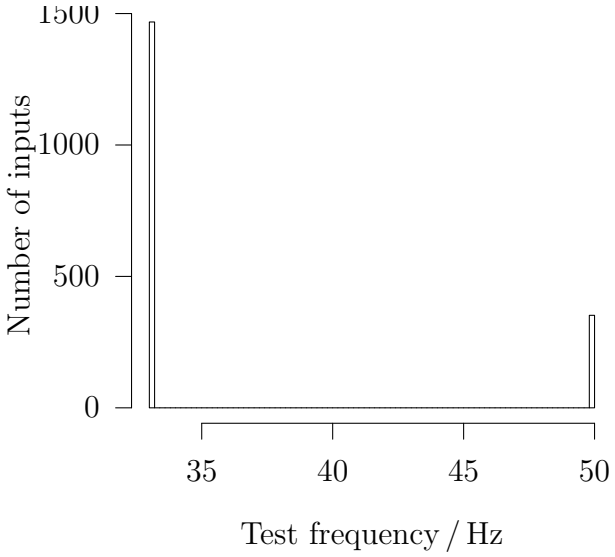


Figure 20: Distribution of test frequencies used in the training of the artificial neural network of fatigue life under torsional loading.

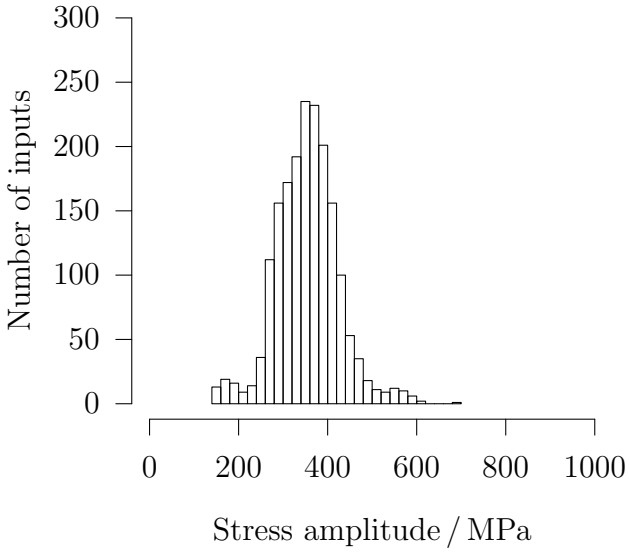


Figure 21: Distribution of stress amplitudes in the input data for training of the torsional fatigue life artificial neural network.

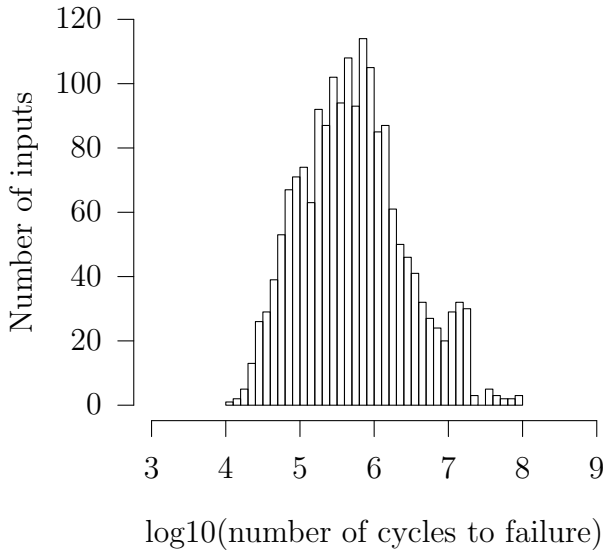


Figure 22: Distribution of the (decadic) logarithm of the assessed fatigue life, $\log_{10}(N_\mathrm{f})$, in the input data for training of the torsional fatigue life artificial neural network.

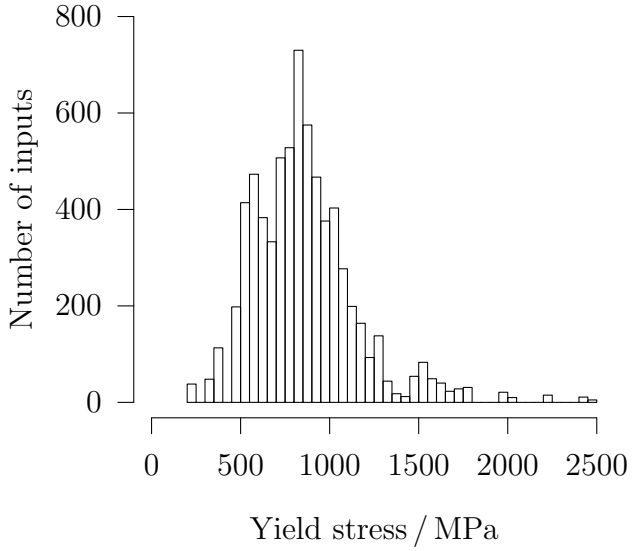


Figure 23: Distribution of yield stresses in the input data for training of the four-point bending fatigue life artificial neural network.

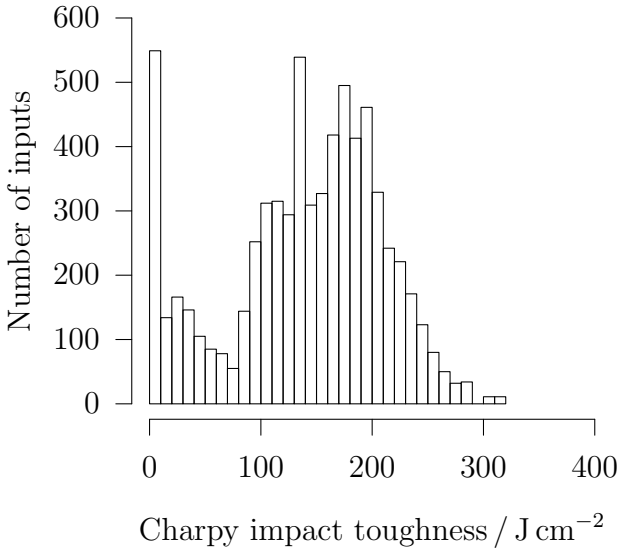


Figure 24: Distribution of Charpy impact toughness in the input data for training of the four-point bending fatigue life artificial neural network.

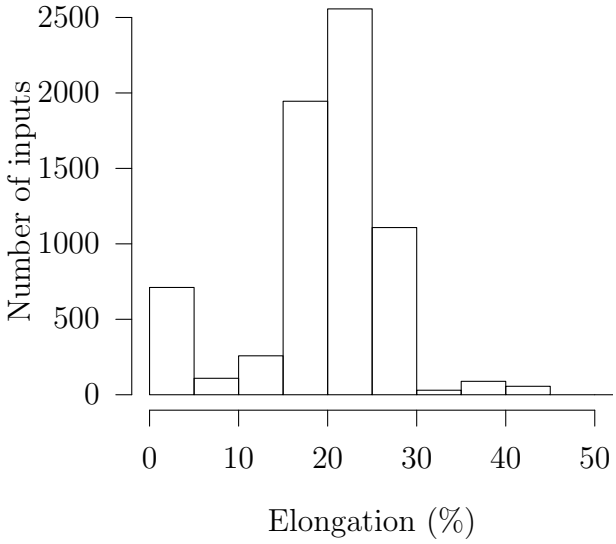


Figure 25: Distribution of elongation in the input data for training of the four-point bending fatigue life artificial neural network.

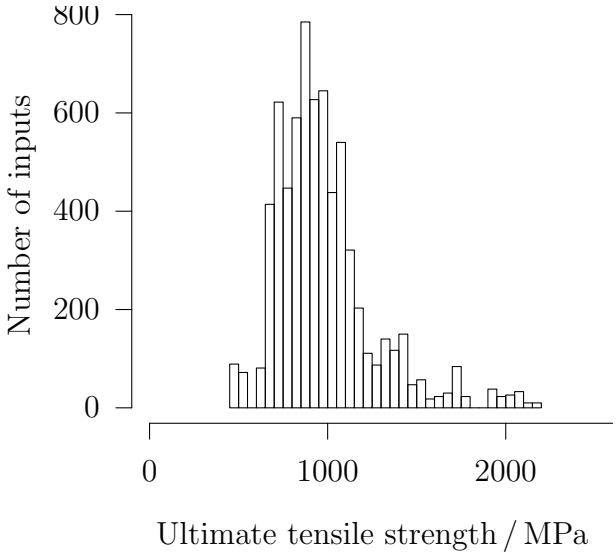


Figure 26: Distribution of ultimate tensile stresses in the input data for training of the four-point bending fatigue life artificial neural network.

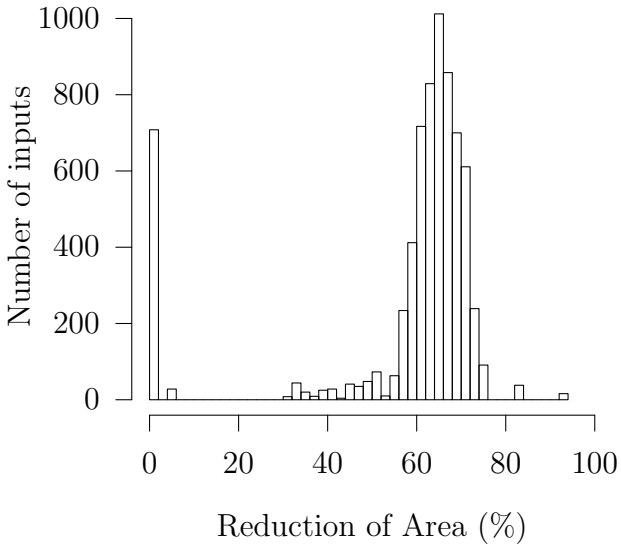


Figure 27: Distribution of reduction of areas in the input data for training of the four-point bending fatigue life artificial neural network.

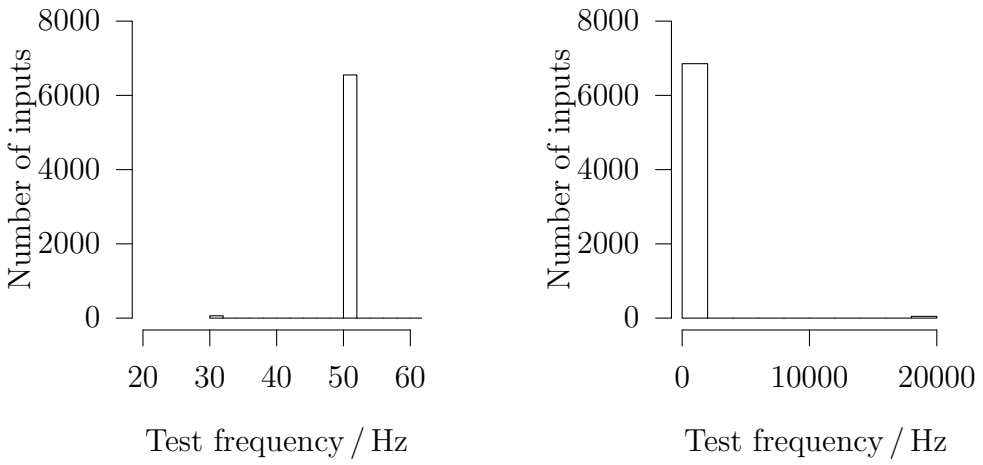


Figure 28: Distribution of test frequencies used in the training of the artificial neural network of fatigue life under four-point bending loading.

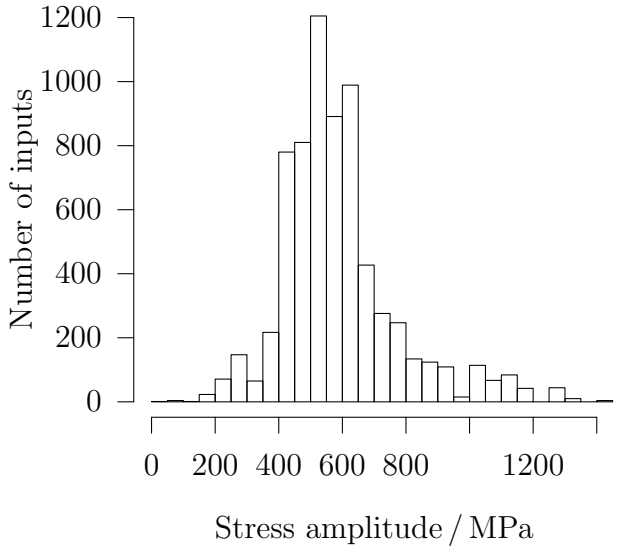


Figure 29: Distribution of stress amplitudes in the input data for training of the four-point bending fatigue life artificial neural network.

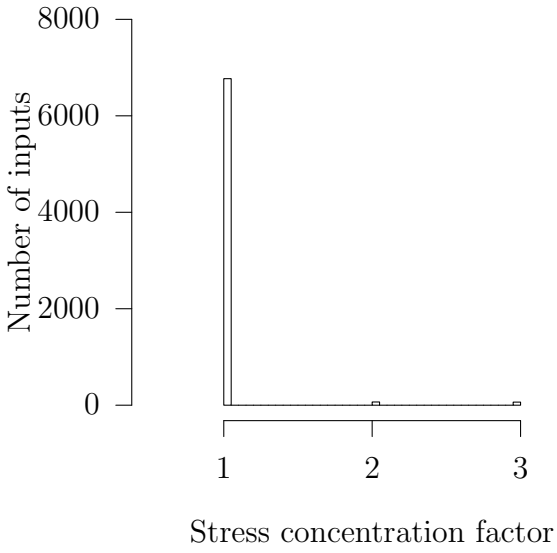


Figure 30: Distribution of stress concentration factors used in the training of the artificial neural network of fatigue life under four-point bending loading.

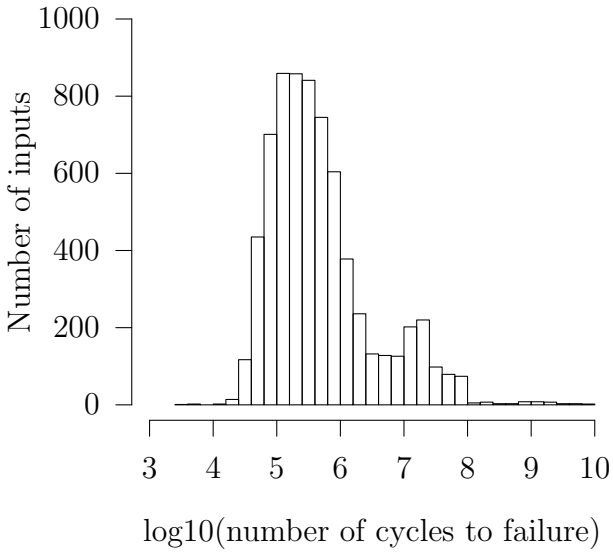


Figure 31: Distribution of the (decadic) logarithm of the assessed fatigue life, $\log_{10}(N_{\mathrm{f}})$, in the input data for training of the four-point bending fatigue life artificial neural network.