Proceedings of the ASME 2020 39th International Conference on Ocean, Offshore and Arctic Engineering
OMAE2020
August 3-August 7, 2020, Virtual Edition
DOI: https://doi.org/10.1115/OMAE2020-18675

OMAE2020-18675

ARTIFICIAL INTELLIGENCE BASED APPROACH FOR PREDICTING FATIGUE STRENGTH USING COMPOSITION AND PROCESS PARAMETERS

Arvind Keprate
Department of Mechanical, Electronics and Chemical Engineering,
Oslo Metropolitan University, Oslo, Norway

R.M. Chandima Ratnayake
Department of Mechanical and Structural Engineering and Material Science,
University of Stavanger, Stavanger, Norway

ABSTRACT

Accurate prediction of the fatigue strength of steels is vital, due to the extremely high cost (and time) of fatigue testing and the often fatal consequences of fatigue failures. The work presented in this paper is an extension of the previous paper submitted to OMAE 2019. The main objective of this manuscript is to utilize Artificial Intelligence (AI) to predict fatigue strength, based on composition and process parameters, using the fatigue dataset for carbon and low alloy steel available from the National Institute of Material Science (NIMS) database, MatNavi. A deep learning framework Keras is used to build a Neural Network (NN), which is trained and tested on the data set obtained from MatNavi. The fatigue strength values estimated using NN are compared to the values predicted by the gradient boosting algorithm, which was the most accurate model in the OMAE 2019 paper. The comparison is done using metrics such as root mean square error (RMSE), Mean Absolute Error (MAE), Coefficient of Determination (R^2) and Explained Variance Score (EVS). Thereafter, the trained NN model is used to make predictions of fatigue strength for the simulated data (1 million samples) of input parameters, which is then used to generate conditional probability tables for the Bayesian Network (BN). The main advantage of using BN over previously used machine learning algorithms is that BN can be used to make both forward and backward propagation during the Bayesian inference. A case study illustrating the applicability of the proposed approach is also presented. Furthermore, a dashboard is developed using PowerBI, which can be used by practicing engineers to estimate fatigue strength based on composition and process parameters.

INTRODUCTION

Fatigue is one of the most dominant degradation mechanisms causing component failure in the oil & gas (O&G) and maritime industries [1]. A recent research study from DNV GL points out that fatigue resulted in about 30% of all component failures (out of the 1000 cases considered in the study), while tubes & piping were the most commonly failed equipment in the O&G and maritime sectors. Figures 1 and 2 (in the Appendix) depict other relevant findings of the research study [1]. Fatigue cracking of metallic materials is attributed to the effect of cyclic stresses. Currently, the S-N approach is used to estimate the fatigue life of offshore structures during the design phase. This approach is generally considered reliable, as S-N curves are derived directly from the fatigue test data [2]. It is particularly suited to predicting the fatigue behavior of components subjected to fluctuating stresses below the yield (i.e. High Cycle Fatigue (HCF)). For ductile metals, HCF is generally considered to be greater than 100,000 cycles of operation [3].

Most of the time, offshore structures and mechanical items (like topside piping near rotary equipment) are subjected to HCF. Therefore, during the design stage, the S-N approach is the preferred option to predict fatigue damage for these items. The reason for this is easy application, simple calculation and extensive guidance [2]. DNV GL also recommends the use of the S-N approach for fatigue analysis, as long as the predicted fatigue life meets the regulatory requirements and is above the service life of the structure [2]. For constant amplitude loading, the fatigue properties of materials (and structures) are usually described by the S-N diagram/curve [2]. The common terms used with the S-N diagram are “fatigue life”, “fatigue strength”, “fatigue limit” and “endurance limit”. Each of these terms has been described differently by different researchers. A short review of this was carried out by the authors in [4].

The fatigue limit in a material may be typically obtained experimentally or through the traditional approach of 50% of the

tensile strength [5]. During the design and operational phases, this value is recalculated as the effective or admissible fatigue limit, taking into account the effect of variables such as average stress, surface finish, environment, stress concentrators, reliability, component size, grain size, heat treatment conditions, chemical composition, and level of inclusions [5]. To evaluate the fatigue strength based on S-N curves, many tests are needed, which require a lot of time and incur high costs [5]. Lately, researchers have focused on applying various machine learning (ML) approaches to estimate the fatigue strength, using composition and process parameters. To the best of our knowledge, Agarwal and Choudhary [6] were the first researchers to employ different ML algorithms to predict fatigue strength using the NIMS database. Thereafter, authors have used a cross-industry process for data mining (CRISP-DM) approach in order to gain meaningful insights from the NIMS dataset and to estimate the fatigue strength of carbon and low alloy steels, using composition and processing parameters. Twelve different ML algorithms were tested and compared, and gradient boosting was found to be the most accurate algorithm. The complete research can be found in [4].

The research presented in the current manuscript explores the possibility of other techniques of Artificial Intelligence (i.e. Deep Learning and Bayesian Network) to predict the fatigue strength. In comparison to the previous works, which were based on ML, the main advantage of the current approach is that the trained Bayesian Network (BN) can be used to make both forward and backward uncertainty propagation and to make inference on fatigue strength, given input parameters (for forward propagation), or generate updated probability distributions of input parameters, given evidence on fatigue strength (for backward propagation). The authors have also developed a PowerBI dashboard, which can be used by practicing engineers to make inference and to perform sensitivity analysis. Although the authors have used the same data as that used by previous studies [4, 6], a different AI approach for predicting fatigue strength is followed in this manuscript, the details of which are discussed in Section 2. Thereafter, in Section 3, an illustrative case study is presented. Finally, a conclusion is provided in Section 4.

ARTIFICIAL INTELLIGENCE

Artificial Intelligence (AI) is the study of how to build computer systems that exhibit human intelligence in some manner [7]. AI has resulted in many breakthroughs in computer science. Many core research topics in computer science today have developed out of AI research, for example neural networks, evolutionary computing, machine learning, Bayesian networks, natural language processing, etc., to name a few. In many cases, the primary focus for these research topics is no longer the deployment of AI, as researchers have become a discipline in themselves. The two most commonly used AI sub-types used in the scientific community are machine learning (ML) and deep learning (DL). Figure 3 shows the relationship between AI, ML and DL.

Lately, AI has been used by researchers in the various domains of the oil and gas sector. Rachman and Ratnayake [9] used an Artificial Neural Network (ANN)-based approach for screening assessment of production facilities. The authors have also used AI (fuzzy logic) for determining fatigue critical topside piping locations on offshore platforms [10, 11]. Furthermore, the authors have utilized BN (another AI approach) for remaining fatigue life and reliability assessment [12, 13] of topside piping. Although there are various sub-branches of AI, the two facets of AI used in this paper are neural network (NN) and BN, which are described next.

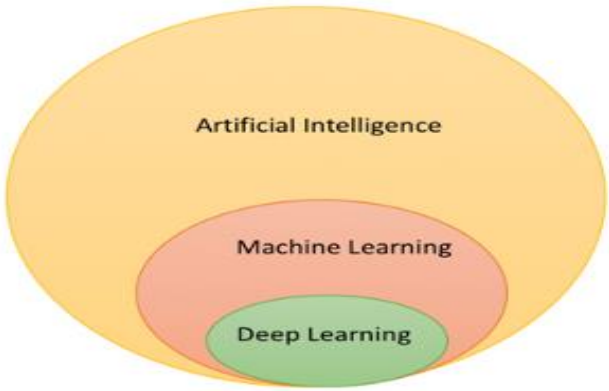

Figure 3. AI vs ML vs DL [8].

Neural Network

A neural network (NN) is defined as "*a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs*" [14]. A NN typically comprises a large number of layers of neurons, as shown in Figure 4 [8]. As can be seen, there are interconnecting lines between different neurons; these represent the trail of information flow [8]. Furthermore, each interconnecting line has a weight associated with it, which regulates the signal amid two connecting neurons. Several frameworks are available in Python for implementing NN such as TensorFlow (by Google), MxNet, PyTorch (by Facebook), Gluon, Keras, etc. Owing to its simplicity, Keras is used in this paper. The three steps to construct a deep learning neural network in Keras are shown in Figure 5 [15]:

Define Model: To build a deep learning model, an analyst needs to define the layers (Input, Hidden, and Output). In this manuscript, the authors built a sequential model, which means that they define layers sequentially. Furthermore, a fully connected deep network shall be employed.
Compile Model: At this stage, the defined model for training is configured, by setting optimizer to change the weights and biases of the NN model. Setting the loss function to evaluate the model's performance is also a sub part of compiling an NN model.
Fit Model: Finally, the compiled model is fitted on the training dataset (which generally is 70% of the full dataset). In this stage, we set the number of epochs and the batch size. Epochs is forward and backward pass of all our training examples, while batch size

is the number of training examples in one forward and backward pass. As the batch size decreases, the computation time increases; also, as the epoch increases, the computation time increases [15].

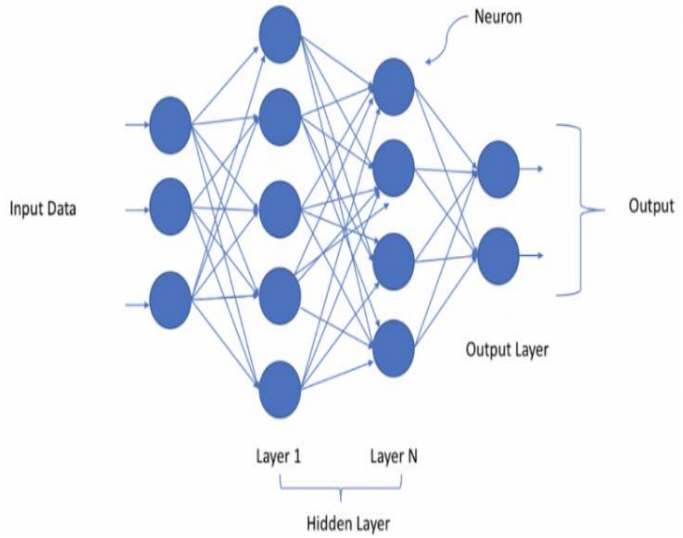

Figure 4. A deep neural network with N hidden layers [8].

Figure 5. Three steps for constructing NN in Keras [15].

Bayesian Network

Bayesian networks are directed acyclic graphs (DAGs), in which nodes represent random variables and arcs represent direct probabilistic dependencies among them [16]. The two vital parts of the BNs (as shown in Figure 6) are their nodal structure (borrowed from graph theory) and the directed links/arcs, representing conditional probability tables (CPTs) (borrowed from probability theory). An example of a BN is shown in Figure 6, where X and Y are referred to as parent nodes, while Z is a child node. The nodes represent random variables, while the probabilistic relationship/dependencies between parent and child nodes are represented by the directed arcs/links [16]. BNs are thus a graphical representation of uncertain quantities (and decisions) that explicitly reveals the probabilistic causal dependence between the variables, as well as the flow of information in the model [17]. More discussion about the BN can be found in [13, 17]. One vital feature of the BNs is that they can be built either by using expert knowledge (i.e. theory) or they can be machine-learned from the data, i.e. BNs can be developed from a combination of human and artificial intelligence [16]. Furthermore, an added advantage of BNs over other ML techniques is that they can be used both for forward and backward uncertainty propagation and to make inference in the parameter of interest by setting evidence in other parameters. Owing to the aforementioned advantage, the authors have utilized BN in this manuscript. In the next section, an illustrative case study is performed, to predict the value of fatigue strength, based on composition and process parameters.

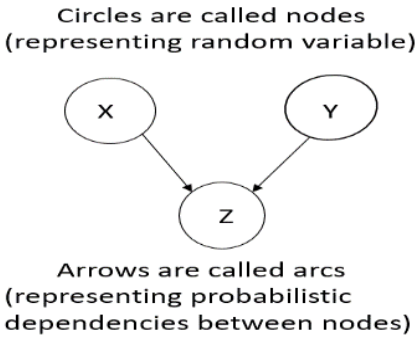

Figure 6. Illustrative example of a BN.

ILLUSTRATIVE CASE STUDY

Data Understanding

The Fatigue Dataset for Steel from the National Institute of Material Science (NIMS) MatNavi [18] was used in this work. It is one of the largest databases in the world, with details on composition, mill product (upstream) features and subsequent processing (heat treatment) parameters. The database comprises carbon and low-alloy steels, carburizing steels and spring steels. However, as only carbon and low-alloy steels are of importance for the offshore industry, we have only chosen the dataset relevant to the aforementioned material type.

The original data used by the authors in [4] has 371 instances/rows, 25 features/columns (composition and processing parameters), and one target property (fatigue strength). However, the 10 most important input features were identified using feature engineering in the previous work carried out by the authors [4] and have been used in this paper to train the NN and BN. The input and target parameters used in the case study are shown in Table 1.

Table 1. Data attributes for NIMS database.

Abbreviation	Description	Parameter Type
C (%)	Carbon	10 Most Important Input Parameters Extracted from Feature Engineering
Si (%)	Silicon	
Mn (%)	Manganese	
P (%)	Phosphorus	
S (%)	Sulfur	
Ni (%)	Nickel	
Cr (%)	Chromium	
Cu (%)	Copper	
dA	Area Proportion of Inclusions Deformed by Plastic Work	
TT	Tempering Temperature	
Fatigue	Rotating Bending Fatigue Strength (at	Target Variable Which Needs to be Predicted

	10^7 cycles and room temperature)	

Data Preparation

Normalization of the data was performed, to change the values of numeric columns in the dataset to use a common scale, without distorting differences in the ranges of values or losing information. Normalization is also required for some algorithms to model the data correctly [19]. The great difference in the scale of the numbers could be problematic, while we try to combine the values as features during predictive modeling. Thus, normalization is performed to avoid the aforementioned problem, by creating new values that maintain the general distribution and ratios in the source data, while keeping values within a scale applied across all numeric columns used in the model [19]. In the further analysis, only the normalized dataset has been used.

Neural Network Modeling

After the data has been preprocessed, the next step is to split the data into training and testing. Various methodologies are available for the aforementioned; however, the most robust technique is k-fold cross validation (shown in Figure 7), which has been used in this manuscript.

Figure 7. Illustrative example of k-fold cross validation.

To build a deep learning model, sequential modeling was used, consisting of input, hidden and output layers of neurons. The complete architecture of the NN model is shown in Figure 8. As can be seen from Figure 8, 100 Neurons are used in the input layer, 10 neurons are used in the first hidden layer, 5 Neurons are used in the second hidden layer and, finally, 1 neuron is used in the output layer. The activation functions used for input and output layers are sigmoid and linear, respectively. The optimizer, Adam, was used during compilation of the NN model, while mean absolute error was used as the loss function.

During the modeling stage, it is important to choose the parameters of the NN, so that the prediction accuracy is improved. For instance, one of the parameters which needs to be selected for NN is the number of epochs and batch size. As described earlier, epoch is the forward and backward pass of all our training examples, while batch size is the number of training examples in one forward and backward pass; as the batch size decreases, the computation time increases; also, as the epoch increases, the computation time increases. For example, let us say that we have 2000 training examples that we are going to use. We can divide the dataset of 2000 examples into batches of 500; it will then take four iterations to complete one epoch. Figure 9 illustrates the relation of loss function (mean absolute error) to number of epochs. It is deduced from Figure 9 that, when the number of epochs is equal to 150, the loss on test data becomes almost constant; hence, we chose 150 as the number of epochs for the NN. A batch size of 2 was chosen to improve the accuracy of NN at the expense of the computational time.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_21 (Dense)             (None, 100)               1100
_________________________________________________________________
dense_22 (Dense)             (None, 10)                1010
_________________________________________________________________
dense_23 (Dense)             (None, 5)                 55
_________________________________________________________________
dense_24 (Dense)             (None, 1)                 6
=================================================================
Total params: 2,171
Trainable params: 2,171
Non-trainable params: 0
```

Figure 8. Neural network architecture.

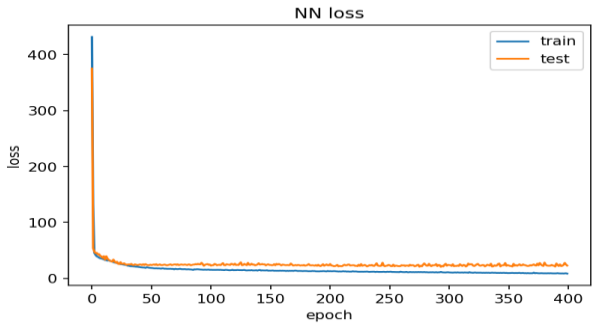

Figure 9. Number of epochs for NN.

Evaluation

In order to compare the accuracy of the gradient boosting algorithm (from OMAE 2019) to the NN model, four metrics, namely, Root Mean Square Error (RMSE), Mean Absolute Error (MAE), coefficient-of-determination (R^2) and Explained Variance Score (EVS), are used. Mathematically, these are given by Eq. (1), where y_i is the actual response, $\hat{y}_i$ is the predicted response, n is the total number of samples, and Var is variance. By looking at the value of various metrics in Table 2, it is seen that NN is not as accurate as the gradient boosting (GB) algorithm. However, the authors wish to state that they spent considerably more time in tuning the parameters of the GB algorithms, in comparison to those of the NN. Furthermore, it is possible to enhance the accuracy of NN, if, rather than the Multi-Layer Perceptron (MLP) used in this paper, a Convolutional Neural Network (CNN) is used. One of the main aims of this paper was to test the performance of the deep learning algorithm (i.e. NN) in comparison to the machine learning algorithm (i.e. GB). This comparison is also shown in Figure 10, where the values of the actual fatigue strength (i.e. from the test dataset) and the predicted fatigue strength (GB and NN) have been

plotted. Most of the dat points tend to follow the regression fit line indicating a good degree of fit between actual fatigue strength and predicted values. Furthermore, it can be seen from Figure 10, GB has better prediction in comparison to NN, and, for both the algorithms, there are very few outliers; in general, the trend between the actual and predicted fatigue strength is almost linear, thus indicating good prediction accuracy of both the GB and NN algorithm.

$$RMSE = \sqrt{\frac{\left(\Sigma_{i=1}^{n}(y_i - \hat{y}_i)^2\right)}{n}}$$

$$MAE = \frac{\Sigma_{i=1}^{n}|y_i - \hat{y}_i|}{n} \quad (1)$$

$$R^2 = 1 - \frac{SS_{Error}}{SS_{Total}}$$

$$EVS = 1 - \frac{Var(y_i - \hat{y}_i)}{Var(y_i)}$$

Table 2. Comparison of different regression algorithms.

	RMSE	MAE	EVS	R^2
Gradient Boosting	11.87	9.67	0.966	0.966
Neural Network	19.09	15.84	0.913	0.912

Bayesian Network Modeling

As described earlier in the manuscript, there are two parts to building a Bayesian Network. The first is to generate the nodal structure of various parameters. For the problem at hand, Hugin [20] software was used to build the graphical network, which is shown in Figure 11 in the Appendix. All the input parameters (shown in Table 1) act as parent nodes, while the target feature (i.e. fatigue strength) acts as a child node. The next step is to populate the conditional probability (CP) table of the developed nodal structure. This is done by generating an input parameter space of 1 million samples (using uniform distribution of all the input parameters) and then using the trained NN to predict the value of fatigue strength for this input parameter space. Once the new dataset of 1 million is available, then the CPs are generated using the crosstab function of the Pandas library [21]. The next step is to fill in the generated CPs into Hugin, which can then be used for making forward and backward propagation. However, in this manuscript, the Bayesian inference is not made using Hugin but by using PowerBI. This is because the data visualization in PowerBI is much more interactive then in Hugin. Thus, an interactive dashboard is made in PowerBI which can be used by the practicing engineers to make inference on fatigue strength. A snapshot of the developed dashboard is presented in Figure 12 in the Appendix.

Result Discussion

Once the PowerBI dashboard is ready to use, the next step is to perform forward and backward uncertainty propagation while making Bayesian inference. During the forward propagation the evidence is set on one or more input parameters, to see how the distribution of the output parameter before and after evidence changes. For example, in Figure 13, evidence is set in the input parameter (Cr %), and its effect is shown by the posterior distributions of fatigue strength. Figure 13 clearly shows that, for lower values of the input parameter (Cr %), the distribution (posterior) of the fatigue strength shifts considerably to the left of prior distribution. This implies a strong positive correlation between these two variables (the strong positive correlation between these two variables can also be seen from the correlation matrix created by the authors in [4]). In this way, an analyst can set evidence in multiple input parameters and see the change in the probability distribution of the output parameter, i.e. fatigue strength.

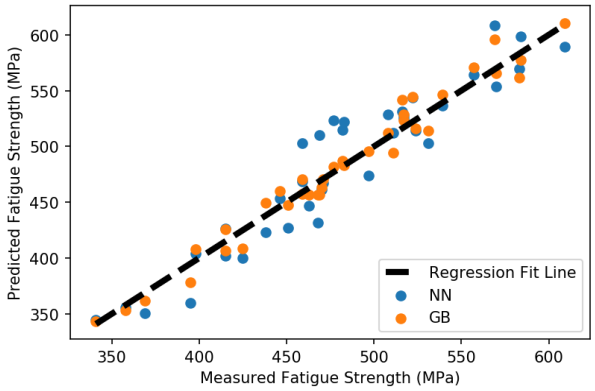

Figure 10. Actual vs. predicted fatigue strength (GB vs. NN).

A more interesting use of the developed BN-based PowerBI dashboard is for backward propagation. During the backward propagation, the evidence is set on the output parameter, to see how the distribution of the input parameter changes. For example, in Figure 14, evidence is set on the output parameter (i.e. the fatigue strength upper threshold value is set to 400 MPa). The two ways in which valuable information can be extracted from the developed dashboard while performing backward propagation are sensitivity analysis and updated posterior distributions of input parameters. From Figure 14, it is clearly visible that the three input parameters highlighted in the red boxes have least variability in their probability distributions after setting evidence on fatigue strength. This implies that fatigue strength is least sensitive to these three input parameters (namely, S (%), P (%), Cu (%)). Likewise, similar interpretations about the most sensitive input parameters can be made from the dashboard. Another use of the dashboard during backward propagation is to get updated input parameter distributions for the given fatigue strength. A possible use of the developed tool could be to use the updated input parameter distributions, which would yield a material with a fatigue strength in the given threshold (i.e. less than 400 MPa). In this way, the developed BN can serve as an optimization or recommendation tool for the design/material engineers and help them to gain more understanding about the relationship between composition/process parameters and fatigue strength.

CONCLUSION

The manuscript utilized an AI-based approach for predicting the fatigue strength of carbon and low-alloy steels from process and composition parameters. The fatigue dataset for steel, available from the National Institute of Material Science (NIMS) MatNavi, was used for analysis. The dataset consisted of 371 instances, 10 input parameters (process and composition parameters) and one target variable (fatigue strength). K-fold cross validation was performed to split the data into training and testing. A deep learning sequential NN, with one input layer, two hidden layers and one output layer, was developed, using Keras API. The results of the NN were compared to the gradient boosting (GB) algorithm on the basis of four metrics, namely, RMSE, MAE, R^2 and EVS. NN had slightly less accuracy than GB; however, overall, the performance of NN was good and could be improved in the future by using CNNs. Thereafter, a BN was developed, and a CP table was populated using the predictions from NN on the simulated input parameter space (consisting of 1 million points). The paper also demonstrated the use of a PowerBI dashboard, which can be used for forward and backward uncertainty propagation while making Bayesian inference. The developed dashboard possesses the capability to perform sensitivity analysis and generate posterior distributions of various parameters, given evidence in one or more parameters. The interactive feature of the dashboard allows evidence to be set in multiple parameters at the same time and thus can serve as an optimization/recommendation tool for design and material engineers interested in understanding the relationship between fatigue strength and process/composition parameters.

ACKNOWLEDGMENT

The authors would like to thank NIMS for making the raw experimental data on fatigue strength publicly available.

REFERENCES

1. https://www.dnvgl.com/oilgas/laboratories-test-sites/article/tubes-and-piping-are-the-most-failure-prone-components.html
2. DNV RP C203, (2010), "Fatigue Design of Offshore Steel Structures". Det Norske Veritas, Høvik, Norway.
3. Keprate, A. and Ratnayake, R.M.C., (2015), "Fatigue and Fracture Degradation Inspection of Offshore Structures and Mechanical Items: The State of the Art". In: Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering, 2015, St. John's, Newfoundland, Canada.
4. Keprate, A. and Ratnayake, R.M.C., (2019), "Data Mining for Estimating Fatigue Strength based on Composition and Process Parameters". In: Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering, 2019, Glasgow, Scotland, UK.
5. Martinez, A., Jaimes, A., Serna, A., Uribe, I., (2010), "Development of a Correlation to Estimate the Fatigue Strength for Steels based on Low-Cost Tests", CT&F, Vol. 4, pp. 71-81.
6. Agarwal, A. and Choudhary, A., (2018), "An Online Tool for Predicting Fatigue Strength of Steel Alloys Based on Ensemble Data Mining", International Journal of Fatigue, Vol. 113, pp. 389-400.
7. W.J. Teahan, (2010), *Artificial Intelligence – Agents and Environments*. BookBoon.com, 2010.
8. https://towardsdatascience.com/a-laymans-guide-to-deep-neural-networks-ddcea24847fb
9. Rachman, A. and Ratnayake, R.M.C., (2018), "Artificial Neural Network Model for Risk Based Inspection Screening Assessment of Oil and Gas Production System". In: Proceedings of the International Ocean and Polar Engineering Conference, 2018, Sapporo, Japan.
10. Keprate, A. and Ratnayake, R.M.C., (2016), "Determining the Degree of Fuzziness for Fuzzy-AHP Methodology used for Identifying Fatigue Critical Piping Locations for Inspection". In: IEEE International Conference on Intelligent Engineering Systems, Budapest, Hungary.
11. Keprate, A. and Ratnayake, R.M.C., (2016), "Enhancing Offshore Process Safety by Selecting Fatigue Critical Piping Locations for Inspection Using Fuzzy-AHP Based Approach", Process Safety and Environmental Protection, Vol. 102, pp. 71-84.
12. Keprate, A. and Ratnayake, R.M.C., (2019), "Assessment of Reliability and Remaining Fatigue Life of Topside Piping Using Dynamic Bayesian Network". In: IEEE International Conference on Industrial Engineering and Engineering Management, Macau.
13. Keprate, A. and Ratnayake, R.M.C., (2019), "Use of Bayesian Network for Risk-Based Fatigue Integrity Assessment Application for Topside Piping in an Arctic Environment", International Journal of Offshore and Polar Engineering, Vol. 29(4), pp. 421-428.
14. Maureen, C., (1989), "Neural Network Primer: Part I". AI Expert, 2 (12), 1987.
15. https://medium.com/analytics-vidhya/build-your-first-neural-network-model-on-a-structured-dataset-using-keras-d9e7de5c6724
16. Pearl, J., (1988), *Probabilistic Reasoning in Intelligent Systems*. San Francisco, USA: Morgan Kaufmann, 1988.
17. Mahadevan, S., Zhang, R., Smith, N., (2001), "Bayesian Networks for System Reliability Reassessment", Structural Safety, Vol. 23, pp. 231-251.
18. https://smds.nims.go.jp/fatigue/index_en.html
19. https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/normalize-data
20. https://www.hugin.com/index.php/hugin-explorerhugin-educational/
21. https://pandas.pydata.org/

APPENDIX

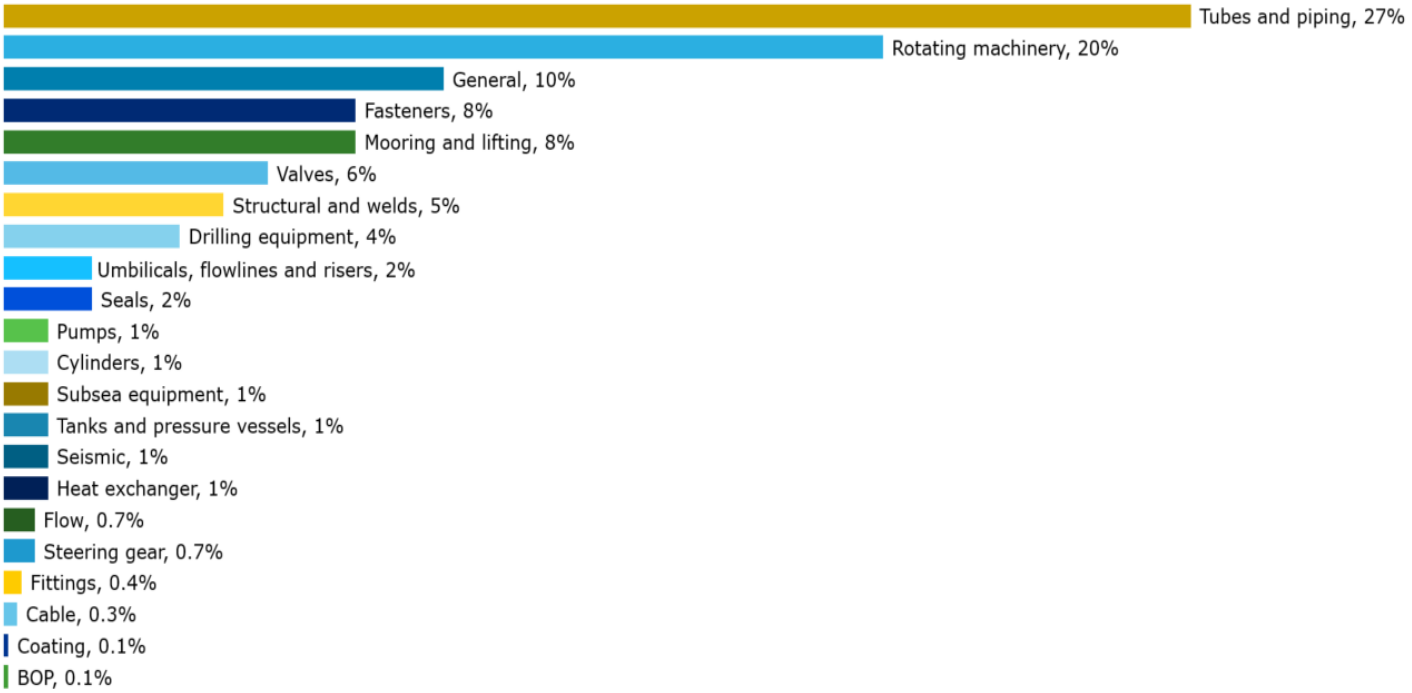

Figure 1. Most commonly failed equipment [1].

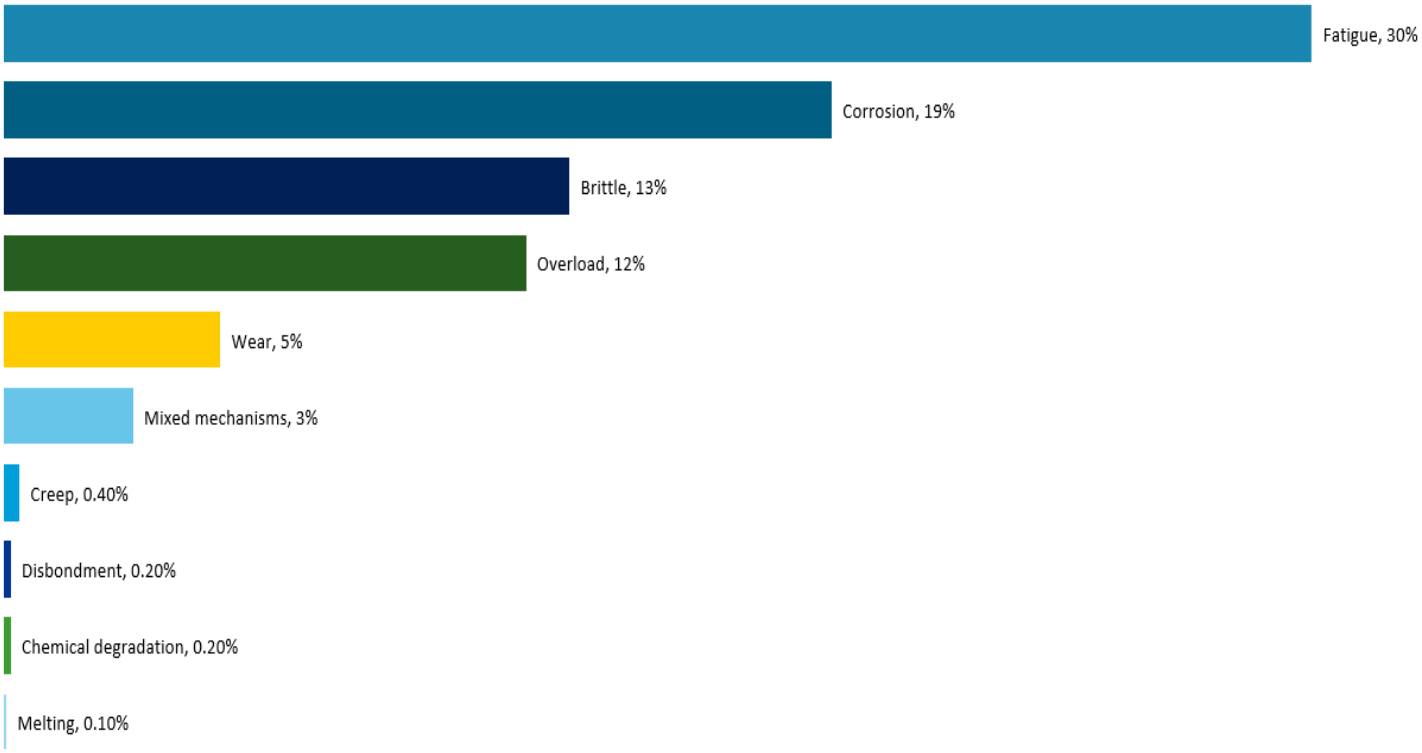

Figure 2. Most dominant degradation mechanisms [1].

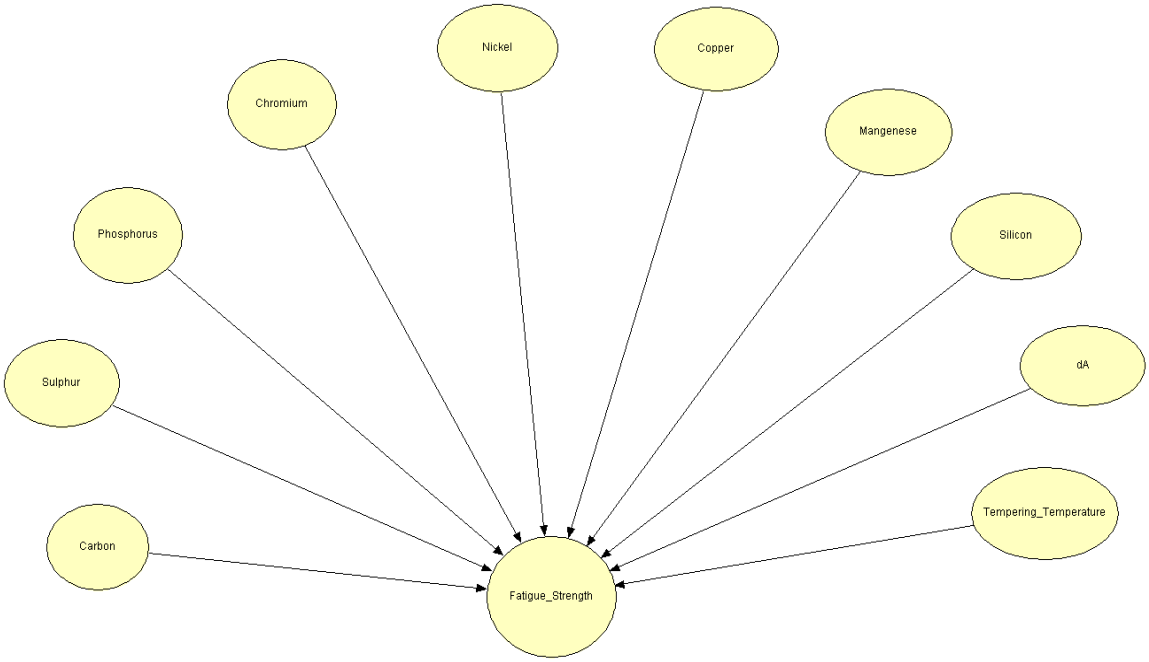

Figure 11. Bayesian network nodal structure.

Figure 12. Snapshot of PowerBI dashboard showing initial prior distributions of 10 input features (blue colored) and 1 target feature (yellow colored).

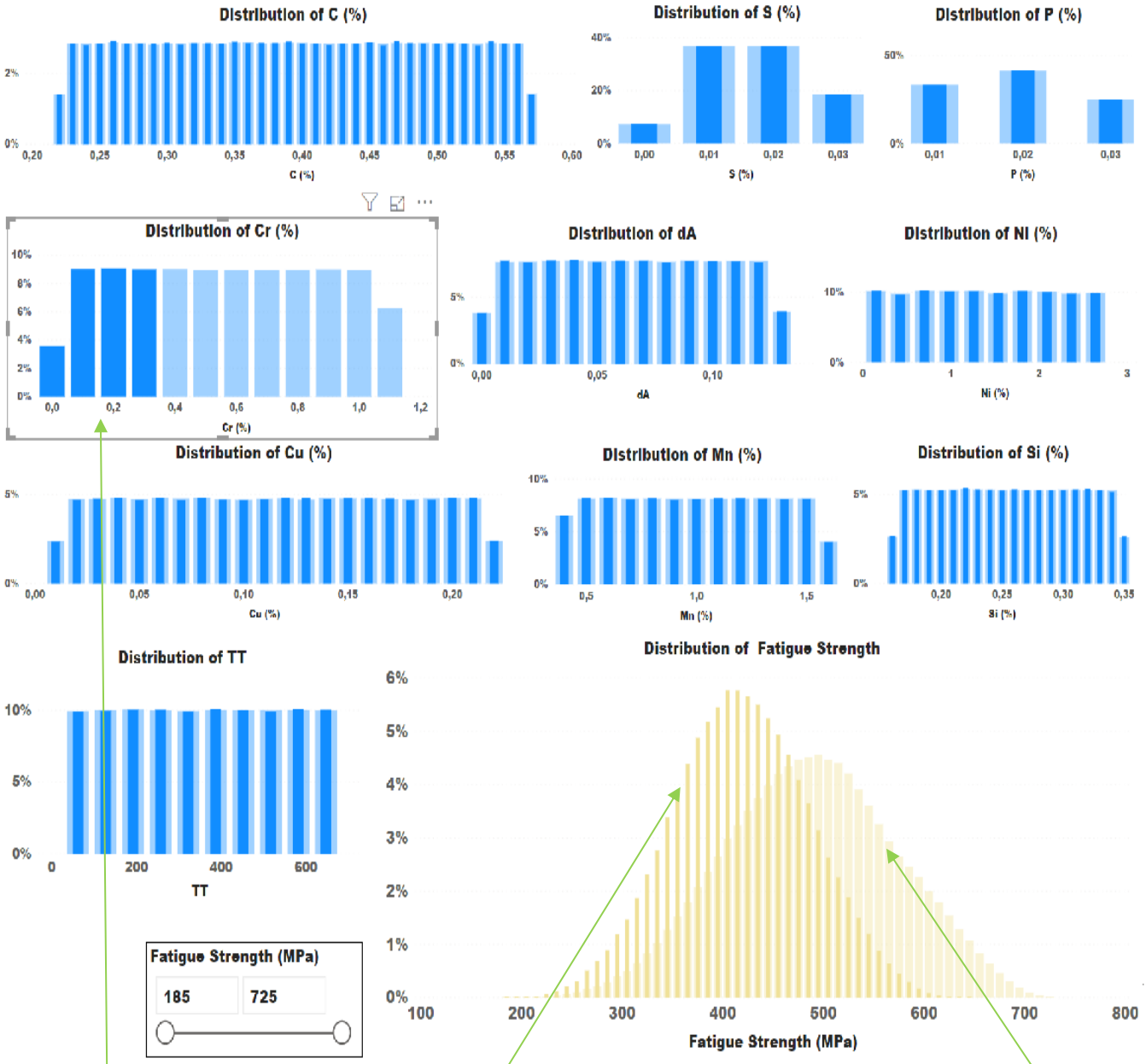

Figure 13. Forward propagation distributions after setting evidence in the input parameter Cr (%).

Posterior distribution (after setting evidence) of fatigue strength

Prior distribution (before setting evidence) of fatigue strength

Evidence setting in input parameter, Cr (%)

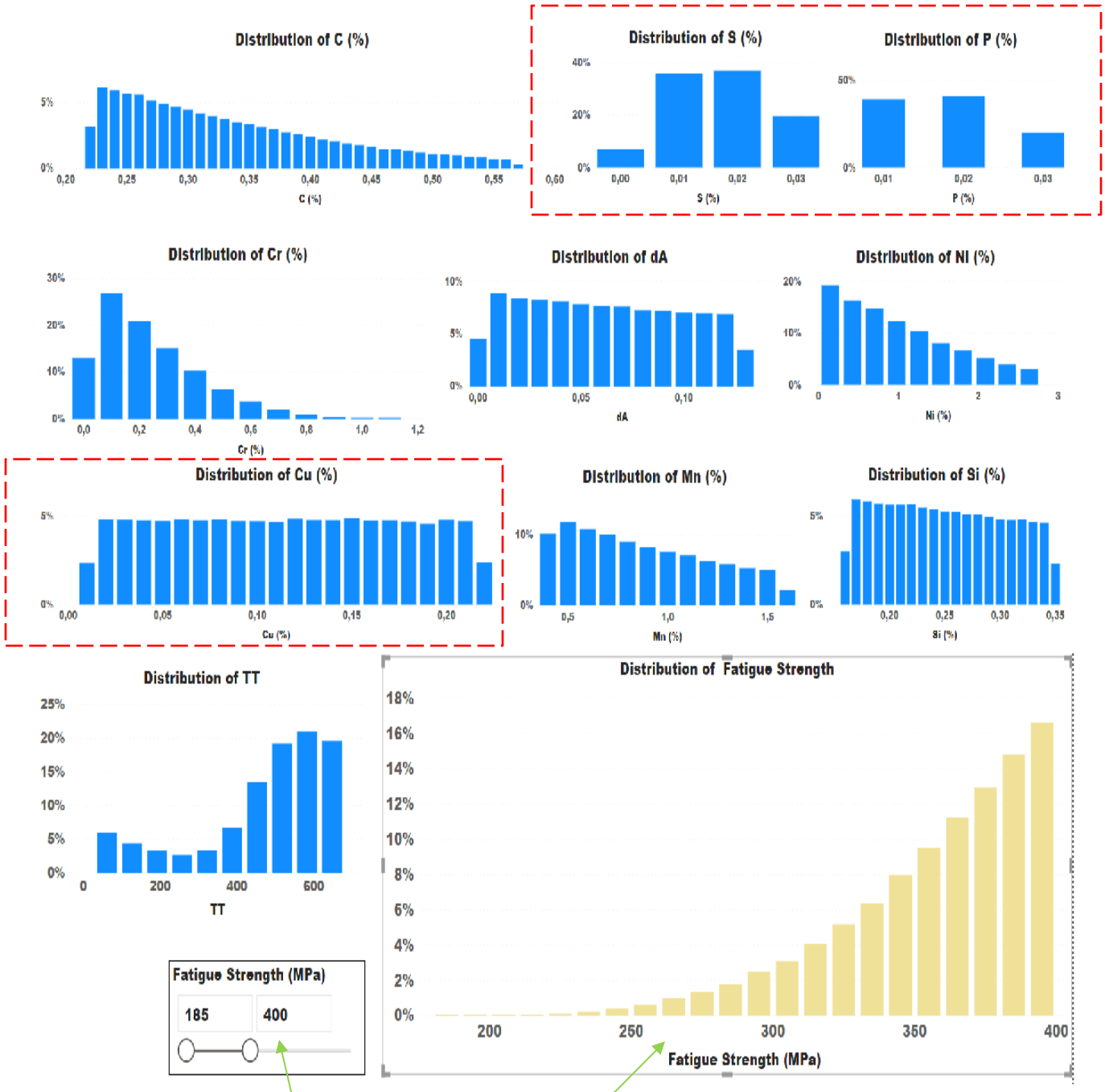

Figure 14. Backward propagation distributions after setting evidence in output parameter, fatigue strength (MPa).

Evidence setting in output parameter, fatigue strength (400 MPa upper threshold value)