Predicting Remaining Fatigue Life of Topside Piping Using Deep Learning

Supratik Chatterjee
Engineering, Research & Development
Wipro Limited
India
supratik.chatterjee@outlook.in

Arvind Keprate
Department of Mechanical, Electronics and Chemical Engineering
Oslo Metropolitan University
Norway
arvind.keprate@oslomet.no

Abstract – **Topside piping is the most commonly failed equipment in the Petroleum and Maritime industry. The prominent degradation mechanism causing piping failure is fatigue which results in unnecessary hydrocarbon release from these assets. In order to avoid the unexpected fatigue failure of piping, it is essential to estimate the remaining fatigue life (RFL) of the aforementioned assets. Generally, engineering companies either rely on experimentally derived SN curves or on a probabilistic fracture mechanics approach to predict RFL. More recently, researchers have utilized surrogate models (classical ML models) to predict the same, and the results seem to be promising. In this manuscript, authors have tried to employ Deep Learning in order to predict the RFL of a topside piping in which crack has been detected. Firstly, different sources of uncertainty in the crack growth process are identified and quantified, with suitable distributions and parameters obtained from the literature. Thereafter, Monte Carlo Simulation is used to generate 5000 samples of the data consisting of 3 input parameters and 1 target feature (RFL class). Afterwards, the data is preprocessed, and feature importance criteria is applied. Finally, a custom Deep Learning model is developed to estimate the RFL class. An accuracy of 0.96 is achieved.**

Keywords - **Deep Learning, Feature engineering, Data preprocessing, Remaining fatigue life, Topside piping**

I. INTRODUCTION

Topside piping contributes significantly towards the hydrocarbon release on offshore platforms in the North Sea [1]. Thus, it is essential to maintain the integrity of vibrating piping by following efficient design practice, high-quality fabrication, good operating practices and risk-based in-service inspection. Despite that, several cases of HCR from offshore piping have been reported. Consequently, operators rely on risk-based inspection (RBI) of topside piping to maintain its technical integrity according to the design. In his previous works second, author have formulated an alternate approach that finds its usage in selecting fatigue critical piping locations on the offshore and onshore oil and gas platforms [2,3, 4].

One of the vital steps in the proposed approach is the estimation of the remaining fatigue life (RFL) of the vibrating piping, using probabilistic crack growth (PCG) analysis and Monte Carlo simulation (MCS), the details of which can be found in [5, 6]. However, MCS is computationally expensive as RFL is generally of orders 105 cycles. Previously, authors have employed various classical machine learning models such as linear regression, gradient boosting, gaussian process regression to estimate RFL [1]. However, in this paper, an alternate approach using Deep Neural Networks is postulated for the prediction of RFL. The difference between the predictions done in [1] and this paper is that in the former paper, RFL prediction was a regression problem; however, in this manuscript, it is a classification problem as the target variable (RFL) has been divided into four classes.

A feedforward wide artificial deep neural network model has been designed to solve this particular supervised classification problem. This model is curated based on the usability of the dataset. As the target variable is categorical in nature and has a cardinality of more than two, sparse categorical cross-entropy has been chosen as a loss function. This, in turn, provides us with the benefit of preserving the non-ordinality of the target variable (as we assumed earlier) while training the model, without creating dummy variables.

The remaining of the manuscript is structured as follows. In Section II, a brief summary to the Deep Learning is presented, followed by an illustrative study in Section III, and a conclusion in Section IV.

II. DEEP LEARNING

In general, deep learning or deep structured learning is a wide group of artificial neural network that imitates human brain's neuron system to find a solution of a complex problem by finding a pattern and relationship among different components. Different authors have defined deep learning in different ways. Deep learning allows computational models composed of multiple processing layers to learn representations of data with multiple abstraction levels [7]. Deep learning algorithms seek to exploit the unknown structure in the input distribution to discover good representations, often at multiple levels, with higher-level learned features defined in terms of lower-level features [8]. A deep neural network (DNN) is an artificial neural network (ANN) with multiple layers between the input and output layers [9, 10]. These techniques are especially useful when the understanding relationship among available information is difficult to achieve. With the advent of DNN, non-linear solutions often outperform conventional methods. Authors are interested in a deep neural network in the exploration of creating a state-of-the-art customized solution for the defined problem, which is comprised of linearly inseparable data. A typical artificial neural network comprised of the following conceptual artificial components, neurons, synapses, weights, biases, and functions is shown in Fig. 1.

DOI: http://dx.doi.org/10.1109/ICAPAI49758.2021.9462055.

III. ILLUSTRATIVE CASE STUDY

A. Data Generation.

The data used in the case study is generated using probabilistic crack growth analysis, the details of which can be found in [6]. The uncertainty quantification of the parameters of interest for this case study is shown in Table I. The uncertain parameters, i.e., initial crack size, remote stress range, and material parameter C, have lognormal distribution with respective mean and standard deviation shown in Table I. Monte Carlo Simulation coupled with Paris law is used to generate 5000 samples of the RFL. The details of this process are already discussed by authors previously in [5]. The snapshot of the generated data set is shown in Fig. 2. The target variable in the dataset (RFL) is then classified into four categories, namely "High", "Medium", "Low", and "Very Low". This is achieved by considering the first quartile RFL values as "Very Low", the second quartile as "Low", the third quartile as "Medium" and the fourth quartile as "High". This classification is performed in order to convert the regression problem into a classification problem, wherein the target variable (RFL) is not a continuous number rather a class.

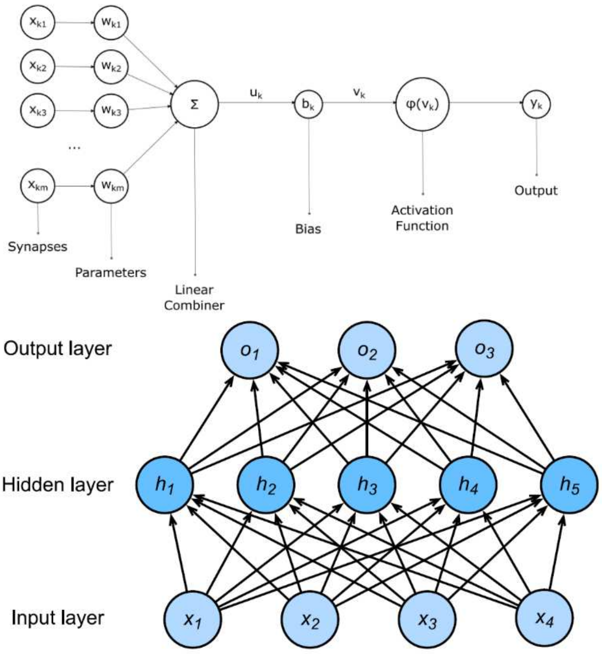
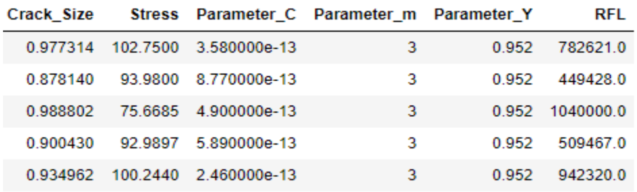


Fig. 1. Basic architecture of DNN [11].

TABLE I
UNCERTAINTY QUANTIFICATION FOR RFL ESTIMATION

Uncertainty Source	Random Variable	Parameters Value
Initial crack size	a_0 (mm)	LN (0.94, 0.05)
Material parameter	C	LN (5.22e-13, 2.07e-13)
Material parameter	m	3
Remote stress range	$\Delta\sigma$ (MPa)	LN (100, 10)
Critical crack size	a_{cr} (mm)	4.5
Geometric function	Y	0.952



Crack_Size	Stress	Parameter_C	Parameter_m	Parameter_Y	RFL
0.977314	102.7500	3.580000e-13	3	0.952	782621.0
0.878140	93.9800	8.770000e-13	3	0.952	449428.0
0.988802	75.6685	4.900000e-13	3	0.952	1040000.0
0.900430	92.9897	5.890000e-13	3	0.952	509467.0
0.934962	100.2440	2.460000e-13	3	0.952	942320.0

Fig. 2. First five rows of generated dataset.

B. Exploratory Data Analysis.

To summarize the characteristics of the data, specific explorations have been performed. Four classes, namely "High", "Medium", "Low", and "Very Low" are distributed uniformly. Therefore, weights or importance of each class can be equal as well. Feature "Parameter_C" has extreme low range of values, minimum and maximum values are 1.160000e-13 and 1.940000e-12, while, "Stress" feature has a significantly higher range than others. Highest being 150.87 and 72.12. This extreme disparity between value ranges of features and extreme may lead to prefer one feature over the other while training, as features, are cardinal in nature. Misinterpretation of further calculations on "Parameter_C" is very much possible due to its proximity to zero. To suppress the effect of miscalculation, before making any further observations, scaling of data to an understandable fixed range was necessary. A range of [0, 1] has been chosen.

It can be perceived upon plotting kernel density estimation (KDE) of the scaled features that all distributions are skewed towards the right, as shown in Fig. 3A. Skewness of the features "Crack_Size", "Stress" and "Parameter_C" are 0.134745, 0.323129 and 1.197923 respectively. Kurtosis has been measured as 0.165000, 0.219906 and 2.658968 is the same order of skewness. This implies that transformation of distribution to gaussian alike distribution might help in faster convergence of the neural network, as any natural phenomena follow the gaussian distribution. Furthermore, the Box plot of Fig. 3B. implies that features are highly spread and consists a large number of outliers. Therefore, a robust transformation scheme should be used.

Likewise, authors checked the non-linear correlation among features to discard highly correlated features. For this purpose, authors have used pairwise scatter plot with hue as defined classes. It is observed that features are not correlated with each other. Graphical representation of the same can be found in Fig.4. Even though target is ordinal in nature, treating individual types of data as separate entity, i.e., non-ordinal may increase chance of better segregation of individual instances. With this assumption, a fixed random categorical integer number y, for each target class type has been mapped based on number of available class labels, ℓ. from the set $L = \{y \mid y \in \mathbb{N}_0 \wedge x < \ell\}$. |L| is 4 in this case.

As there are more than two possible categories of target class labels, correlation ratio has been considered while trying to understand association among categorical target variable and nominal features. As shown in Fig 5. all features are poorly correlated with target, especially effect of "Crack_Size" is insignificant in contrast with other two. Therefore, authors decided to generate more features from existing feature columns to make more accurate model.

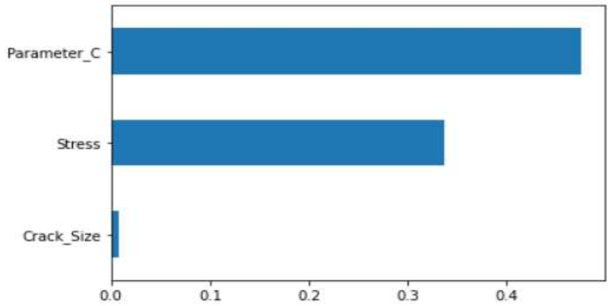


Fig. 5. Correlation of input features to target feature.

C. Preprocessing: Feature generation, scaling, transformation and feature selection.

Scaling by removing the mean and scaling to unit variance, i.e., using standard scaler, will not produce the desired result because the value of parameter C is of order 10^-13. Therefore, the authors have employed a different methodology for scaling, with a fixed range of [0, 1].

As for feature generation, authors targeted single instance prediction rather than holding unseen instances for computing additional features, such as rolling statistics. Therefore, polynomial features of up to degree three with product interaction features have been considered. Generated features were 19 in number, which includes the original three features as well. Using all generated features may lead to the curse of dimensionality. Consequently, the authors decided to further reduce the number of features by eliminating any feature less than the median of correlations among target classes and generated features. Here, the authors used correlation ratio for calculating correlation, same as before. Correlations have been computed after applying transformation as well, as the selected transformation process changes the correlations. The process of comparison on selected transformation has been discussed in subsequent sections.

At this point, the available population have been split into two parts – 60% training data and 40% holdout data for testing. The authors made sure to shuffle the data before doing the split. A stride split has been stratified on the target variable to make sure each part of the split has proportionated number of class instances. Any computation other than evaluation of the model did not use the test data. This is being done to avoid data leakage while training the scaler and transformer.

To transform the generated features into a Gaussian distribution, a robust transformation scheme, quantile transformation, with a number of quantiles as the number of available instances have been used on the scaled data. It has been observed that if we apply the transformation before applying a scaling scheme, all values of “Parameter_C” becomes a constant value. Therefore, scaling before applying transformation was necessary, as mentioned previously. The first two diagrams from the left of Fig. 6 represent the comparative distribution of feature “Parameter_C”, if scaling is applied before transformation and if not. It is to be observed that transformed data now ranges between -5.12 to +5.2. The second scaling of fixed range [-1, 1] has been applied on top of the quantile transformation to further reduce the spread of data. The last diagram in the rightmost of the three diagrams in Fig. 6 shows the final distribution of preprocessed data.

A comparison on central tendency, arithmetic means of correlation of generated features and target, between before and after applying transformation and scaling, reveals this process certainly increases correlation among numerical features and categorical target. Values were measured as 0.468 and 0.567, respectively. Now, the number of features has been trimmed based on the median of target correlation, as described before. Generated features, inversely sorted correlation among features and selected features after trimming operation can be seen in Fig. 7. The remaining features are now a mix of low, medium and high correlation and 10 in total, which is fairly acceptable number of features to be used in the context of deep learning.

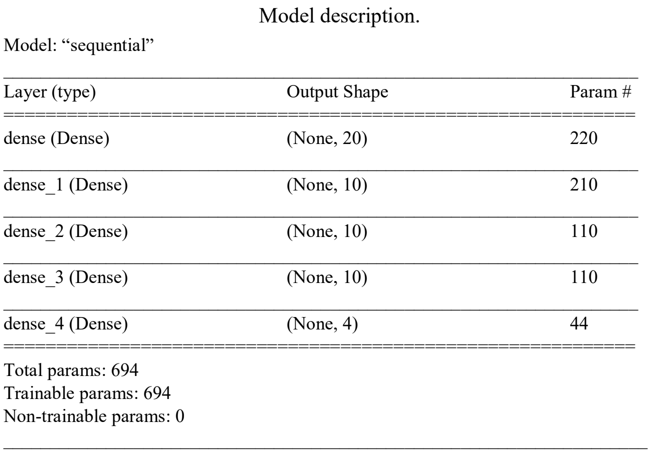
D. Model development, Training and Evaluation

A feedforward wide artificial deep neural network model has been designed to solve this particular supervised classification problem. This model is curated based on the usability of the dataset. As the target variable is categorical in nature and has a cardinality of more than two, sparse categorical cross-entropy has been chosen as the loss function. This, in turn, provides us with the benefit of preserving the non-ordinality of the target variable (as we assumed earlier) while training the model without creating dummy variables. The metric of performance was set as sparse categorical accuracy. Each layer used bias and initialized to 0. As for optimizer, Adabound with AMSGrad - a stochastic optimization method, variable learning rate - initial learning rate as 0.001 and final learning rate of 0.01 was used. Table II describes various other parameter aspects of each layer.

TABLE II

Parameters for layers in DNN

Layer type	Input/Output dimension	Activation	Kernel initializer	Number of output neuron units
Input	10	Leaky ReLU (alpha = 0.1)	He Normal	20
Hidden	NA	Leaky ReLU (alpha = 0.1)	He Normal	10
Hidden	NA	Leaky ReLU (alpha = 0.1)	He Normal	10
Hidden	NA	Leaky ReLU (alpha = 0.1)	He Uniform	10
Output	4	Softmax	NA	4



Model description.

```
Model: “sequential”
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 20)                220
_________________________________________________________________
dense_1 (Dense)              (None, 10)                210
_________________________________________________________________
dense_2 (Dense)              (None, 10)                110
_________________________________________________________________
dense_3 (Dense)              (None, 10)                110
_________________________________________________________________
dense_4 (Dense)              (None, 4)                 44
=================================================================
Total params: 694
Trainable params: 694
Non-trainable params: 0
_________________________________________________________________
```

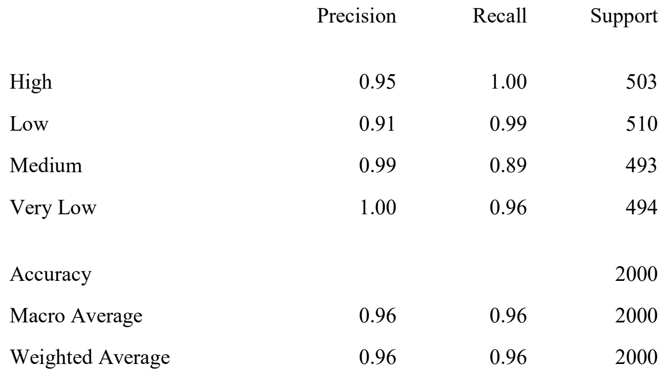
Fig. 8. Summary of neural network design.

It is to be noted that the number of output neuron units and input dimension directly depend on number of used features, rather than using a predefined value. For input layer, output neuron unit is double of input dimension. Next three hidden layers uses same number of output layers of input dimension. Fig. 8 describes summary of the created model.

In the training process, authors used following settings to train the model - Number of epochs: 500, batch size: 2, shuffled the training data before each epoch, another 20% of training data has been used as validation data during training. To avoid overfitting, a monitoring function which will check change in validation loss after end of each epoch has been used. Minimum change in the validation loss as an improvement has been set as 0.1, with minimum patience 20, i.e., if validation loss is not decreasing by 0.1 after 20 epochs then the training will stop automatically and will restore weights automatically to the last known minimal validation loss. Line graphs of Fig. 9A and Fig. 9B shows smoothed values of loss function and performance metrics after each passed epochs, for both training and validation data. Training was early stopped by callback monitoring function after around 53 epochs. Training & validation loss function smoothed values has been observed as

0.1219 and 0.1291 respectively. Similarly, performance metrics smoothed values of training and validation has been measured as 0.9488 and 0.9481 respectively. Training has been executed only once.

To mimic real life performance of the model, accuracy of each class has been measured on the 40% holdout dataset. A detailed report can be seen in Fig. 10. Overall accuracy was 0.96, which is acceptable. Therefore, authors did not change any other parameter.



	Precision	Recall	Support
High	0.95	1.00	503
Low	0.91	0.99	510
Medium	0.99	0.89	493
Very Low	1.00	0.96	494
Accuracy			2000
Macro Average	0.96	0.96	2000
Weighted Average	0.96	0.96	2000

Fig. 10. Recreated classification report on test (holdout) data set.

IV. CONCLUSION

Since fatigue failure of the topside piping is a common source of hydrocarbon release and leakages in the Oil & Gas and maritime sector, therefore it is vital to estimate the remaining fatigue life (RFL) of the aforesaid component. The manuscript posited the use of a specialized deep neural network to classify RFL of topside piping. Even though available parameters had little correlation with the target, with the help of feature engineering, the authors achieved a result with an accuracy of 0.96. Besides that, special treatment of scaling on the parameter "C", which was in the vicinity of absolute zero, making it possible to increase the importance of this feature, rather than discarding it all together, thus resonating with the physics of crack growth analysis.

DECLARATION

The views expressed in this article/presentation are the author's own and Wipro does not subscribe to the substance, veracity, or truthfulness of the author's views. Conflict of interest declaration number 13153 has been acknowledged and accepted by Wipro on 12/04/2021.

REFERENCES

[1] A. Keprate and R. M. C. Ratnayake, "Remaining Fatigue life prediction of topside piping using response surface models," *IEEE International Conference on Industrial Engineering and Engineering Management,* Bangkok, Thailand, 2018.

[2] A. Keprate, and R. M. C. Ratnayake, "Enhancing offshore process safety by selecting fatigue critical piping locations for inspection using Fuzzy-AHP based approach," *Process Saf. Environ.*, vol. 102, pp. 71-84, 2016.

[3] A. Keprate and R. M. C. Ratnayake, "Selecting fatigue critical inspection location of offshore topside piping using fuzzy AHP framework," *IEEE International Symposium on Applied Machine Intelligence and Informatics,* Herlany, Slovakia, 2016.

[4] A. Keprate and R. M. C. Ratnayake, "Generic approach for risk assessment of offshore piping undergoing fatigue degradation," ASME *J. Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering*, vol. 4, 2018.

[5] A. Keprate, R. M. C. Ratnayake, and S. Sankararaman, "Minimizing hydrocarbon release from offshore piping by performing probabilistic fatigue life assessment," *Process Saf. Environ.*, vol. 106, pp. 34–51, 2017.

[6] A. Keprate and R. M. C. Ratnayake, "Handling uncertainty in the remnant fatigue life assessment of offshore process pipework", *International Mechanical Engineering Congress and Exposition*, Phoenix, Arizona, USA, 2016.

[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning", Nature 521, 436–444, 2015.

[8] Bengio, Y.. (2012). Deep Learning of Representations for Unsupervised and Transfer Learning. Proceedings of ICML Workshop on Unsupervised and Transfer Learning, in PMLR 27:17-36

[9] Schmidhuber, J. (2015). "Deep Learning in Neural Networks: An Overview". Neural Networks. 61: 85–117.

[10] Bengio, Yoshua (2009). "Learning Deep Architectures for AI" Foundations and Trends in Machine Learning. 2 (1): 1–127.

[11] A. Zhang, Z. C. Lipton, M. Li, and A. J. Simola. "Dive into Deep Learning", Release 0.16.1., 2019.

APPENDIX

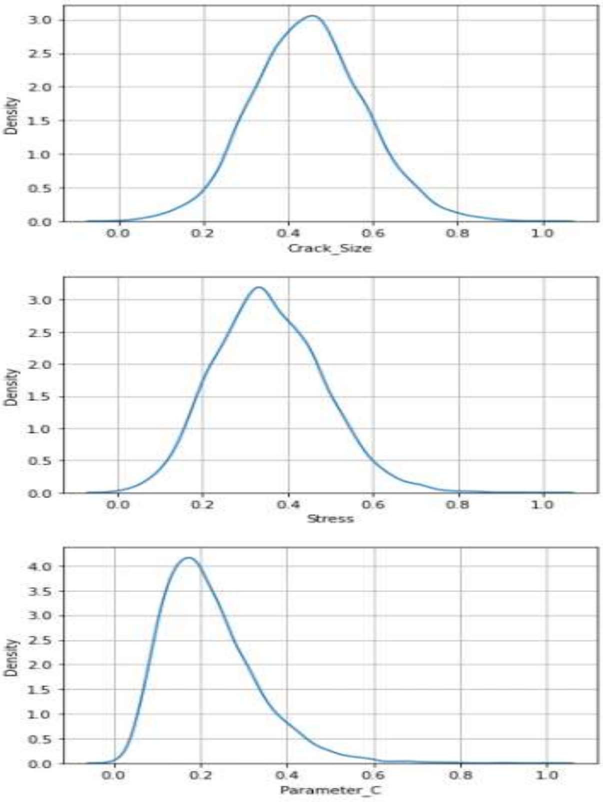


Fig. 3A. Distribution of scaled features.

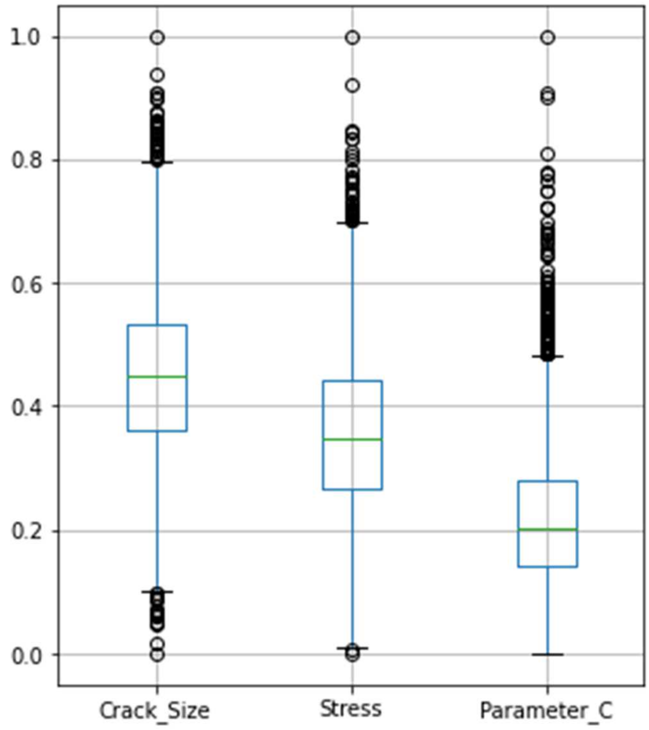


Fig. 3B. Spread of scaled features.

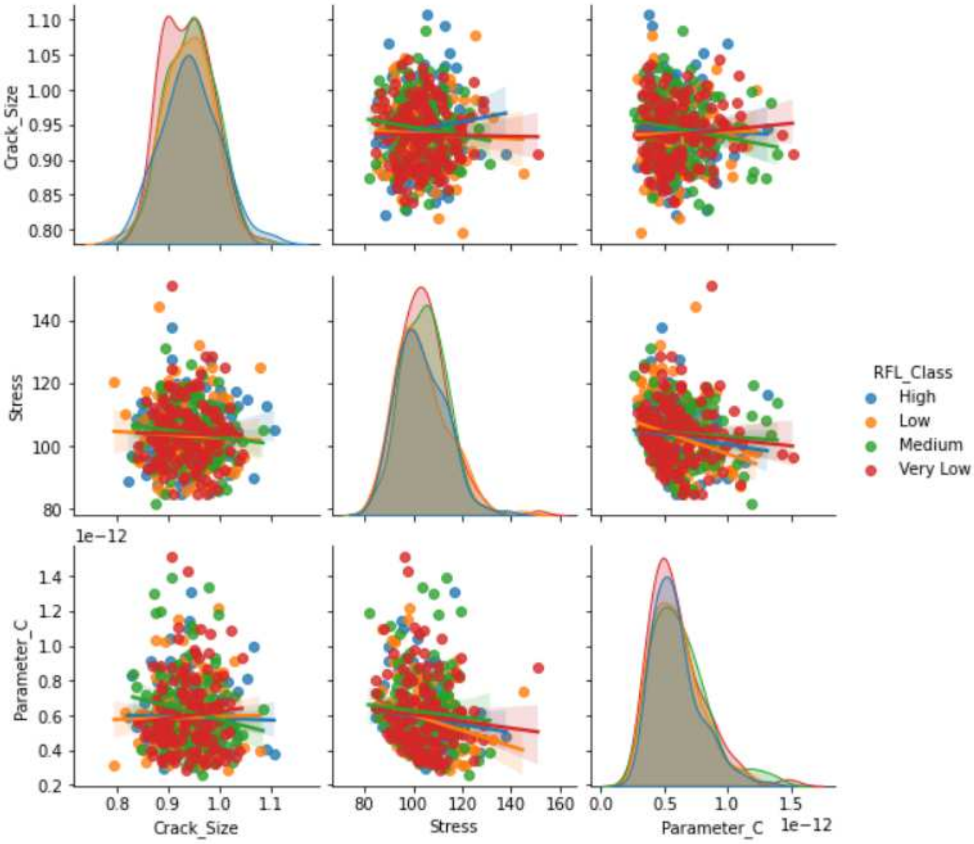


Fig. 4. Feature pairwise scatter plot.

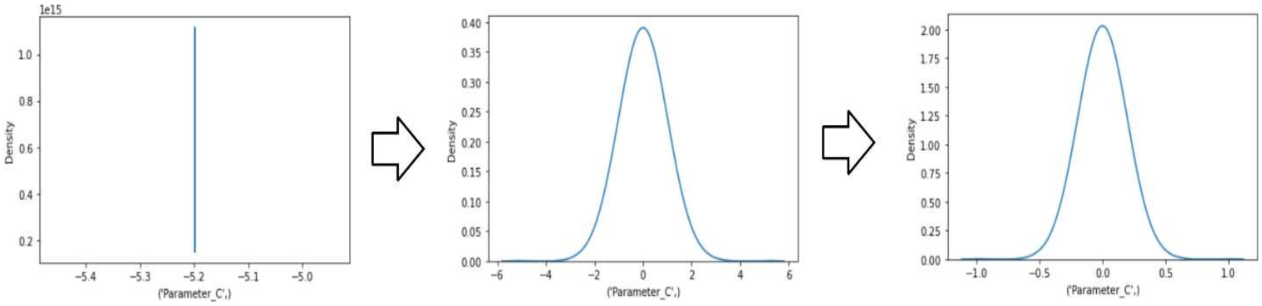


Fig. 6. Quantile Transformation, Before and After fixed boundary (Min-Max) scaling.

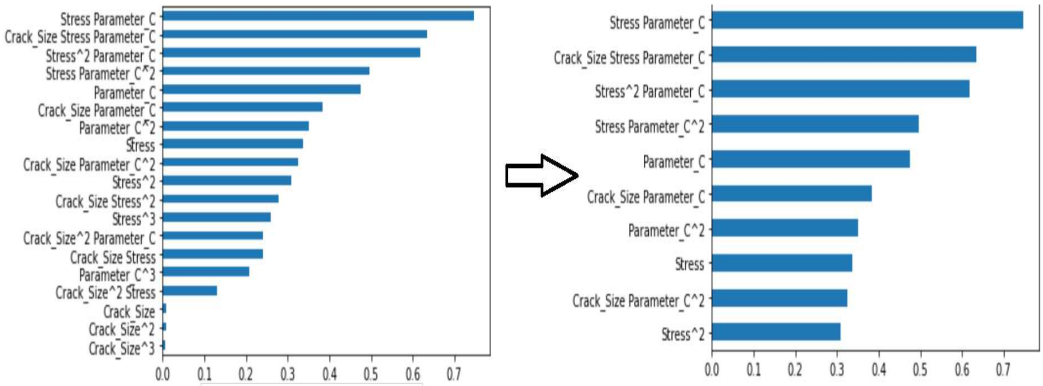


Fig. 7. Feature Selection: Correlation of generated features and target.

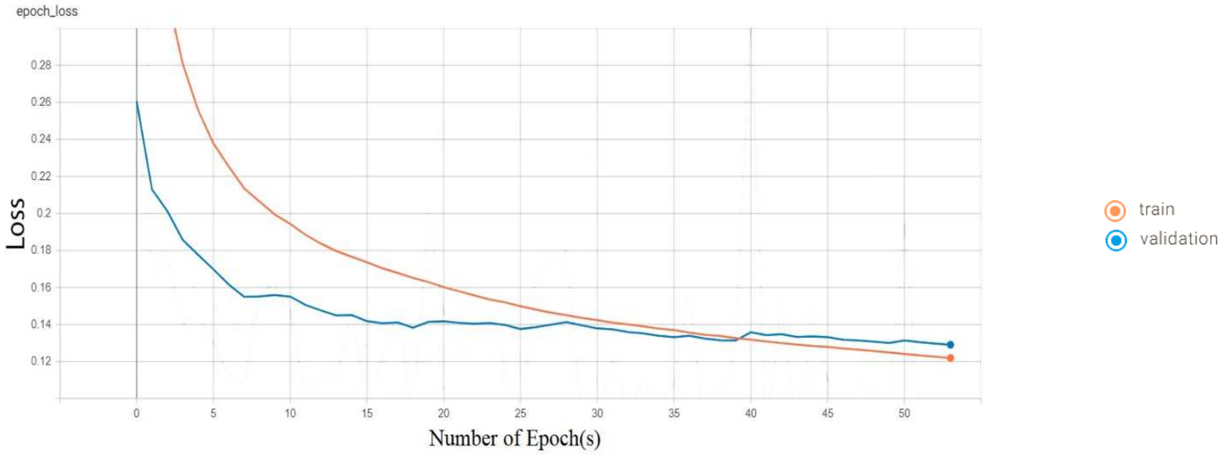


Fig. 9A. Epoch vs. sparse categorical cross-entropy loss function.

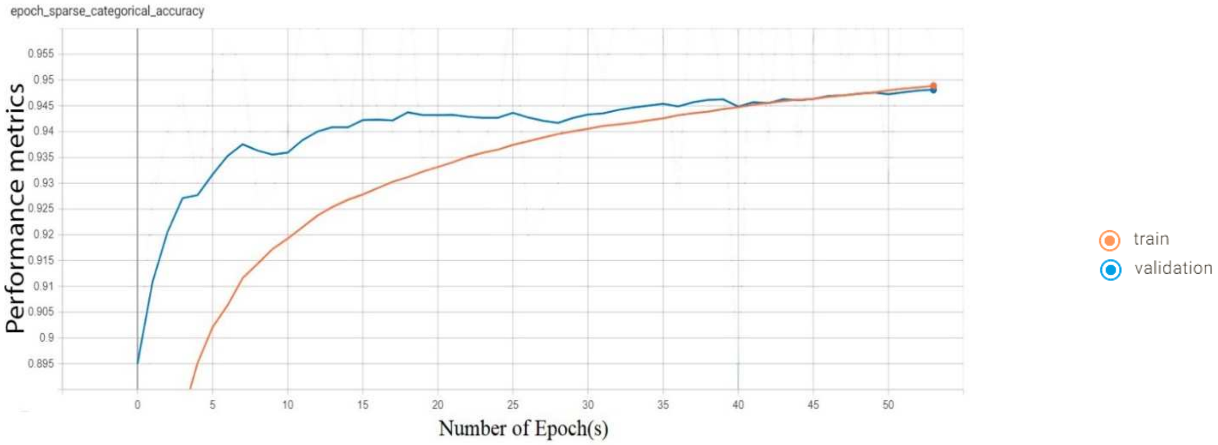


Fig. 9B. Epoch vs sparse categorical accuracy.