

Article citation info:

Badora M, Sepe M, Bielecki M, Graziano A, Szolc T. Predicting length of fatigue cracks by means of machine learning algorithms in the small-data regime. *Eksploracja i Niezawodność – Maintenance and Reliability* 2021; 23 (3): 575–585, <http://doi.org/10.17531/ein.2021.3.19>.

Predicting length of fatigue cracks by means of machine learning algorithms in the small-data regime

Indexed by:



Maciej Badora^{a,b,*}, Marzia Sepe^c, Marcin Bielecki^b, Antonino Graziano^c, Tomasz Szolc^a

^aInstitute of Fundamental Technological Research, Polish Academy of Sciences, ul. Pawińskiego 5b, 02-106 Warsaw, Poland

^bBaker Hughes, BH Poland sp. z o.o., Aleja Krakowska 110/114, 02-256 Warsaw, Poland

^cBaker Hughes, Via Felice Matteucci 2, 50127 Florence, Italy

Highlights

- Machine learning algorithms can be utilized effectively in the small-data regime.
- The length of fatigue cracks can be predicted based on operational data of the engine.
- The lowest root mean squared error (RMSE) is achieved with the AdaBoost.R2 algorithm.
- Small datasets should be processed in a fully controlled manner to get valuable results.
- A custom cost function may favor certain solutions and drives capabilities of the model.

Abstract

In this paper several statistical learning algorithms are used to predict the maximal length of fatigue cracks based on a sample composed of 31 observations. The small-data regime is still a problem for many professionals, especially in the areas where failures occur rarely. The analyzed object is a high-pressure Nozzle of a heavy-duty gas turbine. Operating parameters of the engines are used for the regression analysis. The following algorithms are used in this work: multiple linear and polynomial regression, random forest, kernel-based methods, AdaBoost and extreme gradient boosting and artificial neural networks. A substantial part of the paper provides advice on the effective selection of features. The paper explains how to process the dataset in order to reduce uncertainty; thus, simplifying the analysis of the results. The proposed loss and cost functions are custom and promote solutions accurately predicting the longest cracks. The obtained results confirm that some of the algorithms can accurately predict maximal lengths of the fatigue cracks, even if the sample is small.

Keywords

This is an open access article under the CC BY license (<https://creativecommons.org/licenses/by/4.0/>)

empirical models, fatigue cracks, predictive maintenance, regression analysis, small data, statistical learning, turbomachinery.

1. Introduction

In the last decade repair intervals of certain parts of gas turbines, which are installed in the combustor or turbine sections, have been extended by 100%. Constantly growing energy needs force operators of petrochemical and power plants to reduce downtime and maximize production. Many companies sign a Long-term Service Agreement with the engine's OEM to assure seamless execution of the optimized maintenance plan, with availability and productivity guarantees. The value delivered by the manufacturer is based on the knowledge, experience and data accumulated over the years. Combining this with selected operating parameters of the units allows to develop predictive models, which support data-driven decisions and provide actionable insights to the end users [6]. Due to the complexity of the analyzed system, numerous estimators are built and utilized, e.g., for diagnostic and anomaly detection [17, 13], classification [2], regression, or as a synthetic sensors [33]. A structured collection of such models reflecting part-to-part interactions and fed with operational data is referred to as the digital twin of the gas turbine [30].

Survival analysis for rotating equipment is typically carried out on limited datasets due to the low occurrence of failures, long duration

and high costs associated with the destructive tests. These constraints set Weibull analysis for years as the default method for forecasting failure probability and damage size [1]. Scientific papers on the correct execution of regression analysis in the small-data regime [28] have not disappeared in the big data era. New contributions have been published [23, 14] and are needed to provide guidance on how to use machine learning algorithms, which were not so commonly used in the past, for predictive modeling based on small samples. Such studies will be executed regardless of the growth of big data, but with the development of new data infrastructures the small data might be pooled and scaled more frequently [26].

Fatigue is among the most frequent causes of failures in mechanical systems. Deep knowledge of the fatigue crack growth mechanism is needed for the design and maintenance of gas turbines, especially with rising expectations regarding the efficiency and comprehensive optimization of the production process. Since the groundbreaking contribution of Paris and Erdogan [32], numerous fatigue crack growth equations have been published and these have been summarized well in [4]. For the most part, these models are based on experimental data and vary in complexity. There is no universal approach to fatigue life prediction; thus, for a particular problem, the optimal method should

(*) Corresponding author.

E-mail addresses: M. Badora - mbadora@ippt.pan.pl, maciej1.badora@bakerhughes.com; M. Sepe - marzia.sepe@bakerhughes.com; M. Bielecki - marcin.bielecki@bakerhughes.com; A. Graziano - antonino.graziano@bakerhughes.com; T. Szolc - tszolc@ippt.pan.pl

properly balance the accuracy and computational costs. Usually, the finite element method is used to solve the crack growth equation. Nevertheless, discretization of the domain at each iteration is a challenge, thereby alternative approaches are utilized, e.g.: by the extended finite element method, the boundary element method, and their hybrids and meshless methods [36].

The growth in computational power, the availability of open-source software and simple, user-friendly libraries has resulted in the enormous popularity of machine learning in recent years. These methods have also been adapted to estimate the fatigue life, while the artificial neural network (ANN) is the most widely used algorithm [21]. The ANN was successful in including the effect of mean stress for fatigue damage prediction [12] [22] or the effects of load sequences and temperatures on the fatigue life [20]. Attempts to utilize regression trees [27], random forests [43], kernel-extreme learning machine [15], or Bayesian network [37] can be found in recent publications. Typically, the models are based on experimental data, though outputs of a finite element analysis were used for the training [10] [41]. Data-driven estimators are effective for complex, high-dimensional problems, where analytical solutions do not exist. A fully specified model responds quickly, therefore it can be installed in the edge devices or utilized for real-time monitoring of the damage. However, these methods also have certain drawbacks, e.g., a variety of challenges caused by small samples, time-consuming optimization, poor extrapolation abilities, or a lack of interpretability [31]. In this contribution the authors try to address these problems, providing a tutorial on the execution of regression analysis by means of machine learning algorithms in the small-data regime.

2. The Analyzed Object and Problem Setup

This paper focuses on the 1st stage Nozzle (S1N) of a heavy-duty gas turbine. Three different engine configurations are analyzed and referenced as Type A, Type B and Type C. The Nozzle assembly has 18 segments cast from a cobalt-based alloy FSX-414. A compressor discharge airflow is used to cool the part by means of a cooling insert in the airfoils and cooling holes at the leading and trailing edges. The analyzed object is shown in Fig. 1 and Fig. 2 [38].

Failure modes recorded during intermediate inspections of the component are as follows:

- cracks, oxidation and erosion of airfoils/platforms,
- corrosion of airfoils due to contaminants,
- deformation of airfoils due to creep.

A degradation of the surface of S1N due to oxidation, erosion, deformation, or corrosion results in the gradual loss of efficiency of the gas turbine. However, these negative effects should be captured by the operator, analyzed by a cross-functional team, and properly addressed. In the case of cracks, the standard instrumentation provides

no direct indications of the presence and size of the damage. If the engine is not maintained as per the OEM guidelines, S1N cracks may even lead to an unplanned outage. Examples of trailing edge cracks of the analyzed Nozzle are shown in Fig. 3. The damage was found in locations where the trailing edges have the highest temperatures. The main cause of this damage are thermo-mechanical stresses, which are maximal during transient states, caused, e.g., by engine startups and shutdowns. Some locations are also subjected to tension during steady-state operations.

The ability to evaluate the length of crack without having to stop the engine and inspect the parts may result in substantial benefits for the owner, such as:

- avoiding component repair by using the Condition Based Maintenance approach,

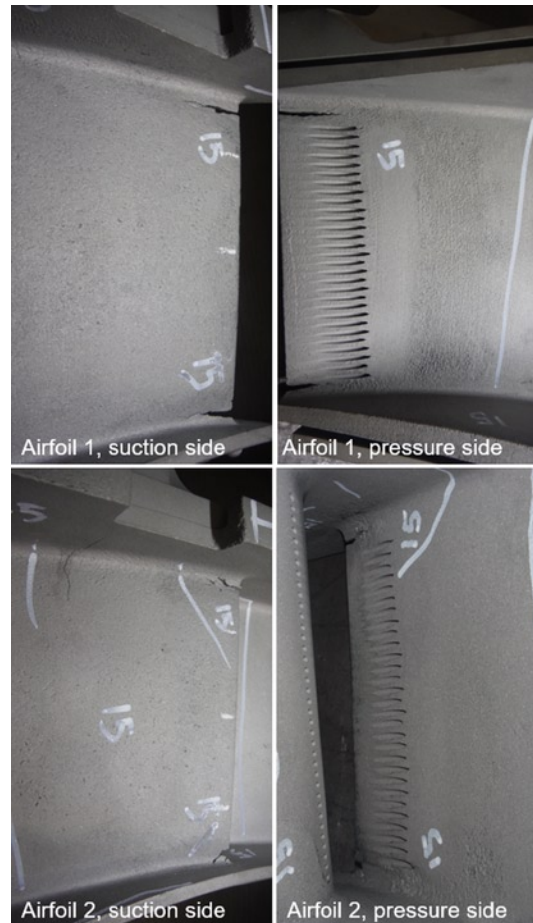


Fig. 3. Examples of S1N trailing edge cracks after service

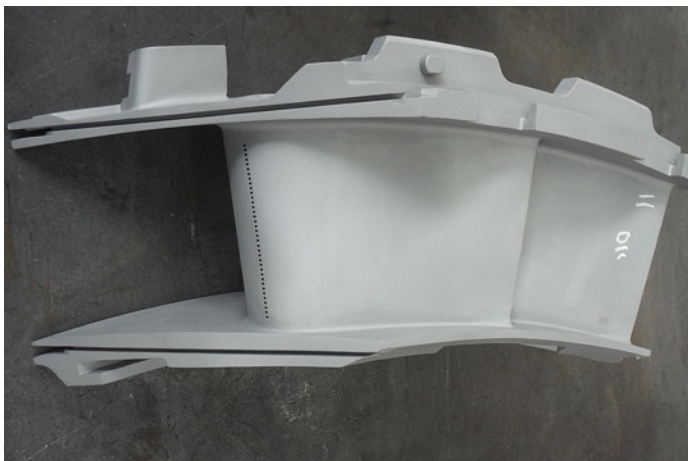


Fig. 1 Suction side view of the 1st stage Nozzle

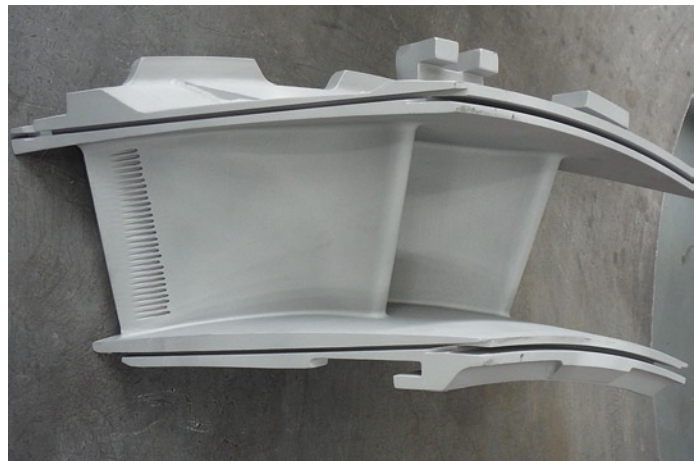


Fig. 2. Pressure side view of the 1st stage Nozzle

- optimizing the maintenance schedule enabled by real-time evaluation of the damage size,
- forecasting the scope of work for future outage(s),
- making data-driven decisions.

3. An Overview of Available Empirical Data

The utilized data are the proprietary property of Baker Hughes Company LLC and cannot be published. Data provided in this paper allow the reader to understand the context and the decisions made by the authors. The analyzed positions on trailing edges of the Nozzles are referenced as Position 1, 2, 3 and 4, without disclosing any further details.

The first piece of utilized input data contains the following:

- numbers of fired hours (FH), fired starts (FS) and emergency shutdowns (ESD) accumulated by each segment since the last repair and since the part was manufactured,
- measurements taken and damages observed during repair activities, after the operation,
- configuration of the Nozzle and the engine(s) where the part previously operated.

Trailing edge cracks were found on 754 out of 868 (87%) S1N segments subjected to the repair. The longest cracks, or less than 5 mm shorter than the longest, were recorded at Position 2 for 640 out of 754 segments (85%). Modeling of the crack lengths at Position 2 shall be prioritized with respect to the remaining positions (Fig. 4).

Gas turbines are equipped with a wide range of measuring instruments. Remote Monitoring and Diagnostic services are used to record the time series of operational parameters. These random variables are utilized to predict damage to parts. The operational data are available for 555 out of 868 segments (31 Nozzle sets) subjected to the repairs. The sampling interval is set to 1 hour. The optimal interval allows for the correct modeling of dynamics of the failure mode, but does not simultaneously enlarge the dataset excessively. A preprocessing focuses on the removal of erroneous data and variables correlated with each other. The second piece of utilized input data contains the following time series:

- pressure, temperature, and relative humidity of ambient air,
- pressure and temperature of air at the discharge of the axial compressor,
- pressure losses in the inlet and exhaust ducts,
- axial compressor pressure ratio,
- mean temperature of gases at the inlet and outlet of the S1N assembly, and output of the engine (these are calculated parameters),

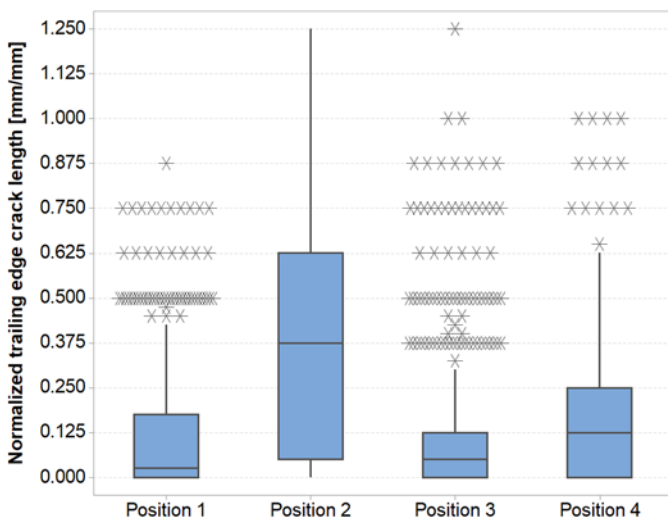


Fig. 4. Box plot of the lengths of trailing edge cracks of the Nozzles. The data are normalized by dividing them by the maximal measured crack length

- temperatures of exhaust gases and their spread,
- speeds of the high-pressure and low-pressure shafts,
- positions of the Inlet (or Nozzle) Guide Vanes.

Missing readings of ambient air parameters are completed with the data published online [44]. For each of the variables a value range is defined beyond which the reading is considered either incorrect or related to transient states. Such records are removed from the dataset.

The set of operational data is composed of 1,029,215 records in total, although for some periods of the Nozzles' service time the data are unavailable. The coverage varies from 26% to 100% with the average equal to 73%. The Nozzles operated in units that drive centrifugal compressors in the process of natural gas liquefaction. Such engines operate at base loads with a very stable operational profile. Therefore, it is assumed that the available data also describe the missing periods well.

4. Feature Selection

Long cracks of the Nozzles may jeopardize the availability of the gas turbine; hence, the maximal crack size at Position 2 is selected as the dependent variable. Each Nozzle set is labeled with just one value, composing the sample with 31 data points. The complete input dataset has 32 independent variables; thus, to avoid overfitting, to reduce dimensionality and to make the models interpretable [18, 24], feature selection is executed. The maximal number of predictors is set to 5, which gives ~6 subjects per feature. The predictors are the same, regardless of the regression method used, thus simplifying the interpretation of the results.

The time series of operating parameters related to each observation are simplified to medians. This transformation greatly reduces the dataset and is performed to pool together the different types of inputs previously described. It is justified for the analyzed units, which operate in steady-state conditions.

The features are selected using the Scikit-learn library [34] based on the simultaneous evaluation of mutual information, Pearson's and Spearman's correlation coefficients, and an analysis of chi-square tests. The removal of the least important features is done iteratively (i.e. ~20% of remaining variables are removed after each iteration) until 10 independent variables are identified.

The next step concerns feature engineering. The features are computed based on the time series of the operating parameters previously filtered. The variables reflect the distributions of these parameters in a simplified way. For each of the time series, considering all the records, certain statistical measures are calculated (i.e. 50th, 70th, and 90th percentile). The new features are computed as the number of service hours with a reading below the 50th percentile, between the 50th and 70th percentiles, between the 70th and 90th percentiles, and higher than the 90th percentile. The enlarged set is again iteratively reduced to 10 variables.

The last step is about the application of a wrapper method to find the optimal combination of the features but using a specific algorithm. A random forest is chosen given its data-driven nature of decorrelated trees and leveraging the law of large numbers [5]. Each iteration removes one variable based on the importance of the features. Additionally, a 5-fold cross-validation (CV) with 50 repetitions is used during each iteration. The following features are selected for the modeling:

- median of average exhaust temperatures, \tilde{T}_{EXH} ,
- number of fired starts accumulated by the part since the manufacturing, TFS ,
- number of service hours with the mean temperature of gases at the inlet of the S1N assembly between the 50th and 70th percentiles, $Tin_{P50-P70}$,
- median of ambient air temperatures, \tilde{T}_{AMB} ,
- median of gas turbine output, \tilde{P} .

These predictors are consistent with the physical phenomena that cause the damage. Thermal stresses σ are proportional to the Nozzle's material temperature change, ΔT_{MATL} :

$$\sigma = E\alpha\Delta T_{MATL} = E\alpha(\phi_{COLD} - \phi_{HOT})(T_{in} - T_{COOL}) \quad (1)$$

where E is the Young's modulus, α is the thermal expansion coefficient, ϕ_{COLD} and ϕ_{HOT} are the cooling effectiveness coefficients on the cold/hot side of the part, T_{in} is the mean temperature of gases at the S1N assembly inlet, and T_{COOL} is the coolant temperature. A specific range of T_{in} is present in the set of predictors, while T_{COOL} is strongly correlated with the gas turbine output and the ambient temperature. TFS is considered as the number of load cycles, an inherent term in any fatigue crack growth equation.

5. Analysis Setup

As per [19], the expected mean squared error (MSE), for a given test value x , can be decomposed to the sum of variance of $\hat{f}(x)$, the squared bias of $\hat{f}(x)$, and the irreducible error $Var(\varepsilon)$:

$$E[y - \hat{f}(x)]^2 = Var[\hat{f}(x)] + [bias(\hat{f}(x))]^2 + Var(\varepsilon) \quad (2)$$

Minimization of the MSE leads to simultaneous minimization of the variance and bias to find a trade-off between them. A robust approach to regression analysis is to split the data into three subsets: the training set used to fit the models, the validation set used to tune the parameters of the models, and the test set used to assess the generalization capabilities and performance of the fully specified model. The sample has just 31 records, therefore an appropriate data splitting method is required to properly balance the sets and to avoid errors that may affect the entire analysis [42]. To facilitate the decision, the observations are separated into three classes:

- “short” class composed of cracks with a length lower than L_{LOW} (9 out of 31 observations, 29%),
- “medium” class composed of cracks with a length between L_{LOW} and L_{HIGH} (10 observations, 32%),
- “long” class composed of cracks with a length higher than or equal to L_{HIGH} (12 observations, 39%).

Then the dataset composition can be described as follows:

- 24 records (77%) correspond to gas turbines that operate in the marine environment (salty, onshore), while in the case of “short” and “medium” classes the fraction is equal to 95% (18 out of 19 observations);
- Records related to engines that operate in the tropical environment (damp, onshore) make up 58% of the “long” class (7 out of 12 observations);
- 19 records (61%) correspond to Type C units and make up the majority of the “medium” and “long” classes (90% and 58% respectively);
- 7 records (23%) correspond to Type A units and make up 33% of the “short” and “long” classes;
- 5 records (16%) correspond to Type B units and make up 33% of the “short” class and minority of the remaining classes.

Regression models estimating damage to the parts of the gas turbines should be accurate across the entire range of observations, to support decisions about the necessity to execute maintenance. Consequently, the main constraint on the test set composition is to properly reflect the split between the “short”, “medium”, and “long” classes. The test subset is composed as follows:

- Each class of crack lengths represents 1/3 of the set;
- To avoid excessive reduction of the training and validation subsets, the test set is composed of 6 data points (19% of the dataset) with 2 observations from each of the classes;

- 5 out of 6 records correspond to gas turbines that operate in marine (salty, onshore) environment;
- 4 out of 6 records correspond to Type C units, while the remainder correspond to Type A and Type B;
- The “short” class is represented by the longest observations assigned to this class;
- Records characterized by moderate values of the predictors are selected to the test subset, while the rest of the sample (with higher variance) is used for training and validation.

In the regression analysis a loss function is the prediction error for a single data point, while a cost function represents the error for all observations in the dataset. A squared error is commonly used as the loss function; thus, the MSE or root mean squared error (RMSE) are the most popular cost functions. The configuration of the cost function affects all subsequent steps of the analysis, i.e.: the tuning of the models' hyperparameters, the interpretation of results, and the selection of the optimal model. Also, it influences contributions that the deployed model brings when decisions are made. To obtain satisfactory results, the learning objective should be defined carefully with a deep understanding of the problem and user requirements.

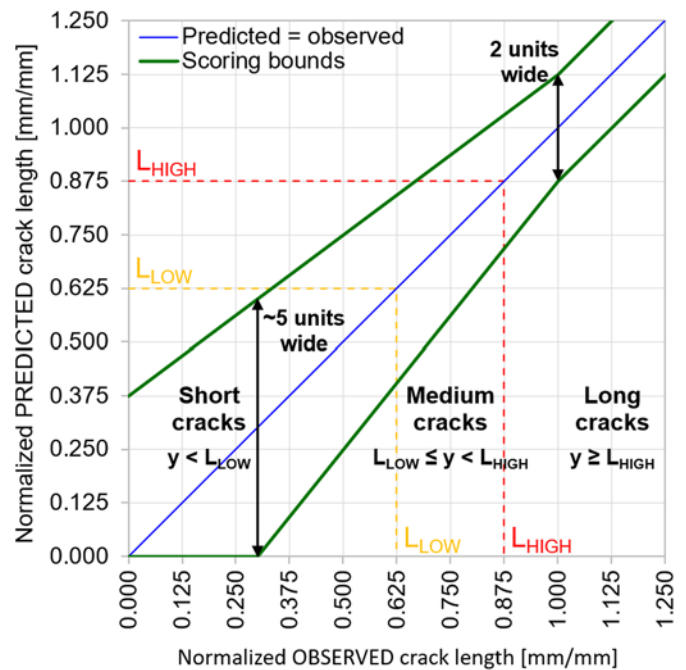


Fig. 5. Loss function scoring bounds on the "Predicted vs Observed" plot

For the analyzed Nozzle cracks, a certain value of absolute error is acceptable for the “short” class, which do not require immediate action. The same error is unacceptable for the “long” class, so cracks requiring careful evaluation due to the severity of potential consequences. The MSE does not meet this requirement, thereby custom loss and cost functions are used in this paper. The loss function has a variable width of the scoring bounds dependent on the length of observations (Fig. 5). The interval between the bottom bound $b(x)$ and the top bound $t(x)$ decreases as the size of actual cracks increases. The minimal width of the interval is equal to ~40% of the maximal width. The loss function is defined as follows:

$$l(x, \hat{f}(x)) = \begin{cases} 0, & x \notin [b(x), t(x)] \\ 1, & x \in [b(x), t(x)] \text{ and } x < L_{HIGH} \\ 2, & x \in [b(x), t(x)] \text{ and } x \geq L_{HIGH} \end{cases} \quad (3)$$

The loss function is positive only if the predicted crack length is within the bounds. The correct predictions of cracks belonging to the “long” class are scored 2x higher with respect to the remaining classes. This bonus favors regression models that accurately predict the longest cracks. The magnitude of the bonus should quantify the difference of importance of one aspect over others and should not artificially promote certain solutions. The authors noticed heavily biased predictions for short and medium observations when the bonus was too high.

The cost function is defined as the mean loss:

$$c(l(x, \hat{f}(x))) = \frac{1}{n} \sum_{i=1}^n l(x_i, \hat{f}(x_i)) \quad (4)$$

where $i = 1, 2, \dots, n$ is the ordinal number of the observation. The optimal regression model maximizes the cost function.

General constraints on the composition of the validation set are as follows:

- To correctly reflect the proportions between the “short”, “medium”, and “long” classes;
- To avoid excessive reduction of the training subset, the validation set has 1 observation from each of the classes; thus, 3 data points in total (~10% of the entire dataset);
- Cross-validation is repeated 560 times, for all the possible train-validation splits (the set is composed of 7 short cracks, 8 medium and 10 long cracks).

The codomain of the cost function is represented by the 5-element set: $\{0, \frac{1}{3}, \frac{2}{3}, \frac{3}{3}, \frac{4}{3}\}$. Such an extensive approach to CV is used to

avoid misinterpretation caused by a random choice of the validation set. Each CV repetition utilizes 22 observations to fit the model, and 3 observations to compute the value of the cost function. The model with the optimal values of hyperparameters maximizes the average value of the cost function from 560 repetitions. The fully specified model is trained on all 25 data points and evaluated against the test set.

The training, validation and test subsets were created in a fully controlled manner, based on the analysis of composition and clusters of the entire dataset. The authors observed that this approach gives better outcomes, simplifies the interpretation of obtained results, and improves the accuracy of the model. The loss and cost functions are custom, favor models that predict correctly in a specific range of the domain and reflect well the requirements of the user. Use of these functions significantly improved the accuracy of the prediction with respect to the MSE or the RMSE.

6. Description of Mathematical Model

Multiple linear regression (MLR) models assume that the response variable depends linearly on the independent variables. In scalar form it is represented as follows:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_k x_{ik} + \varepsilon \quad (5)$$

where k is the number of predictors, x_{ik} is the value of the k th predictor for the i th observation, β_0 is the intercept, β_k are the regression equation coefficients, and ε is the error term. The Nozzles are not damaged prior to the service, thereby $\beta_0 = 0$. In matrix notation, (5) simplifies to:

$$\vec{y} = \vec{\beta}X + \vec{\varepsilon} \quad (6)$$

where $\vec{\beta}$ is composed of coefficients of the regression equation (5) and X is the matrix of features of size $i \times k$.

Polynomial regression models assume that the response variable depends nonlinearly on the independent variables. In scalar form it is represented as follows:

$$\begin{aligned} y_i = & \beta_0 + \beta_{11}x_{i1} + \beta_{12}x_{i1}^2 + \beta_{13}x_{i1}^3 \dots + \beta_{1p}x_{i1}^p + \\ & + \beta_{21}x_{i2} + \beta_{22}x_{i2}^2 + \beta_{23}x_{i2}^3 \dots + \beta_{2p}x_{i2}^p + \\ & + \beta_{31}x_{i3} + \beta_{32}x_{i3}^2 + \beta_{33}x_{i3}^3 \dots + \beta_{3p}x_{i3}^p + \\ & + \dots + \\ & + \beta_{k1}x_{ik} + \beta_{k2}x_{ik}^2 + \beta_{k3}x_{ik}^3 \dots + \beta_{kp}x_{ik}^p + \varepsilon \end{aligned} \quad (7)$$

where p is the degree of the polynomial equation. If $\beta_0 = 0$, then (7) in matrix notation has the same form as (6), although the size of the matrix of features X depends on the degree p . X is complemented by all the possible interaction features of the degree j ($j = 2, \dots, p$) calculated as products of distinct independent variables. The degree of the polynomial equation $p \in \{2, 3, 4\}$ is the only hyperparameter tuned during the analysis.

The support vector regression (SVR) formulates (6) as an optimization problem aimed at finding the narrowest margin around the approximated surface [3]. The maximum error ε sets the width of the margin. The objective is to minimize the Euclidean norm of the coefficients' vector $\vec{\beta}$ that is normal to the approximated surface, subject to $|y_i - \vec{\beta}\vec{x}_i| \leq \varepsilon$:

$$\min \frac{1}{2} \|\vec{\beta}\|^2 \quad (8)$$

Only predictions outside the margin, called support vectors, are penalized. The solution of this constrained optimization problem is as follows:

$$\hat{f}(\vec{x}) = \vec{\beta}\vec{x} = \sum_{i=1}^n (\alpha_i - \alpha_i^*) \vec{x}_i \cdot \vec{x} \quad (9)$$

where α_i and α_i^* are Lagrange multipliers, and \cdot denotes the dot product in the space of input data \mathcal{X} . $\vec{\beta}$ is represented as a linear combination of the training vectors \vec{x}_i .

For non-linear relations, data are mapped into a higher dimension feature space \mathcal{F} using a similarity function called a kernel $k(\vec{x}_i, \vec{x})$. The explicit mapping $\Phi: \mathcal{X} \rightarrow \mathcal{F}$ is not required if the kernel satisfies:

$$k(\vec{x}_i, \vec{x}) := \Phi(\vec{x}_i) \cdot \Phi(\vec{x}) \quad (10)$$

In the space \mathcal{F} the solution of the optimization problem is as follows:

$$\begin{aligned} \hat{f}(\vec{x}) = \vec{\beta}\Phi(\vec{x}) &= \sum_{i=1}^n (\alpha_i - \alpha_i^*) \Phi(\vec{x}_i) \cdot \Phi(\vec{x}) = \\ &= \sum_{i=1}^n (\alpha_i - \alpha_i^*) k(\vec{x}_i, \vec{x}) \end{aligned} \quad (11)$$

The kernel function is also used with the ridge regression (KRR), which is a linear model with a regularization parameter $\lambda_2 \geq 0$. If the squared error is the cost function, then the minimized objective function has this form:

$$\min \left((\bar{y} - \mathbf{X}\bar{\beta})^T (\bar{y} - \mathbf{X}\bar{\beta}) + \lambda_2 \|\bar{\beta}\|^2 \right) \quad (12)$$

with the following solution:

$$\begin{aligned} \bar{\beta} &= \mathbf{X}^T (\mathbf{X}\mathbf{X}^T + \lambda_2 \mathbf{I}_n)^{-1} \bar{y} = \\ &= \mathbf{X}^T (\mathbf{K} + \lambda_2 \mathbf{I}_n)^{-1} \bar{y} = \mathbf{X}^T \boldsymbol{\alpha} = \sum_{i=1}^n \alpha_i \bar{x}_i \end{aligned} \quad (13)$$

where $\mathbf{K} = \mathbf{X}\mathbf{X}^T$ is the Gramian matrix and the kernel function, $\boldsymbol{\alpha} = (\mathbf{K} + \lambda_2 \mathbf{I}_n)^{-1} \bar{y}$ is the dual variable. The solution of the optimization problem is as follows:

$$\hat{f}(\bar{x}) = \bar{\beta}^T \bar{x} = \sum_{i=1}^n \alpha_i \bar{x}_i^T \cdot \bar{x} = \sum_{i=1}^n \alpha_i k(\bar{x}, \bar{x}_i) \quad (14)$$

A random forest is an ensemble of decision trees. A single tree divides the space of inputs \mathcal{X} into j high-dimensional rectangles R_j , in order to minimize the error at each tree split:

$$\min \sum_{j=1}^j \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2 \quad (15)$$

where \hat{y}_{R_j} is the mean value of the dependent variable in the R_j rectangle. A random subsample is drawn at each split; thus, the decision trees are decorrelated. Responses from all the trees comprising the random forest are averaged to obtain the final estimate:

$$\hat{f}(\bar{x}) = \frac{1}{m} \sum_{m=1}^m \widehat{f}_m(\bar{x}) \quad (16)$$

where m is the number of decision trees and $\widehat{f}_m(\bar{x})$ is the response of the m th decision tree.

In the AdaBoost.R2 algorithm [11] an ensemble of m weak learners, one-node decision trees, is created. Their training focuses on observations with the most inaccurate predictions obtained at the preceding iteration. The prioritization is based on weights assigned to each \bar{x}_i that depend on the confidence in the weak learner θ , being a function of the average loss of this weak learner. For an unseen vector of predictors \bar{x} , the response is calculated as the weighted median:

$$\hat{f}(\bar{x}) = \inf \left\{ y \in Y : \sum_{t: \widehat{f}_t(\bar{x}) \leq y} \log \left(\frac{1}{\theta} \right) \geq \frac{1}{2} \sum \log \left(\frac{1}{\theta} \right) \right\} \quad (17)$$

where the meanings of m and $\widehat{f}_m(\bar{x})$ are the same as in (16).

The extreme gradient boosted algorithm (XGBoost) [7, 45] is an ensemble of gradient boosted decision trees, whose responses are summed to get the final estimate:

$$\hat{f}(\bar{x}) = \sum_{m=1}^m \widehat{f}_m(\bar{x}) \quad (18)$$

The following function is minimized at each iteration t :

$$obj^t = \sum_{j=1}^T \left[G_j w_j + \frac{1}{2} (H_j + \lambda_2) w_j^2 \right] + \gamma T \quad (19)$$

where $j = 1, 2, \dots, T$ is the leaf's ordinal number, G_j is the gradient and H_j is the Hessian of the loss function, w_j is the similarity score assigned to the j th leaf, λ_2 is the regularization parameter and γ is the minimum loss reduction to split a node. Pruning of the decision trees is based on the gain value:

$$\begin{aligned} Gain &= \frac{1}{2} \left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma = \\ &= \frac{1}{2} [L + R + N] - \gamma \end{aligned} \quad (20)$$

where L and R are the scores on the new left/right leaf, and N is the score on the new node. The new branch is removed if the gain is negative.

Artificial neural networks are non-linear statistical models. The output from a neuron Z comprising a hidden layer of the network is a linear combination of inputs x_i :

$$Z_m = \sigma(\alpha_{0m} + \bar{\alpha}_m^T \bar{x}) \quad (21)$$

where $m = 1, 2, \dots, M$ denotes the neuron's ordinal number, σ is the activation function, α_{0m} is the bias term and $\bar{\alpha}_m$ is the vector with weights. The output of the entire network with a single hidden layer is as follows:

$$\hat{f}(\bar{x}) = g(\beta_0 + \bar{\beta}^T \mathbf{Z}) \quad (22)$$

where β_0 is the bias term, $\bar{\beta}$ is the vector with weights and $\mathbf{Z} = (Z_1, Z_2, \dots, Z_M)$ is composed of neuron outputs. Function g is the identity function in the case of regression problems. (Stochastic) gradient descent and back-propagation algorithms are used to calculate the weights.

7. Results of the Regression Analysis

Python programming language is utilized to create the regression models. NumPy [16] and Pandas [29] libraries are used for the data manipulation, Keras library [8] is used to build artificial neural networks, while Scikit-learn library is used to build the remaining models. The z-score is utilized to standardize the input data.

The Differential Evolution [39] method available in the SciPy library [40] is used to solve the optimization problem and to find the optimal vector $\bar{\beta}$ in the multiple linear and polynomial regression equations. The verification and validation of the models are presented in Fig. 6 and Fig. 7. The observed and the predicted values are normalized by dividing them by the maximal measured crack length. The dimensionless quantities generalize the results and make them more scientifically meaningful. It applies to all the figures in this section of the paper. The verification is done to assess the behavior of the model on the training set, while the validation evaluates the performance on the unseen, test dataset. The observations related to the obtained results are as follows:

- Regardless of the degree of the polynomial equation, the mean value of the cost function is equal to 1.200, therefore $p = 2$ is chosen to reduce the complexity of the model; No further efforts are taken by the authors to simplify the regression equation; However, an analysis of variance should be done to determine whether all the independent variables are statistically

significant, especially if some components of the $\bar{\beta}$ vector are close to zero.

- \tilde{T}_{EXH} and $\tilde{T}_{in_{P50-P70}}$ are the most important features in the MLR model, while their product and squared TFS are essential in the polynomial model (i.e. the modulus of the β_k coefficients corresponding to these features is higher than the modulus of the remaining coefficients of the regression equation).
- The polynomial model has a lower normalized RMSE than the MLR model (0.529 and 0.612, respectively), but this comes with the higher complexity and lower interpretability of the model.
- Responses of the models are not sufficiently sensitive to changes of the input parameters; hence, the length of short cracks is overpredicted and the longest cracks are underestimated.
- The accuracy is good only for the “medium” class.

In the case of support vector regression and kernel ridge regression, the values of the following hyperparameters are tuned:

- the maximum error $\varepsilon \in [0, 1)$,
- regularization parameters $C, \lambda_2 \in [0.001, 100)$,
- the kernel function,
- the degree of the polynomial kernel $p \in \{2, 3, 4, 5\}$,
- kernel coefficients $\gamma, r \in [0.001, 1]$,
- the tolerance for stopping criterion $t \in [0, 1)$.

To reduce the space of hyperparameters a fixed number of parameter values is sampled using the RandomizedSearchCV class. The entire space is iteratively limited to the subspaces in which the cost function is maximized. When the space is sufficiently small, the GridSearchCV class is used to evaluate all the remaining combinations of parameter values. The highest CV scores are achieved with the polynomial kernel:

$$k(\bar{x}_i, \bar{x}) = (\gamma \bar{x}_i \bar{x} + r)^p \quad (23)$$

The verification and validation of the models are presented in Fig. 8 and Fig. 9. The observations related to the obtained results are as follows:

- The mean values of the cost function (1.126 and 1.097 for the SVR and the KRR, respectively) are comparable with the mul-

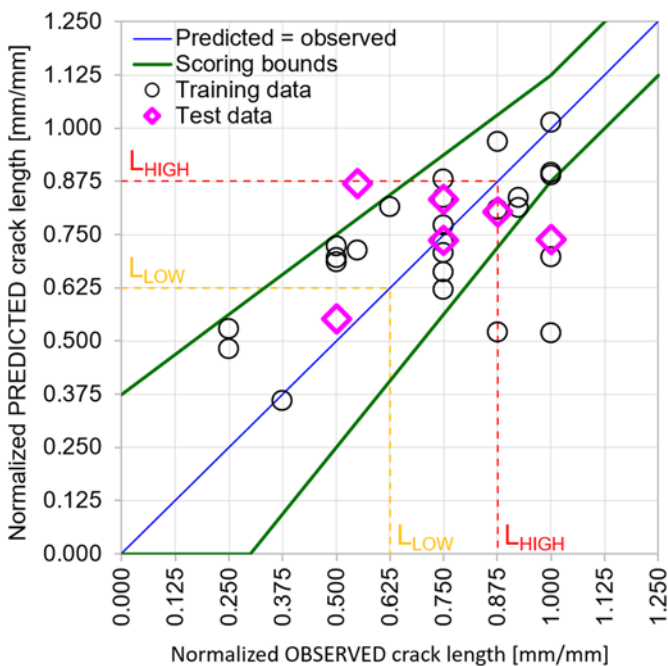


Fig. 6. Verification and validation of the multiple linear regression model

– tiple linear regression (1.133), so does the normalized RMSE (0.635 and 0.611, respectively).

- Responses of the models are sensitive to changes of the input parameters; hence, the predictions and the observations are in the same range.
- An exhaustive hyperparameter tuning can be done, since the mean time to fit these models is very short.
- The accuracy is unsatisfactory for the “long” class.

In the case of random forest regression, the AdaBoost.R2 and XGBoost algorithms, the following parameters are tuned:

- the maximum number of decision trees m ,
- the maximum depth of decision trees,
- whether to build the trees on the entire training set or on its subset, or whether to use bootstrap samples,
- the number of predictors used during each split,
- the minimum loss reduction γ , or the minimum decrease in impurity required to split a node,
- the minimum sum of the instance weight H_j required to split a node, or the minimum number of samples on the leaf after the split,
- the minimum number of observations required to split a node (not applicable to the XGBoost),
- the learning rate η (not applicable to random forest regression).

These hyperparameters are common for the considered tree-based algorithms. The size of the space of hyperparameters varies depending on the type of algorithm. Additionally, in the case of XGBoost regression, the following parameters are tuned:

- the booster type $\in \{\text{gbtree, dart}\}$,
- the L1 regularization parameter $\lambda_1 \in [0, 5]$,
- the L2 regularization parameter $\lambda_2 \in [0, 500]$,
- the fraction of decision trees neglected (dropped out) during each boosting step $d_r \in \{0.2, 0.4, 0.6, 0.8\}$,
- the probability of skipping the dropout procedure during a boosting step $d_s \in \{0, 0.2, 0.4, 0.6\}$.

The verification and validation of the models are presented in Fig. 10, Fig. 11, and Fig. 12. The observations related to the obtained results are as follows:

- \tilde{T}_{EXH} is the most important feature, regardless of the method utilized. The importance is calculated based on the values of

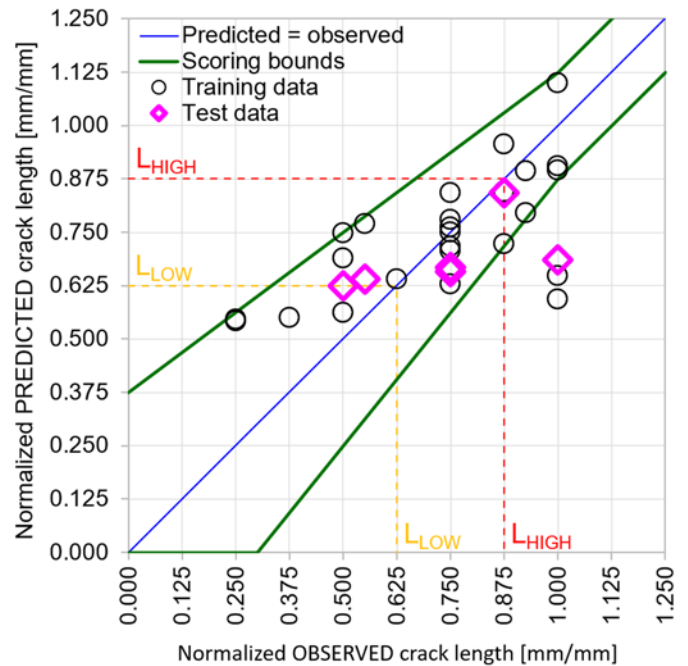


Fig. 7. Verification and validation of the polynomial regression model ($p = 2$)

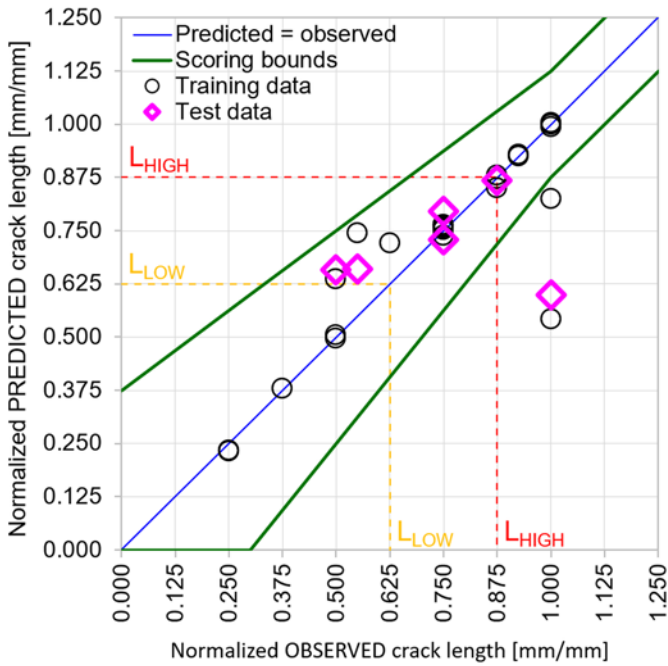


Fig. 8. Verification and validation of the support vector regression model

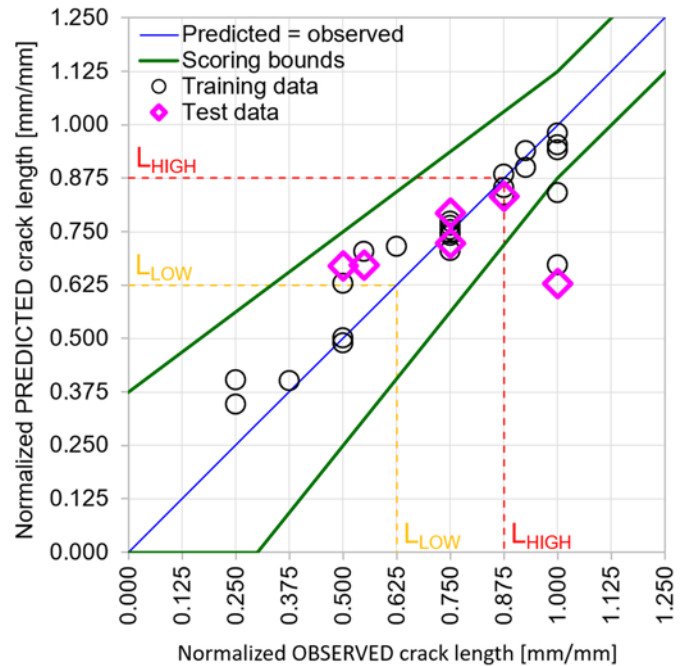


Fig. 9. Verification and validation of the kernel ridge regression model

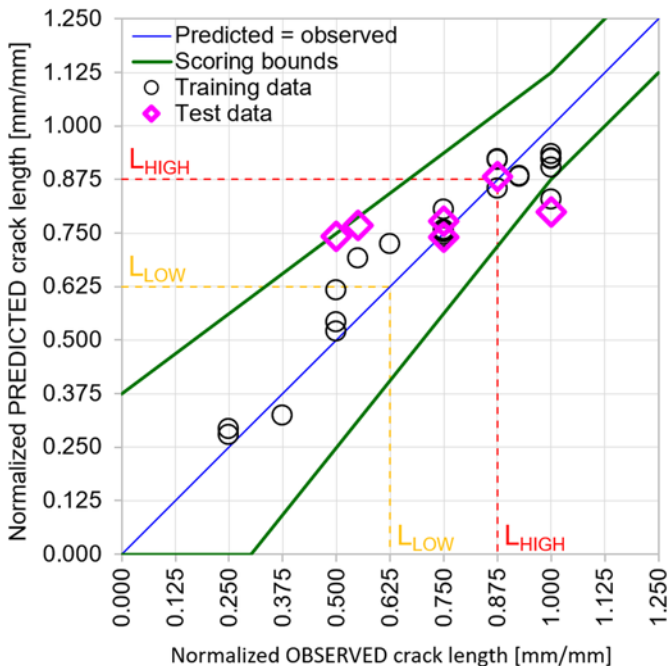


Fig. 10. Verification and validation of the random forest regression model

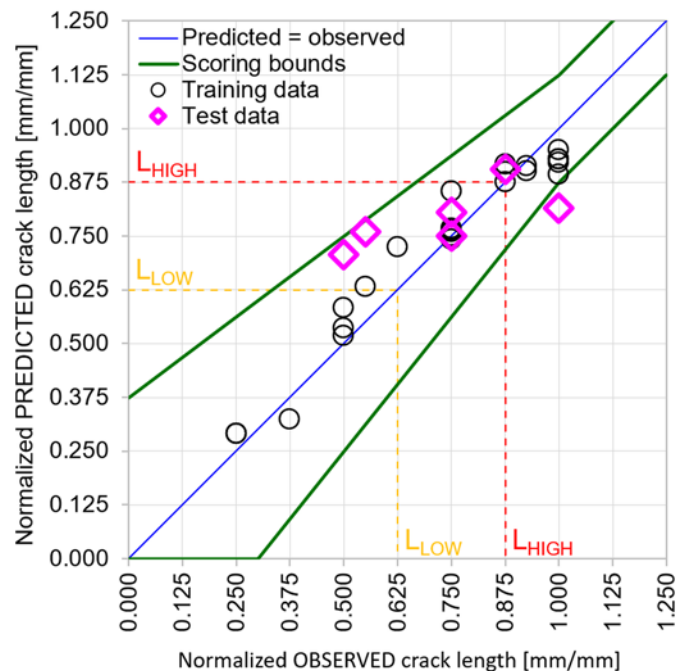


Fig. 11. Verification and validation of the AdaBoost regression model

gain for each decision tree in the XGBoost model or based on the mean decrease in impurity (Gini importance) in the remaining models.

- The models based on random forest and AdaBoost algorithms give low values of the normalized RMSE (0.544 and 0.500, respectively). However, based on the CV scores (0.962 and 0.980, respectively), these models are ranked lower than the remaining ones.
- The XGBoost regression model has a significantly higher CV score (1.096), but the normalized RMSE calculated against the test set is the highest (1.000). The tuning of hyperparameters aimed at reducing the error (i.e. to increase prediction accuracy for the longest crack) results in a substantial decrease of the cross-validation score. Despite some regularization parameters

having non-zero values, estimates for the training observations are almost perfect. A stronger regularization or reduction of the number of decision trees has a detrimental effect on the CV score and does not improve the normalized RMSE.

- The responses of the models are sensitive to changes in the input parameters; hence, the predictions and the observations are in the same range.
- Due to the higher complexity of these algorithms and the numerosity of user-defined constants, the tuning of hyperparameters requires more time in comparison with the previously discussed methods.
- The models underestimate the longest observations and none of them have a high cross-validation score or good results of the validation against the test set.

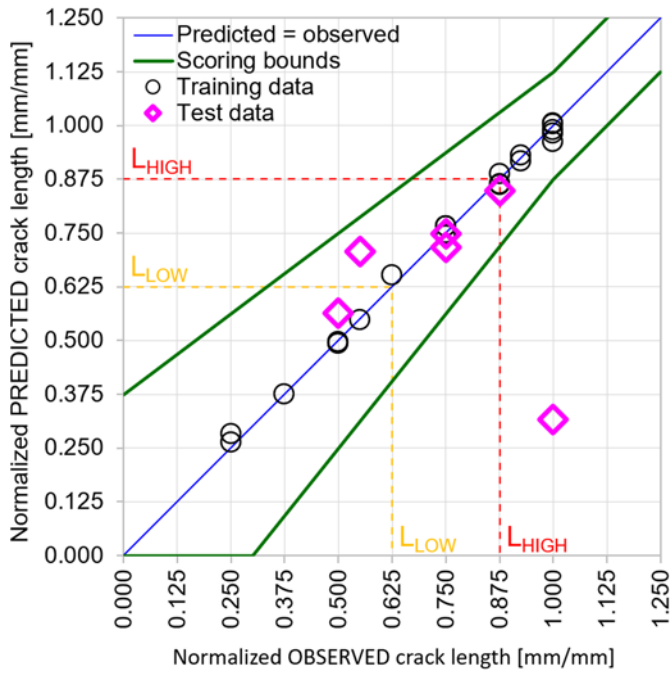


Fig. 12. Verification and validation of the XGBoost regression model

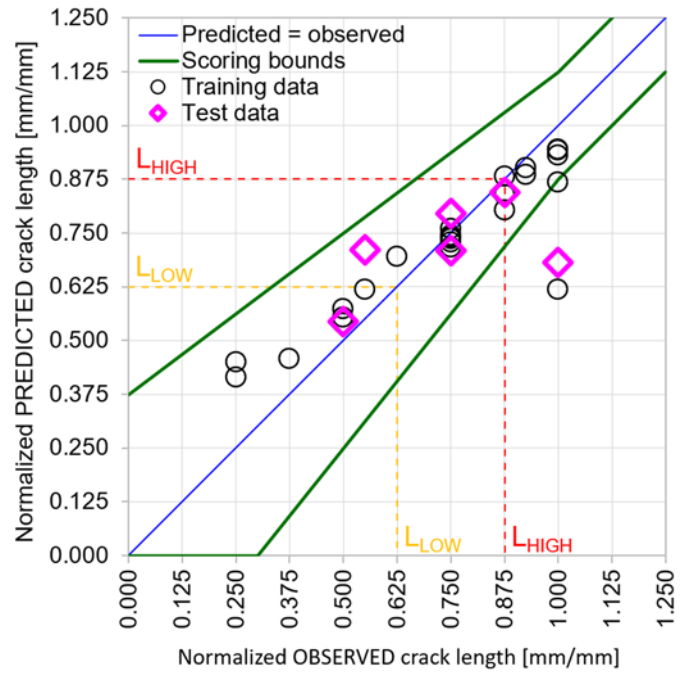


Fig. 13. Verification and validation of the neural network regression model

Table 1. A comparison of the regression models based on the mean CV score and the normalized RMSE evaluated against the test set

Algorithm type	Mean cross-validation score	Normalized RMSE
Polynomial regression	1.200	0.529
Multiple linear regression	1.133	0.612
Support vector regression	1.126	0.635
Kernel ridge regression	1.097	0.611
XGBoost regression	1.096	1.000
AdaBoost regression	0.980	0.500
Random forest regression	0.962	0.544
ANN regression	0.910	0.519

In the case of artificial neural network regression, the values of the following hyperparameters are tuned:

- the optimization algorithm ϵ {Adadelta, Adam, Adamax, Nadam},
- the activation function for hidden layers $\sigma \in$ {relu, exponential, hard sigmoid, sigmoid, softplus, tanh},
- the initializer used to set the initial weights of the ANN \in {GlorotNormal, GlorotUniform, he_normal, he_uniform, lecu_normal, lecu_uniform},
- the number of hidden layers \in {2, 3, 4, 5, 6, 8, 10},
- the number of neurons in the input layer and in the hidden layers $M_{inp}, M \in$ {5, 10, 15, 20},
- the number of training epochs \in [100, 1300],
- the number of training observations shown to the network before each update of the weights \in [4, 10],
- the fraction of neurons of the hidden layers that drop out during the training $d_r \in$ {0, 0.2, 0.25, 0.3, 0.4},
- the maximum norm of the vectors with weights $\|\vec{\alpha}_m\|, \|\vec{\beta}\| \in$ {3, 4, 5},
- the learning rate $\eta \in$ [0.0001, 0.1],
- the exponential decay rates for the 1st and 2nd moment estimates \in [0, 0.999] (applicable only to Adamax optimizer).

The highest cross-validation score (0.910) is obtained with the Adamax algorithm [25], the rectifier linear unit (relu) activation function defined as $\sigma(x) = \max(0, x)$, and the Lecun initializer that draws the initial weights from the uniform distribution $U(-\sqrt{3/M_{inp}}, \sqrt{3/M_{inp}})$. The artificial neural network has 3 hidden layers each with 15 neurons.

The verification and validation of the model are presented in Fig. 13. The observations related to the obtained results are as follows:

- the model gives low value of the normalized RMSE (0.519), although it has the worst mean value of the cost function;
- the model is a black-box and it is unknown which independent variable is the most important;
- responses of the models are sensitive to changes in the input parameters; hence, the predictions and observations are in the same range;
- regardless of the dataset size, the tuning of hyperparameters is very time-consuming, due to the quantity of user-defined constants and the length of time needed to fit a single artificial neural network;
- the model underestimates the longest observations.

The objectives of this study are to apply the statistical learning algorithms to a real technical problem, to share the approach and learnings with other researchers, and to evaluate if these algorithms are effective in the small-data regime. The results presented herein are sufficient to achieve the objectives. In the authors' opinion, the inclusion of additional results (e.g.: the scripts, the results before the normalization, or the final form and coefficients of the regression models) do not increase the scientific value of the paper. A quantitative summary of the obtained results is reported in Table 1.

8. Conclusions

The main outcomes of this study are as follows:

- The utilized algorithms can accurately predict the crack lengths based on operational parameters of the engine, i.e.: the temperature of exhaust gases, the number of fired starts accumulated by the part, the temperature of gases at the Nozzle's inlet, the gas turbine output and ambient air temperature. Based on the analysis of the linear/polynomial regression equations and the

values of gain/Gini importance, the median of the average exhaust temperatures \tilde{T}_{EXH} is the most important feature in the majority of the regression models.

- The polynomial regression model is the best model, considering the cross-validation score and the normalized RMSE evaluated against the test set. Nevertheless, it is not sufficiently sensitive to changes in the input parameters. It underestimates the longest observations as well, which is a common drawback with the created models. The AdaBoost regression model predicts these cracks with the lowest normalized RMSE. Before being used as a standalone support for data-driven decisions, the model should be subjected to further validation.
- The split into training, validation and test subsets was done in a fully controlled manner, considering the clusters and the composition of the entire dataset. Each subset represents the sample in a quantitative and qualitative way. Such a consistent approach reduces ambiguity during the cross-validation, testing and interpretation of obtained results, especially in the case of small datasets.
- The (root) mean squared error should not be automatically chosen as the cost function, as the results might be suboptimal from a business, risk management or other relevant perspective. A custom cost function better reflects the requirement of the user/customer and may favor certain solutions. In this paper, accurate predictions of the longest cracks are awarded with a bonus. The structure of the cost function drives the form and capabilities of the model.
- A variant of leave-p-out cross-validation is utilized, since the usage of k-fold cross-validation provides unclear and hard-to-interpret results for the training set composed of 25 records. This is computationally feasible because of the effective feature selection and the small number of observations.
- None of the models outperforms the remainder. Some of them are accurate for the “short” class, while others are better for the “long” class. Combining these models into an ensemble could improve the overall accuracy and better support data-driven decisions. None of the analyzed algorithms surpasses the others because of certain advantages or characteristics. The order of regression models in Table 1, resulting from the mean cross-

validation score, will change for a different dataset. Nevertheless, finding the optimal structure and hyperparameters of the artificial neural network was the most complicated and time-consuming. In general, the short time needed to fit a model allows for an extensive tuning of hyperparameters and is one of the few benefits of small samples.

- Building conclusions on a small dataset can be reliable, but it requires a rigorous approach and understanding of the decisions made throughout the entire analysis. The applicability domain must be defined for such models to limit extrapolation attempts and application of the model to items not covered by the training set. If possible, results of the data-driven model should be compared with a different approach, e.g., the results of the finite element analysis. A staggered implementation of the regression model is suggested, preferably connected with checks of the hardware (e.g., borescope inspection of the analyzed Nozzles), data gathering, and subsequent update of the model.

The authors applied several known machine learning algorithms to predict the maximal length of fatigue cracks. The analyzed dataset was not created synthetically or obtained during an experiment, but it describes components that operated in the industrial gas turbines under variable operating conditions. The purpose was to assess if the data-hungry methods can provide valuable results for the inhomogeneous sample composed of 31 observations only. The authors proved that the selected algorithms are effective in the small-data regime. The experience shared herein is universal and can support other researchers and professionals working with similarly sized sets.

It is the first attempt of the authors to apply machine learning algorithms in the small-data regime. Further research will focus on techniques that combine physics-based descriptions of the analyzed phenomenon with empirical models, staying in the small-data regime. Physics-informed neural networks [35, 9] are examples of such hybrid models. This approach should assure the consistency of the outputs with the laws of physics, and improve the accuracy of the predictions and extrapolation capabilities. The ultimate objective of the researchers is to propose a method of scaling the hybrid models and transferring knowledge between domains.

References

1. Abernethy RB. An Overview Of Weibull Analysis. The New Weibull Handbook: Reliability & Statistical Analysis for Predicting Life, Safety, Risk, Support Costs, Failures, and Forecasting Warranty Claims, Substantiation and Accelerated Testing, Usin Weibull, Log Normal, Crow-AMSAA, Probit, and Kaplan-Meier Models, 5th edition. North Palm Beach, Florida, Robert B. Abernethy: 2004: 13-24.
2. Allegorico C, Mantini V. A Data-Driven Approach for on-line Gas Turbine Combustion Monitoring using Classification Models. PHM Society European Conference 2014, <https://doi.org/10.36001/phme.2014.v2i1.1461>.
3. Awad M, Khanna R. Support Vector Regression. In Awad M, Khanna R (eds): Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers. Berkeley, CA, Apress: 2015: 67-80, https://doi.org/10.1007/978-1-4302-5990-9_4.
4. Beden S, Abdullah S, Ariffin AK. Review of Fatigue Crack Propagation Models for Metallic Components. European Journal of Scientific Research 2009.
5. Breiman L. Random Forests. Machine Learning 2001; 45(1): 5-32, <https://doi.org/10.1023/A:1010933404324>.
6. Carlevaro F, Cioncolini S, Sepe M et al. Use of Operating Parameters, Digital Replicas and Models for Condition Monitoring and Improved Equipment Health. Proceedings of the ASME Turbo Expo 2018: Turbomachinery Technical Conference and Exposition, Oslo, Norway, American Society of Mechanical Engineers Digital Collection: 2018, <https://doi.org/10.1115/GT2018-76849>.
7. Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, ACM: 2016: 785-794, <https://doi.org/10.1145/2939672.2939785>.
8. Keras. [<https://keras.io/>].
9. Dourado AD, Viana F. Physics-Informed Neural Networks for Bias Compensation in Corrosion-Fatigue. AIAA Scitech 2020 Forum, Orlando, FL, American Institute of Aeronautics and Astronautics: 2020. doi:10.2514/6.2020-1149, <https://doi.org/10.2514/6.2020-1149>.
10. Dresia K, Waxenegger-Wilfing G, Riccius J et al. Numerically Efficient Fatigue Life Prediction of Rocket Combustion Chambers using Artificial Neural Networks. 2019.
11. Drucker H. Improving Regressors Using Boosting Techniques. Proceedings of the 14th International Conference on Machine Learning 1997.
12. Durodola JF, Ramachandra S, Gerguri S, Fellows NA. Artificial neural network for random fatigue loading analysis including the effect of mean stress. International Journal of Fatigue 2018; 111: 321-332, <https://doi.org/10.1016/j.ijfatigue.2018.02.007>.
13. Escobedo E, Arguello L, Sepe M et al. Enhanced Early Warning Diagnostic Rules for Gas Turbines Leveraging on Bayesian Networks.

- Proceedings of the ASME Turbo Expo 2020: Turbomachinery Technical Conference and Exposition, American Society of Mechanical Engineers Digital Collection: 2021. doi:10.1115/GT2020-16082, <https://doi.org/10.1115/GT2020-16082>.
14. Faraway JJ, Augustin NH. When small data beats big data. *Statistics & Probability Letters* 2018; 136: 142-145, <https://doi.org/10.1016/j.spl.2018.02.031>.
 15. Gan L, Zhao X, Wu H, Zhong Z. Estimation of remaining fatigue life under two-step loading based on kernel-extreme learning machine. *International Journal of Fatigue* 2021; 148: 106190, <https://doi.org/10.1016/j.ijfatigue.2021.106190>.
 16. Harris CR, Millman KJ, van der Walt SJ et al. Array programming with NumPy. *Nature* 2020; 585(7825): 357-362, <https://doi.org/10.1038/s41586-020-2649-2>.
 17. Iannitelli M, Allegorico C, Garau F, Capanni M. A Hybrid Model for on-line Detection of Gas Turbine Lean Blowout Events. PHM Society European Conference 2018, <https://doi.org/10.36001/phme.2018.v4i1.405>.
 18. James G, Witten D, Hastie T, Tibshirani R. Linear Model Selection and Regularization. In James G, Witten D, Hastie T, Tibshirani R (eds): *An Introduction to Statistical Learning: with Applications in R*, New York, NY, Springer: 2013: 203-264, https://doi.org/10.1007/978-1-4614-7138-7_6.
 19. James G, Witten D, Hastie T, Tibshirani R. Statistical Learning. In James G, Witten D, Hastie T, Tibshirani R (eds): *An Introduction to Statistical Learning: with Applications in R*, New York, NY, Springer: 2013: 15-57, https://doi.org/10.1007/978-1-4614-7138-7_2.
 20. Jimenez-Martinez M, Alfaro-Ponce M. Fatigue damage effect approach by artificial neural network. *International Journal of Fatigue* 2019; 124: 42-47, <https://doi.org/10.1016/j.ijfatigue.2019.02.043>.
 21. Kalayci CB, Karagoz S, Karakas Ö. Soft computing methods for fatigue life estimation: A review of the current state and future trends. *Fatigue & Fracture of Engineering Materials & Structures* 2020; 43(12): 2763-2785, <https://doi.org/10.1111/ffe.13343>.
 22. Kalombo RB, Pestana MS, Freire Júnior RCS et al. Fatigue life estimation of an all aluminium alloy 1055 MCM conductor for different mean stresses using an artificial neural network. *International Journal of Fatigue* 2020; 140: 105814, <https://doi.org/10.1016/j.ijfatigue.2020.105814>.
 23. Kamath C, Fan Y J. Regression with small data sets: a case study using code surrogates in additive manufacturing. *Knowledge and Information Systems* 2018; 57(2): 475-493, <https://doi.org/10.1007/s10115-018-1174-1>.
 24. Karagiannopoulos M, Anyfantis D, Kotsiantis S B, Pintelas P E. Feature Selection for Regression Problems. *Proceedings of HERCMA 2007*, Athens, 2007.
 25. Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, San Diego, CA, USA, 2015.
 26. Kitchin R, Lauriault T. Small data in the era of big data. *GeoJournal* 2015; 80: 463-475, <https://doi.org/10.1007/s10708-014-9601-7>.
 27. Liu X, Athanasiou CE, Padture NP et al. A machine learning approach to fracture mechanics problems. *Acta Materialia* 2020; 190: 105-112, <https://doi.org/10.1016/j.actamat.2020.03.016>.
 28. Martens HA, Dardenne P. Validation and verification of regression in small data sets. *Chemometrics and Intelligent Laboratory Systems* 1998; 44(1): 99-121, [https://doi.org/10.1016/S0169-7439\(98\)00167-1](https://doi.org/10.1016/S0169-7439(98)00167-1).
 29. McKinney W. Data Structures for Statistical Computing in Python. *Proceedings of the 9th Python in Science Conference*, Austin, Texas, 2010; 445: 56-61, <https://doi.org/10.25080/Majora-92bf1922-00a>.
 30. Michelassi V, Allegorico C, Cioncolini S et al. Machine Learning in Gas Turbines. *Mechanical Engineering* 2018; 140(09): S54-S55, <https://doi.org/10.1115/1.2018-SEP5>.
 31. Nowell D, Nowell PW. A machine learning approach to the prediction of fretting fatigue life. *Tribology International* 2020; 141: 105913, <https://doi.org/10.1016/j.triboint.2019.105913>.
 32. Paris P, Erdogan F. A Critical Analysis of Crack Propagation Laws. *Journal of Basic Engineering* 1963; 85(4): 528-533, <https://doi.org/10.1115/1.3656900>.
 33. Pawelczyk M, Fulara S, Sepe M et al. Industrial gas turbine operating parameters monitoring and data-driven prediction. *Eksplotacja i Niezawodność - Maintenance and Reliability* 2020; 22: 391-399, <https://doi.org/10.17531/ein.2020.3.2>.
 34. Pedregosa F, Varoquaux G, Gramfort A et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 2011; 12: 2825-2860.
 35. Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics* 2018; 378: 686-707, <https://doi.org/10.1016/j.jcp.2018.10.045>.
 36. Rege K, Lemu H. A review of fatigue crack propagation modelling techniques using FEM and XFEM. *IOP Conference Series Materials Science and Engineering* 2017; 276: 012027, <https://doi.org/10.1088/1757-899X/276/1/012027>.
 37. Rovinelli A, Sangid MD, Proudhon H, Ludwig W. Using machine learning and a data-driven approach to identify the small fatigue crack driving force in polycrystalline materials. *npj Computational Materials* 2018; 4(1): 1-10, <https://doi.org/10.1038/s41524-018-0094-7>.
 38. Shanmugam M. Baker Hughes Company LLC: 2015.
 39. Storn R, Price K. Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization* 1997; 11: 341-359, <https://doi.org/10.1023/A:1008202821328>.
 40. Virtanen P, Gommers R, Oliphant TE et al. SciPy 1.0--Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* 2020; 17(3): 261-272, <https://doi.org/10.1038/s41592-019-0686-2>.
 41. Wang B, Xie L, Song J et al. Curved fatigue crack growth prediction under variable amplitude loading by artificial neural network. *International Journal of Fatigue* 2021; 142: 105886, <https://doi.org/10.1016/j.ijfatigue.2020.105886>.
 42. Xu Y, Goodacre R. On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning. *Journal of Analysis and Testing* 2018; 2(3): 249-262, <https://doi.org/10.1007/s41664-018-0068-2>.
 43. Zhan Z, Li H. Machine learning based fatigue life prediction with effects of additive manufacturing process parameters for printed SS 316L. *International Journal of Fatigue* 2021; 142: 105941, <https://doi.org/10.1016/j.ijfatigue.2020.105941>.
 44. Travel Weather Averages (Weatherbase). [<https://www.weatherbase.com/>].
 45. XGBoost Documentation - xgboost 1.4.0-SNAPSHOT documentation. [<https://xgboost.readthedocs.io/>].