Additional text added by AOC Existing items that need further discussion/clarification

Highlights

A Machine Learning Approach to Automate Ductile Damage Parameter Selection in Finite Element Simulations

A.N. O'Connor, P.G. Mongan, N.P. O'Dowd

- Bayesian optimisation derived parameters provide excellent agreement comparing simulated and experimental data.
- Ductile damage parameters are successfully derived for ambient and higher test temperatures.
- Required user-defined input such as test data can be easily generated from basic engineering tests.

A Machine Learning Approach to Automate Ductile Damage Parameter Selection in Finite Element Simulations

A.N. O'Connor[a,b], P.G. Mongan[a,c], N.P. O'Dowd[a,b,c]

[a] *School of Engineering, University of Limerick, Ireland*
[b] *Bernal Institute, University of Limerick, Ireland*
[c] *Confirm Smart Manufacturing Research Centre, Ireland*

Abstract

A key limitation of finite element analysis is accurate modelling of material damage. While additional material models exist that improve correlations between simulated damage and experimental data, these models often require additional parameters that are difficult to estimate. In this work we show that Bayesian optimisation, a machine learning technique, can be used to identify material model parameters. We show that Bayesian derived material model parameters result in simulated output with less than 2 % error compared to experimental data. The framework detailed here is fully autonomous, requiring only basic information that can be derived from a simple tensile test. We have successfully applied this framework to three datasets of P91 material tested at ambient (20 °C) and higher (500 °C) temperatures.

Keywords: machine learning; bayesian optimisation; ductile damage; parameter selection;

Email address: alison.oconnor@ul.ie (A.N. O'Connor)

1. Introduction

The tensile test is a standard test method [1] that provides information about the mechanical properties of metallic materials. In cases where experimental data are limited finite element (FE) simulations can be used to simulate the mechanical behaviour of materials and examine hypotheses that cannot be experimentally investigated. The FE method offers a level of detail not obtainable from analytical solutions but is relatively computationally expensive and ultimately relies on experimental data for validation. Ductile damage modelling can be used in conjunction with FE simulations to represent mechanical behaviour of metals under high strain conditions when damage mechanisms are important [2–4]. Such models are generally complex and require calibration parameters that are difficult to derive experimentally or analytically. Calibrating a ductile damage modelling can be considered a form of "black box optimisation", where the inputs (calibration parameters) and outputs (data in a tensile test) are known but the functional relationship between the calibration parameters and the material mechanical response is unknown. Such an optimisation problem can be solved using machine learning, a term used to describe algorithms and/or statistical models that allow computers to select the best method of progression in a problem, without human interaction or explicit programming [5]. In particular, Bayesian optimisation (BO) has been shown to outperform other machine learning algorithms in solving black-box optimisation problems [6]. Machine learning algorithms, informed directly by experimental and/or simulation data, have been successfully employed to solve numerous engineering problems across a wide range of industries [7–9]. BO is commonly used in machine learning

for artificial neural network (ANN) hyperparameter selection [5, 7, 10, 11]. In [2, 3, 12, 13] ANNs have been used to assess damage model parameters. However, ANNs require a significant number of FE simulations to 'train' the model to recognise correlations and relationships between input values and results which is computationally expensive. Ryan et al. [14] recently showed that BO can be adapted to identify material model constants based on limited FE simulation data making BO less computationally expensive compared to ANNs.

In this work we demonstrate how BO can be used to identify automatically the material damage model parameters that best match experimental data from a tensile test, reducing the time associated with deriving parameter values and requiring minimal user intervention.

2. Material behaviour in FE simulations

A tensile test consists of a standardised specimen geometry that is loaded in one direction until complete separation of the specimen. Data are typically expressed in terms of engineering stress and strain (σ_{eng} and ϵ_{eng}, respectively), where stress and strain are derived from test measurements of test machine load and specimen displacement, respectively. For metals the relationship between stress and strain typically comprises three regions (see Figure 1): a linear region demonstrating the elastic properties, a work hardening region where stress increases non-linearly with strain, and a necking region where stress decreases non-linearly with increasing strain. Several key mechanical properties derived from a tensile test are illustrated in Figure 1. The yield strength (σ_y) defines the stress at which material behaviour be-

comes non-linear. The linear region comprises data preceding the yield point of the material and the material stiffness or Young's modulus (E) is the slope of the stress-strain curve, as shown in Figure 1. Beyond σ_y the strain increases rapidly and a local geometrical discontinuity known as a 'neck' forms in the specimen. The ultimate tensile strength, σ_{UTS}, denotes the stress beyond which the specimen's load bearing capacity degrades. The strain hardening region occurs for $\sigma_y \leq \sigma \leq \sigma_{UTS}$, as illustrated in Figure 1, and depicts a non-linear stress-strain relationship. The conversion from load-displacement

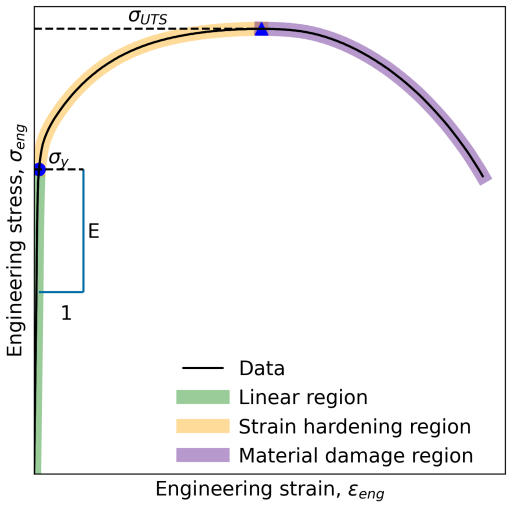


Figure 1: Illustration depicting material behaviour during tensile test experiment.

to engineering stress-strain (Equations 1 and 2) used in Figure 1, assumes that the specimen geometry does not change during the test. Where P represents loading, $Area$ the specimen cross-sectional area, l_0 the original length of the specimen and Δl the change in length. However, as the specimen in stretched in the loading direction, in reduces in thickness in the direction transverse to the load, due to the Poisson effect. Therefore, the cross-section

area of the specimen decreases continuously with increasing load. This effect becomes more noticeable post necking as the geometry changes are focused in the necked region and the cross-sectional area decreases significantly. To account for this geometric change a so-called true stress-true strain relationship (denoted as σ_{true} and ϵ_{true}, respectively) is formulated from engineering stress-strain data, as shown in Equations 1 to 4.

$$\sigma_{eng} = \frac{P}{Area} \tag{1}$$

$$\varepsilon_{eng} = \frac{\Delta l}{l_o} \tag{2}$$

$$\sigma_{true} = \sigma_{eng}\left(1 + \varepsilon_{eng}\right) \tag{3}$$

$$\varepsilon_{true} = ln\left(1 + \varepsilon_{eng}\right) \tag{4}$$

This equation becomes increasingly inaccurate as the neck develops due to the non-uniformity of stress and strain in the neck. Therefore, for $\sigma > \sigma_{UTS}$ the true stress is interpolated from the values for $\sigma \leq \sigma_{UTS}$, with the understanding that, in the absence of damage, the true stress always increases with increasing strain. Here, we assume a linear relationship between true stress and true strain, with slope m, for $\sigma \geq \sigma_{UTS}$. The parameter m is unknown and is calibrated as part of the BO, as discussed later.

2.1. The damage model

A popular damage model, used to describe the softening behaviour seen in the material damage region of Figure 1, is the Gurson-Tvergaard-Needleman

(GTN) model [3, 15, 16]. The parameters are typically estimated by conducting numerical simulations, manually iterating the relevant parameter value, and matching the simulation result to experimental data. The GTN model for a tensile test for a material with no strain hardening is described in Equation 5, where σ is the current (uniaxial) stress, σ_y is the material yield strength, f is the current void volume fraction, the ratio of the volume of voids to the total material volume, and q_1, q_2, and q_3 are fitting parameters.

$$\left(\frac{\sigma}{\sigma_y}\right)^2 + 2q_1 f\left(\cosh\frac{q_2\sigma}{2\sigma_y}\right) - q_3 f^2 = 1. \tag{5}$$

The void volume fraction, f, is described by Equation 6 where $\dot{f}_n$ represents and $\dot{f}_g$ represent the instantaneous void nucleation and growth, respectively. Void growth $\dot{f}_g$, Equation 8, is based on the law of conservation of mass, Equation 9, where $\dot{\varepsilon}$ represents the strain rate.

$$\dot{f} = \dot{f}_n + \dot{f}_g \tag{6}$$

$$\dot{f}_n = \frac{f_N}{S_N\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{\bar{\varepsilon}_m - \varepsilon_N}{S_N}\right)^2\right]\dot{\varepsilon} \tag{7}$$

$$\dot{f}_g = (1-f)\dot{\varepsilon} \tag{8}$$

$$\dot{E_{ij}} = \frac{1}{V}\int_{V_{matrix}} \dot{\varepsilon_{ij}} dV + \frac{1}{2V}\int_{S_{voids}} (v_i n_j + v_j n_i)\, dS \tag{9}$$

Don't we now need to define δS etc...plus we have $_i$, $_j$ nomenclature which is not previously referenced. Equation 7 for the void nucleation $\dot{f}_n$, is a

normally distributed function of the mechanical strain, ε_m with a mean ε_N and standard deviation S_N, with for large strain and no strain hardening,

$$\varepsilon_m = \frac{1}{1-f}\left(\frac{\sigma}{\sigma_y}\right)\varepsilon \tag{10}$$

Since, the void volume, f, is defined through a rate equation, the initial void volume fraction, f_0, must also be defined. Hence, there are a total of seven parameters in the GTN ductile damage model to fit using BO: q_1, q_2, q_3, S_N, f_N, ε_N, and f_0. In addition the slope of the true stress-true strain curve post UTS, m, is to be fitted, giving a total of eight parameters.

3. The Bayesian Optimisation Framework

The goal of BO is to find the global minimum of an unknown (black box) function. There are two key ingredients to a BO framework: a probabilistic model and a loss function. The probabilistic model, sometimes called a surrogate model, describes our belief about the function based on observed data. The loss function describes whether the previously observed data are optimised (i.e. converging towards a global minimum) [17].

To generate a surrogate model some observed data about the function must be provided to the BO framework. These data are then generalised using Gaussian process regression (GPR). The GPR is a collection of random variables that have consistent, joint Gaussian distributions that are completely specified by the mean and covariance function [18, 19]. The covariance function (also referred to as kernel, covarience kernel or kernel function), is user-defined and has significant impact on the GPR, and by extension the overall BO result [20]. For black-box functions where the shape

and characteristics are unknown it is common to combine multiple covariance functions to provide a wide-range of characteristics that might be relevant to our function.

The probability distribution generated by the GPR is based on a ranking system, where input x is evaluated as $f(x)$. The evaluation is then ranked in terms of its performance relative to the objective, this is often called the loss function. In this work we use mean average percentage error (MAPE) to quantify the similarity between simulated and experimental data (see Section 6 for further detail). To minimise our loss function we use the L-BFGS-B optimisation algorithm. The L-BFGS-B algorithm is popular for large scale optimisation problems as it requires less memory compared to other methods. The algorithm uses historical gradient evaluations to build up an approximation of the objective function.

Selecting a new evaluation point, x, is implemented through an 'acquisition' function, which uses historical data to identify an evaluation point that is likely to provide a reduced MAPE. In our implementation of the BO optimisation, the method repeats until either a suitable solution has been found (MAPE is below a specified value) or, some maximum number of iterations has been achieved.

The term 'parameter space' is used to define the minimum and maximum boundaries of the problem. This constitutes the space within which the BO algorithm will search for relevant values. The boundary of our problem is defined in Table 1 for the eight parameters. The minimum value of parameter m is assumed to be zero, which corresponds to no strain-hardening. The process used to define m is outlined in detail in Section 4. The minimum

and maximum parameter values for the GTN model were selected from the literature [21–23].

Table 1: Parameter space

Dataset	**Parameter**	**Minimum**	**Maximum**
1-3	q_1	0.90	1.60
1-3	q_2	0.90	1.10
1-3	q_3	0.81	2.56
1-3	ϵ_N	0.25	0.40
1-3	f_N	0.03	0.09
1-3	s_N	0.10	0.20
1-3	f_0	0.0013	0.0015
1	m	0.00	800
2	m	0.00	1100
3	m	0.00	800

3.1. The BO framework

BO requires a database of initial evaluated data. We use a design of experiment (DoE) approach, [24, 25], to evaluate a limited number of cases within our parameter space. The purpose of the DoE is to generate sufficient statistical data related to the relationship between our loss function (MAPE) and our eight unknown parameters in as few simulations as possible. Table 2 shows a DoE array for 12 initial simulations based on the maximum and minimum parameter space values. Where values of 1 and 2 in Table 2 represent the minimum and maximum parameter bounds, respectively.

Table 2: Design of experiments array for 8 parameters with each parameter at 2 levels.

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	2	2	2
3	1	1	2	2	2	1	1	1
4	1	2	1	2	2	1	2	2
5	1	2	2	1	2	2	1	2
6	1	2	2	2	1	2	2	1
7	2	1	2	2	1	1	2	2
8	2	1	2	1	2	2	2	1
9	2	1	1	2	2	2	1	2
10	2	2	2	1	1	1	1	2
11	2	2	1	2	1	2	1	1
12	2	2	1	1	2	1	2	1

Figure 2 shows a flowchart describing the DoE procedure. The user defines the parameter space. A DoE array, such as that shown in Table 2, is input. The DoE array is then modified so that the minimum and maximum values are replaced by the user-defined parameter space values. We then conduct a FE simulation for each row of the DoE array such that i_{max} is the number of rows in the DoE array. Results from the simulation are processed to extract the force and displacement of relevant elements. The physical experiment data are compared to the simulation data, and an error measurement is calculated (see Section 6 for further detail). After each iter-

ation parameter values and the associated error measurement are appended to a csv file.

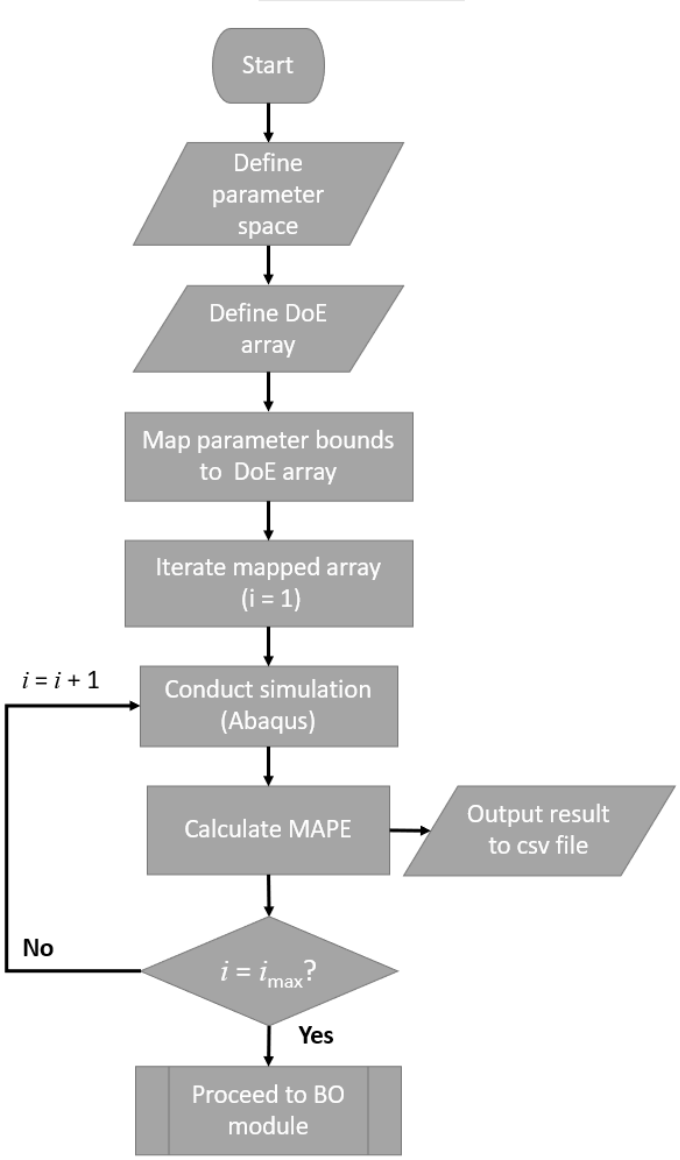


Figure 2: Flowchart demonstrating the design of experiments code procedure (prior to bayesian optimisation).

The output of the DoE (a csv file of parameter values and error measurements, Figure 2) is used to initialise the BO framework. Figure 3 outlines the BO framework. DoE data are analysed by the GPR to create an initial surrogate model. GPR outputs are provided to the acquisition function which identifies new parameter values based on ranking the DoE data. A new simulation, using acquisition function parameter values, is then conducted. Again simulation data is compared to the experiment and an error calculated. Acquisition function parameter values and error measurement data are appended to the existing csv file. As shown in Figure 3 the algorithm will iterate until either an appropriate solution is found (i.e. error of less than 2 % when comparing simulated and experimental results) or a maximum number of iterations (j_{max}) is achieved.

3.2. The covariance function

The covariance function used in Bayesian optimisation has significant influence on the shape and characteristics of the surrogate function [7]. The covariance function is a measure of how similar, or dissimilar, evaluation points are compared to their respective input positions. In this work we employ a covariance function created from a combination of the Matérn, constant and white noise functions as shown in Equation 11.

$$kernel = \text{Matérn} + \text{Constant} \times \text{White} \tag{11}$$

The Matérn covariance function is defined in Equation 12 [26], where $d(\cdot, \cdot)$ is the Euclidean distance, $K_\nu(\cdot)$ is a modified Bessel function and $\Gamma(\cdot)$ is the gamma function. This type of covariance function is a generalisation of

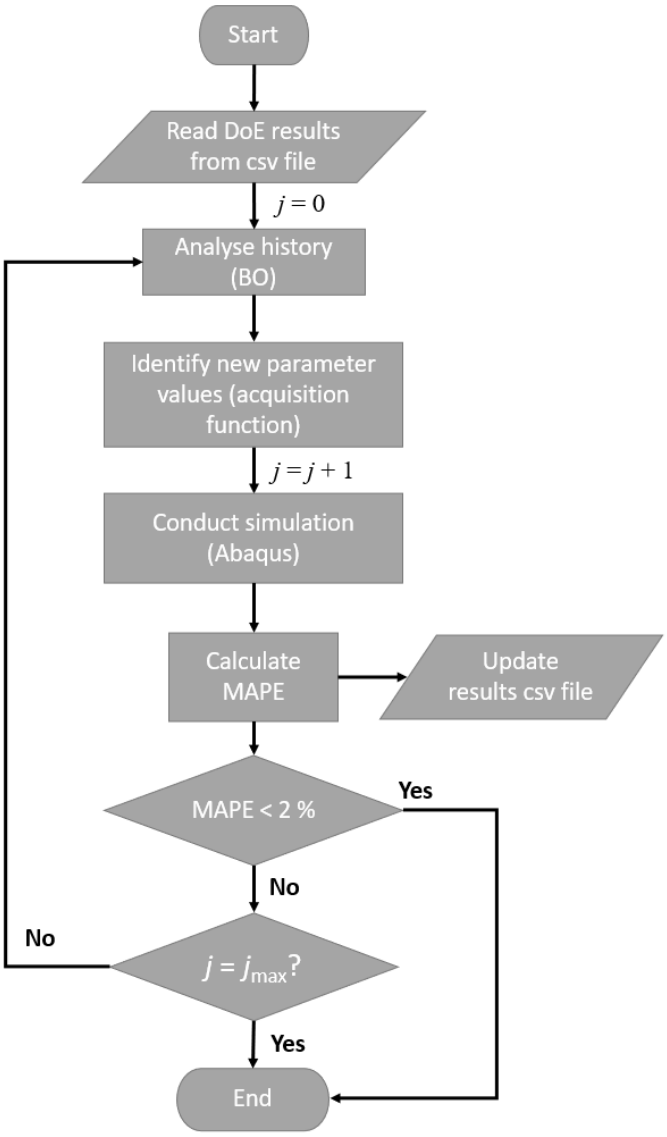


Figure 3: Flowchart depicting code procedure for Bayesian optimisation.

the radial basis function (RBF) with an additional parameter ν controlling the smoothness of the function. When $\nu \to \infty$ the covariance is equivalent to that of the RBF function, when $\nu \to 0.5$ the covariance is identical to the absolute exponential covariance.

$$k(x_i, x_j) = \frac{1}{\Gamma(\nu)\, 2^{\nu-1}} \left(\frac{\sqrt{2\nu}}{l} d(x_i, x_j) \right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}}{l} d(x_i, x_j) \right) \tag{12}$$

To modify the mean of the Gaussian process we use the Constant covariance function see Equation 13 [26].

$$k(x_1, x_2) = constant\ value\ \forall\ x_1, x_2 \tag{13}$$

The white covariance function allows users to more accurately model noisy data. It is a squared exponential function with a short lengthscale. The main use of the White covariance function is to explain noise in the signal as an independent entity that is normally-distributed [26]. The white covariance function is defined in Equation 14 where x and x' are input points, σ^2 is the variance of the noise and $\delta(x - x')$ is the Kronecker delta function which is equal to 1 when $x = x'$ and zero otherwise.

$$k(x, x') = \sigma^2 \times \delta(x - x') \tag{14}$$

As we do not know the shape or characteristics of the black-box function our covariance kernal (Equation 11) is designed to account for shifts in mean position, noise and model potentially multiple minima positions.

3.3. The acquisition function

There are a number of methods that can be applied to identify new parameter values. In this work we have used the upper confidence bound (UCB). The UCB is considered an optimistic strategy given that parameters are identified through a weighted sum of the surrogate function, $\mu(x)$, and the uncertainty, $\sigma(x)$, [17, 27] as shown in Equation 15.

$$\alpha_{UCB}(x) = \mu(x) + \sqrt{B_t}\,\sigma(x) \tag{15}$$

When the weight, $\sqrt{B_t}$, is large the model operates in exploratory mode, reducing the weight switches the model to exploitation mode. Exploration and exploitation balance how the model searches the parameter space. Typically this is a region of high variance where the model is uncertain. Exploitation refers to regions where the objective function is expected to be high ~~assumes that you have located a 'good' result and now wish to search smaller distances from that point to identify the 'best' result~~. One can also considered the exploration and exploitation modes in terms of the input value. Typically, in exploration mode the acquisition function selects a new assessment point that differs signficantly from that of the previous iteration. In exploitation mode, the assessment point will be more similar to the previous value. In BO one typically starts in exploration mode before moving to exploitation. The trade-off between exploration and exploitation is two-fold. Firstly it ensures that the model produces results in a reasonable timescale but secondly, and somewhat more importantly, the trade-off ensures the model doesn't fall into a local minima.

Here we are trying to minimise the difference between simulated and

experimental data where simulated data is a function of eight parameter values. The objective is to identify the global minimum position of our unknown function $f(x)$. Essentially the weight value applied in Equation 15 controls the magnitude of 'x' relative to your current position. In our case the next 'x' position is an array of 8 parameter values. Given the complexity of the GTN model in terms of parameter dependencies we reduced the BO framework complexity by fixing $B_t = 2.5$.

A hyperparameter is a parameter that controls the learning process of a model. The values of hyperparameters are set by the user and can have a significant impact on model performance [6, 11]. In addition to the weight, B_t, the acqusition function has two further hyperparameters: The number of random samples and the number of optimistaion samples.

The number of random samples does not directly affect the acquisition function. In essence the number of random samples is a value that controls the size of the array, over which the range of possible inputs are distributed. The main influence of this hyperparameter lies in discerning the mean of the objective function. Inreasing the number of random samples increases the precision of the GPR, thus reducing uncertainity.

The number of optimisation samples defines how many evaluations of the acquisition function are conducted. This hyperparameter controls the level of uncertainty in the acquisition function. Increasing the number of optimisation samples makes the optimisation process more robust.

Increasing the number of random samples or the number of optimisation samples comes at a computational cost. The overall objective of Bayesian optimisation is to minimise the number of expensive function evaluations.

Large increases in either hyperparameter may cause the optimised solution to approach the computational expense of the original function evaluation method.

4. Experimental Tensile Tests

The three experimental tensile test results are shown in Figure 4 for the material of interest (P91, a piping steel). Load-displacement data were con-

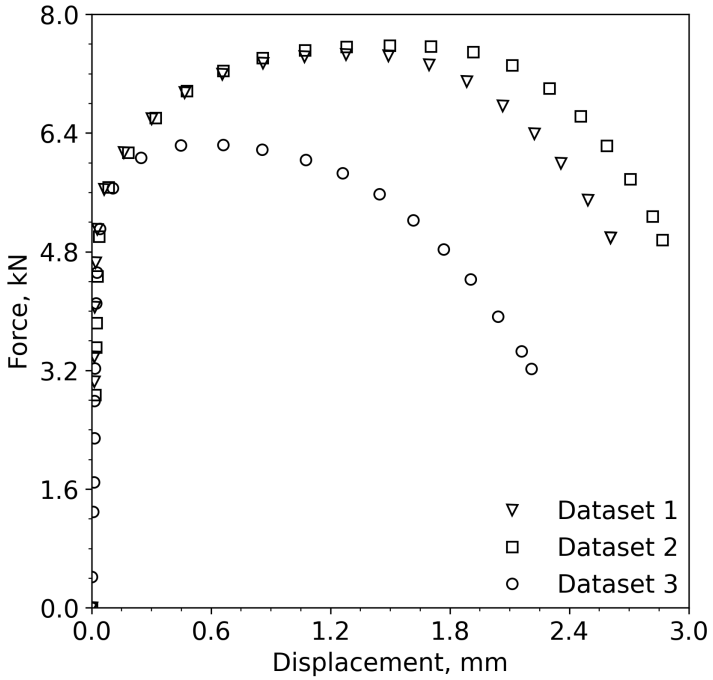


Figure 4: Experimental tensile test results for P91 material at test temperatures of 20 °C (Datasets 1 and 2) and 500 °C (Dataset 3). Note: Markers are placed as specific intervals for data visibility purposes. Higher data acquistion rate was used during testing.

verted to true stress-strain by limiting data to the range below the maximum load.

The ultimate tensile strength (σ_{UTS}) was defined as the true stress corresponding to the maximum load. The second derivative of true strain was calculated using a Savitzy-Golay filter with a three degree polynomial fit.

4.1. Assessment of linear data

To ensure the yield strength was accurately defined the size of the filter window (i.e. the number of data points over which the polynomial is applied) was solved iteratively. For each iteration (i.e. each window size) the yield strength (σ_y) was defined as the position of the inflection point. The 'best' yield strength was selected as a function of the error (between the predicted linear region and the experimental linear region), the overall fit of the linear region (r^2) and the estimated yield strength value. Young's modulus was calculated by employing linear regression to fit all data below the proposed 'best' yield strength. Figure 5 shows the result of the assessment for each of the three datasets.

To extrapolate data beyond σ_{UTS} a linear relationship between stress and strain was assumed. The slope of the extrapolated line, parameter m, is a searchable parameter that the BO framework will derive. The maximum value of parameter m was ascertained from experimental data by applying a linear fit to groups of data preceeding the σ_{UTS} position. The first fit was applied to a single pair of data points immediately adjacent to the σ_{UTS} position, the second fit was applied to the previous pair plus an additional data point, the third fit was applied by expending the second group by an additional data point and so on. This method was applied up to a maximum of five data points preceding the σ_{UTS} position. Parameter m was assessed for each linear fit and the largest value was selected to represent the maximum

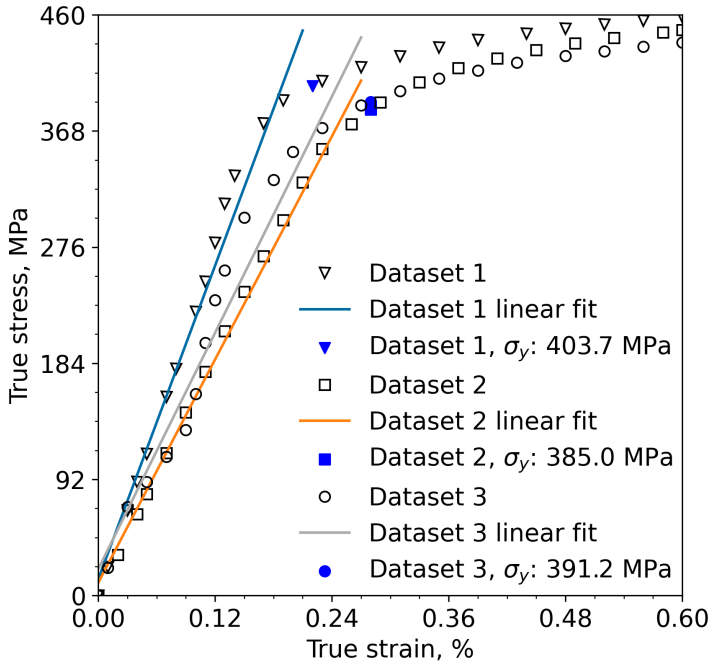


Figure 5: True stress - true strain relationship for three experimental tests.

parameter boundary. True stress-true strain data were extrapolated based on the assumption that the extrapolated line must pass through the σ_{UTS} position with a slope of parameter m. Figure 6 shows the extrapolated data for the parameter space of dataset 1.

5. FE Modelling of Tensile Tests

Finite element modelling was conducted using Abaqus 2021 [23]. Analyses were conducted using Abaqus/Explicit, a dynamics based solution, with a mass scaling factor of 10 to reduce computation time. The model was meshed using axisymmetric elements. A total of 4664 elements were used. Each element in the gauge section represents approximately 0.09 mm.

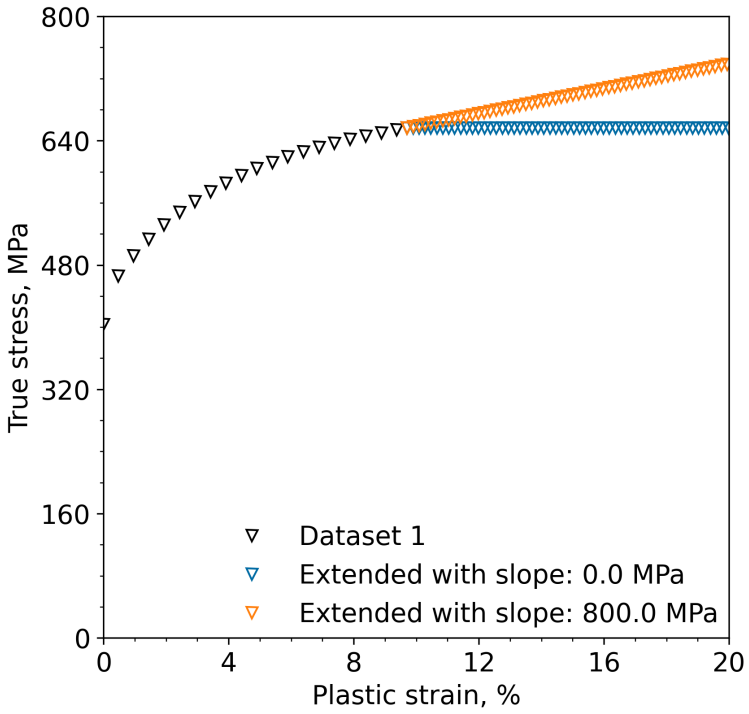


Figure 6: True stress - plastic strain relationship for dataset 1. Test data are shown using black markers. Two parameter values (minimum and maximum) are distingushed using coloured markers.

The experimental tensile test geometry was represented as shown in Figure 7. The use of a 2D geometry reduces the computational expensive associated with the simulation. To predispose the FE model to neck in the central gauge length a small imperfection was introduced to the bottom right hand corner as shown in Figure 7.

Boundary conditions were applied along the symmetry planes (shown by the red dashed lines in Figure 7). A reference node (red point shown in Figure 7) was coupled to the surface of the geometry and loading was applied through displacement control.

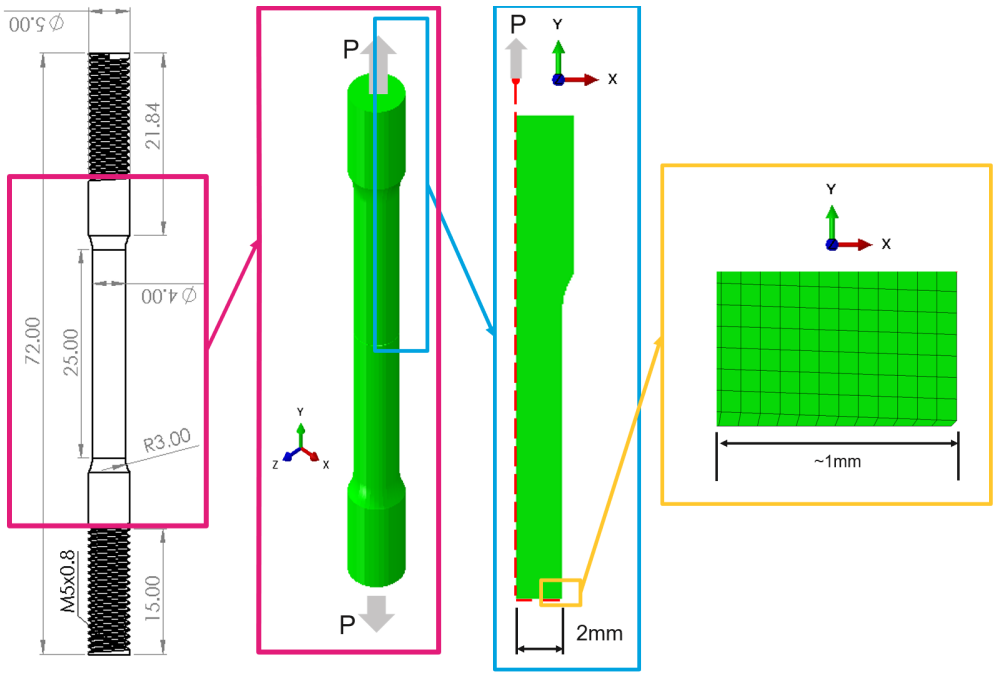


Figure 7: Illustration demonstrating the axisymmetric FE model. A small imperfection, shown in the yellow highlighted region, was modelled to ensure failure occurred in the midsection of the gauge length.

Within the FE simulation linear material behaviour was defined using the E and Poisson's Ratio ($\nu = 0.3$). Non-linear material behaviour was provided as an array of plastic strain versus true-stress (*PLASTIC in Abaqus). To model material degradation due to ductile damage the GTN model was applied (see Section 2.1).

Displacement data was taken from the reference node shown in Figure 7. The applied load was calculated by summing the vertical reaction force in elements located along the X symmetry plane.

6. Comparing experimental and simulated data

Mean average percentage error (MAPE), Equation 16, was used to measure similarity between the simulated output and experimental data. Here P_{exp} represents experimental load, P_{sim} simulation load and n the number of data points. Firstly we find the minimum change in displacement (Δu_{min}) in either the experimental or simulated result. We then interpolate both arrays assuming a cubic relationship with an increment of Δu_{min}. This ensures that both simulated and experimental data have identical displacement arrays. MAPE is calculated based on how similar the applied loading is for each increment of displacement as shown in Equation 16, where P_{exp} is experimental load, P_{sim} simulation load and n the number of data points.

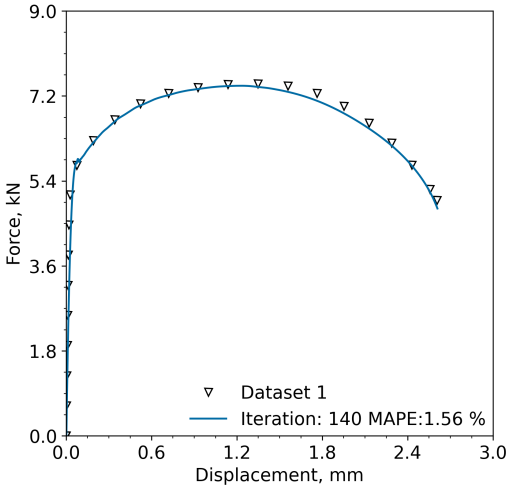
$$\text{MAPE} = \frac{100}{n} \sum^{n} \left| \left[\frac{P_{exp} - P_{sim}}{P_{exp}} \right] \right| \tag{16}$$

MAPE provides an average measure of error over the data. In this work we have assumed that data points are equally weighted.

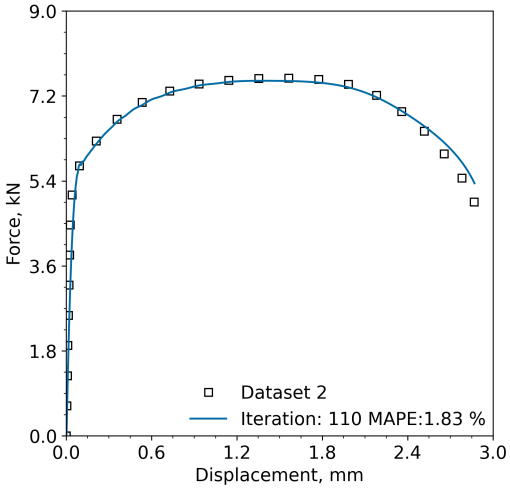
7. Results and Discussion

The parameter values selected by the BO framework are provided in Table 3 for each of the three datasets analysed. Simulated data, based on the parameter values shown in Table 3, are compared to the relevant experimental data in Figure 8 and, as shown, excellent agreement to experimental data was found for all datasets. In all cases the MAPE was found to be less than 2 %. The lowest MAPE, 1.56 %, was found for dataset 1.

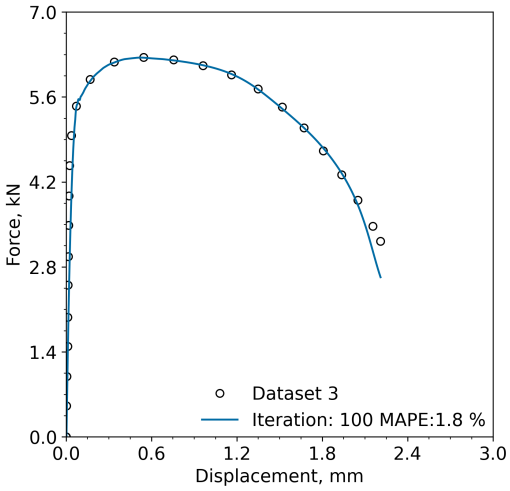
Figure 8a shows simulated output in very good agreement with experimental data for comparable displacement values. In the range $1.3 \leq \Delta u \leq$



(a) Dataset 1



(b) Dataset 2



(c) Dataset 3

Figure 8: Comparison of experimental test data to simulated output for Dataset 1 (8a), Dataset 2 (8b), and Dataset 3 (8c). Parameter values are shown in Table 3.

Table 3: Bayesian optimisation framework parameter values for each of the three datasets analysed.

Parameter	**Dataset 1**	**Dataset 2**	**Dataset 3**
Number of optimisation samples		500	
Number of random samples		50,000	
q_1	1.33244	1.16841	1.18197
q_2	0.99523	0.96665	0.96988
q_3	2.29488	1.40713	2.00065
ϵ_N	0.29000	0.34674	0.37195
s_N	0.16706	0.18562	0.12137
f_N	0.03869	0.08481	0.04706
f_0	0.00137	0.00134	0.00131
m	534.13	725.39	421.41
MAPE	1.56	1.83	1.80
Iteration number	140	110	100

2.2 mm simulated forces are slightly underestimated compared with the experiment. Similarly Dataset 2 (Figure 8b) shows excellent agreement between simulated and experimental data. For $\Delta u \geq 2.5$ mm simulated results for Dataset 2 slightly overestimate the force compared to the experiment. The overestimation is most notable at the final displacement position. It is interesting to note that parameter m, the slope of the extrapolated true stress-true strain curve was found to differ significantly. The optimised parameter m for Datasets 1 was found to be 534.13 MPa while that for Dataset 2 was found

to be 725.39 MPa. Both Datasets were tested under identical conditions and, as shown in Figure 4, are broadly similar up to a displacement of ≈ 1.3 mm after which they begin to diverge.

It is interesting to note that for all three datasets the parameter f_0, representing the initial void volume fracture of the material, are similar but not identical. Generally the initial void volume fraction is assumed to be applicable to all test samples taken from a single piece of material. This is, at least in part, due to the difficulty in assigning a parameter value. The statistical nature of materials testing, often termed material scatter, has been widely acknowledged in the field of material science [28]. Given the difficultly in experimentally deriving appropriate f_0 values and considering the statistical nature of material performance, it may be that machine learning methods like that employed here are more appropriate for defining this parameter. The lack of identical parameter f_0 values seen here are considered a reflection of the expected scatter behaviour seen in the wider field.

Comparing parameters ε_N, S_N and f_N across the three datasets presented in Table 3 show relatively significant changes when comparing any given parameter. ε_N and S_N represent the mean and standard deviation, respectively, associated with a normally distributed curve. These parameters essentially control how many voids nucleate under a given loading condition. As such parameters ε_N and S_N are considered material specific rather than specimen specific. Given the complex nature of the material model and the close relationship between these parameters it is difficult to state with any certainty what impact any given parameter has had on simulated output.

7.1. Hyperparameter effects

As previously discussed, Section 3.3, the acquisition function selects new parameter values based on a cheap function evaluation. To assess the effect of hyperparameter selection on each of the eight model parameters we preformed multiple analyses on dataset 1. For each analyses the number of random and/or optimisation samples were reduced. Results are plotted in Figure 9 and tabulated in Table 4.

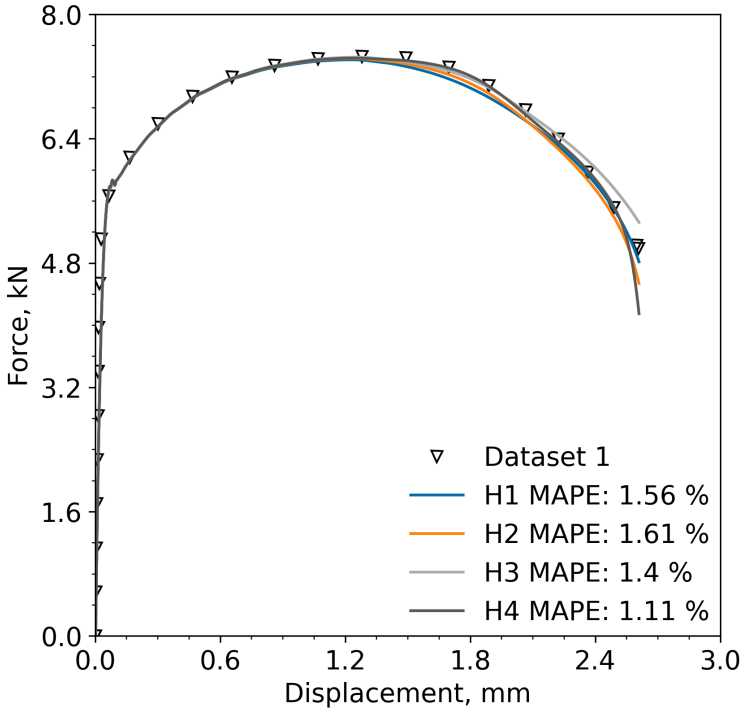


Figure 9: Plot of force versus displacement showing experimental data and simulated output for various hyperparameters.

Figure 9 shows excellent agreement between simulated output and dataset 1. Figure 9 shows that, regardless of hyperparameter selection, excellent agreement to experimental data was found. For each analysis a MAPE < 2 % was

Table 4: Bayesian optimisation framework parameter values for dataset 1. The effect of hyperparameter settings on GTN parameter values.

Parameter	**H1**	**H2**	**H3**	**H4**
Number of optimisation samples	500	100	500	100
Number of random samples	50,000	50,000	10,000	10,000
q_1	1.33244	1.17749	1.29095	1.15780
q_2	0.99523	1.02152	0.95466	1.03653
q_3	2.29488	1.34431	2.25353	1.07016
ϵ_N	0.29000	0.35978	0.35185	0.33307
s_N	0.16706	0.15799	0.10008	0.12343
f_N	0.03869	0.05769	0.03086	0.07594
f_0	0.00137	0.00143	0.00141	0.00131
m	534.13	542.79	514.54	621.85
MAPE	1.56	1.61	1.40	1.11
Iteration number	140	133	46	130

achieved, as expected. As seen in Figure 9 each analysis, though similar, is not identical. Comparing the eight model parameters in Table 4 shows that model parameters change as a function of hyperparameter selection. Given that excellent agreement was achieved for all four analyses this implies that the problem does not have a unique solution. That is to say that multiple combinations of the eight model parameters potentially provide good agreement to experimental data.

The lowest MAPE, 1.11 % was found for analysis H4 where the number of

random and optimisation samples were 10,000 and 100, respectively. Comparing the model parameters (q_1, q_2 etc.) for the four analyses shows that H4 deivates considerably from H1. H1, as it has the highest number of random and optimisation samples, is considered the most precise of the analyses investigated here. Comparing parameter m in analyses H1 (534.13 MPa) and H4 (621.85 MPa) shows a difference of almost 100 MPa. In terms of the expected change, and considering the parameter range, the deviation across analyses is significant. It is therefore not unexpected that the remaining parameter values, those specific to the GTN model, also differ. This supports the non-uniquess of the problem.

Comparing H1 and H2 in Table 4 indicates how reducing the number of optimisation samples affects the analysis. While specific model parameters change slightly overall the analyses are similar in terms of parameter values and MAPE. Though it should be noted that the MAPE found in H2 was slighly higher than that of H1, as expected. Comparing H1 and H3 we can assess the effect of reducing the number of random samples. Here we see that fitting parameters q_1, q_2 and q_3 are broadly similar. Other parameters, specifically nucleation related terms s_N and f_N, exhibit differences with H3 having smaller values compared to H1. The effect of the reduction in these parameters is also notable in Figure 9 where comparing H1 and H3 shows that H3 diviates from H1 at $2.0 \leq \Delta \leq 2.7$ mm. In essence the lower nucleation parameters seen in analysis H3 reduced the nucleation of new voids thus giving a lower rate of damage compared to H1. On the whole comparing H1 and H3 indicates that reducing the number of random samples did not significantly affect the model output.

8. Conclusions

- Bayesian optimisation framework successfully derived an array of 8 parameter values that, when applied to a ductile damage model simulation, produced an accurate representation of experimental data with a mean average percentage error of less than 2 %.
- The framework is fully autonomous requiring only basic information that can be easily generated from experimental tensile test data.
- Model hyperparameters are shown to effect model parameter values.

Data files and code used in this paper can be downloaded from https://github.com/alisonoc/PAPERA.

9. Ackowledgements

This work was funded by the European Union through the Marie Skłowodska-Curie Actions grant number 101028291. William Brennan, a University of Limerick undergraduate, is thanked for his contributions to earlier versions of the algorithm used in this work. We gratefully acknowledge helpful conversations with Dr. Meghana Kshiragar and Gauri Vaidya from the University of Limerick's Lero Centre. We also thank the Mathematics Applications Consortium for Science and Indusry (MACSI) group for their insightful and supportive comments surrounding this research.

References

[1] EN ISO 6892-1:2019 Metallic materials - Tensile testing - Part 1: Method of test at room temperature (ISO 6892-1:2019), 2019.

[2] F. Abbassi, T. Belhadj, S. Mistou, A. Zghal, Parameter identification of a mechanical ductile damage using Artificial Neural Networks in sheet metal forming, Materials & Design 45 (2013) 605–615. doi:http://dx.doi.org/10.1016/j.matdes.2012.09.032.

[3] Y. Chahboub, S. Szavai, Determination of GTN parameters for SENT specimen during ductile fracture, in: Procedia Structural Integrity, volume 16, Elsevier B.V., 2019, pp. 81–88. doi:10.1016/j.prostr.2019.07.025.

[4] T. Zhang, K. Lu, A. Mano, Y. Yamaguchi, J. Katsuyama, Y. Li, A novel method to uniquely determine the parameters in Gurson–Tvergaard–Needleman model, Fatigue & Fracture of Engineering Materials & Structures 44 (2021) 3399–3415. doi:10.1111/ffe.13568.

[5] T. Bikmukhametov, J. Jäschke, Combining machine learning and process engineering physics towards enhanced accuracy and explainability of data-driven models, Computers & Chemical Engineering 138 (2020) 106834. doi:10.1016/j.compchemeng.2020.106834.

[6] J. Snoek, H. Larochelle, R. P. Adams, Practical Bayesian Optimization of Machine Learning Algorithms, in: Advances in Neural Information Processing Systems, volume 25, Curran Associates, Inc., 2012.

[7] P. G. Mongan, V. Modi, J. W. McLaughlin, E. P. Hinchy, R. M. O'Higgins, N. P. O'Dowd, C. T. McCarthy, Multi-objective optimisation of ultrasonically welded dissimilar joints through machine

learning, Journal of Intelligent Manufacturing 33 (2022) 1125–1138. doi:10.1007/s10845-022-01911-6.

[8] X. Liu, C. E. Athanasiou, N. P. Padture, B. W. Sheldon, H. Gao, A machine learning approach to fracture mechanics problems, Acta Materialia 190 (2020) 105–112. doi:10.1016/j.actamat.2020.03.016.

[9] J. Hegde, B. Rokseth, Applications of machine learning methods for engineering risk assessment – A review, Safety Science 122 (2020) 104492. doi:10.1016/j.ssci.2019.09.015.

[10] I. Dewancker, M. McCourt, S. Clark, Bayesian Optimization for Machine Learning : A Practical Guidebook, arXiv:1612.04858 [cs] (2016). arXiv:1612.04858.

[11] F. Ghavamian, Accelerating finite element analysis using machine learning (2021). doi:10.4233/uuid:015bbf35-5e29-4630-b466-1a29d4c5bfb3.

[12] M. Abendroth, M. Kuna, Identification of ductile damage and fracture parameters from the small punch test using neural networks, Engineering Fracture Mechanics 73 (2006) 710–725. doi:10.1016/j.engfracmech.2005.10.007.

[13] D. Chen, Y. Li, X. Yang, W. Jiang, L. Guan, Efficient parameters identification of a modified GTN model of ductile fracture using machine learning, Engineering Fracture Mechanics 245 (2021) 107535. doi:10.1016/j.engfracmech.2021.107535.

[14] S. Ryan, J. Berk, S. Rana, B. McDonald, S. Venkatesh, A bayesian optimisation methodology for the inverse derivation of viscoplasticity model constants in high strain-rate simulations, Defence Technology (2021). doi:10.1016/j.dt.2021.10.013.

[15] W. Wcislik, Experimental determination of critical void volume fraction fF for the Gurson Tvergaard Needleman (GTN) model, Procedia Structural Integrity 2 (2016) 1676–1683. doi:10.1016/j.prostr.2016.06.212.

[16] V. Tvergaard, A. Needleman, Analysis of the cup-cone fracture in a round tensile bar, Acta Metallurgica 32 (1984) 157–169. doi:10.1016/0001-6160(84)90213-X.

[17] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, N. de Freitas, Taking the Human Out of the Loop: A Review of Bayesian Optimization, Proceedings of the IEEE 104 (2016) 148–175. doi:10.1109/JPROC.2015.2494218.

[18] C. E. Rasmussen, Gaussian Processes in Machine Learning, in: O. Bousquet, U. von Luxburg, G. Rätsch (Eds.), Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003, Tübingen, Germany, August 4 - 16, 2003, Revised Lectures, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2004, pp. 63–71. doi:10.1007/978-3-540-28650-9_4.

[19] C. E. Rasmussen, C. K. I. Williams, Gaussian Processes for Machine

Learning, Adaptive Computation and Machine Learning, MIT Press, Cambridge, Mass, 2006.

[20] J. Wang, An Intuitive Tutorial to Gaussian Processes Regression, 2022. arXiv:2009.10862.

[21] R. Kiran, K. Khandelwal, Gurson model parameters for ductile fracture simulation in ASTM A992 steels, Fatigue & Fracture of Engineering Materials & Structures 37 (2014) 171–183. doi:10.1111/ffe.12097.

[22] E. D. Meade, Experimental Study & Multiscale Modelling of the High Temperature Deformation of P91 under Multiaxial Loading, Ph.D. thesis, University of Limerick, 2020.

[23] D. Systemes, Abaqus/CAE User's Guide - SIMULIA User Assistance 2021, Dassault Systemes, 2021.

[24] M. Uy, J. K. Telford, Optimization by Design of Experiment techniques, in: 2009 IEEE Aerospace Conference, 2009, pp. 1–10. doi:10.1109/AERO.2009.4839625.

[25] S. Fraley, M. Oom, B. Terrien, J. Zalewski, 14.1: Design of Experiments via Taguchi Methods - Orthogonal Arrays, in: Chemical Process Dynamics and Controls, 2020.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, É. Duchesnay, Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[27] G. De Ath, R. M. Everson, A. A. M. Rahat, J. E. Fieldsend, Greed is Good: Exploration and Exploitation Trade-offs in Bayesian Optimisation, ACM Transactions on Evolutionary Learning and Optimization 1 (2021) 1–22. doi:10.1145/3425501. arXiv:1911.12809.

[28] A. N. O'Connor, C. M. Davies, S. J. Garwood, The influence of constraint on fracture toughness: Comparing theoretical T0 shifts in master curve analyses with experimental data, Engineering Fracture Mechanics 275 (2022) 108857. doi:10.1016/j.engfracmech.2022.108857.