Article

Data-Oriented Constitutive Modeling of Plasticity in Metals

Alexander Hartmaier

ICAMS, Ruhr-Universität Bochum; alexander.hartmaier@rub.de

Abstract: Constitutive models for plastic deformation of metals are typically based on flow rules determining the transition from elastic to plastic response of a material as function of the applied mechanical load. These flow rules are commonly formulated as a yield function, based on the equivalent stress and the yield strength of the material, and its derivatives. In this work, a novel mathematical formulation is developed that allows the efficient use of machine learning algorithms describing the elastic-plastic deformation of a solid under arbitrary mechanical loads and that can replace the standard yield functions with more flexible algorithms. By exploiting basic physical principles of elastic-plastic deformation, the dimensionality of the problem is reduced without loss of generality. The data-oriented approach inherently offers a great flexibility to handle different kinds of material anisotropy without the need for explicitly calculating a large number of model parameters. The applicability of this formulation in finite element analysis is demonstrated, and the results are compared to formulations based on Hill-like anisotropic plasticity as reference model. In future applications, the machine learning algorithm can be trained by hybrid experimental and numerical data, as for example obtained from fundamental micromechanical simulations based on crystal plasticity models. In this way, data-oriented constitutive modeling will also provide a new way to homogenize numerical results in a scale-bridging approach.

Keywords: plasticity; machine learning; constitutive modeling

1. Introduction

Finite Element Analysis (FEA) is a widespread numerical tool for studying the mechanical behavior of structures. While in many applications it is sufficient to know under which conditions a part of the structure fails plastically or suffers damage or fracture, in some cases, like in sheet forming or for crash simulations, it is important to be able to simulate the plastic deformation during loading and to obtain the shape of the structure after the external load is released. Such non-linear behavior is typically described by constitutive models that relate stress and strain in a material, as described in any textbook on non-linear finite element modeling, e.g., see [1]. Conventionally, constitutive relations for plasticity are formulated as flow rules based on a plastic potential. In the simplest case, the latter is the yield function of the material, determining at which local stress the material starts yielding plastically and which plastic strain increment will result from such plastic deformation. As described in the next section, such yield functions relate the equivalent stress and the yield strength of a material, which needs to be determined experimentally or with the help of more fundamental models in a scale-bridging approach.

Experimentally, the yield strength is typically determined in uniaxial tensile tests, in which materials frequently exhibit an anisotropy in their plastic behavior, i.e., the yield strength depends on the orientation of the loading axis with respect to the material coordinate system, defined for example by rolling, normal, and transverse direction. In conventional approaches such anisotropic plastic behavior is described by defining a proper equivalent stress that takes into account material anisotropy, such that yielding occurs at a constant scalar yield strength. This approach has been introduced by Hill [2] and applied to orthotropic

plasticity in sheet metals [3]. The concept has been generalized to linear transformation-based anisotropic yield functions by Barlat *et al.* [4] and to methods describing distortions of the yield surface caused by anisotropic work hardening [5]. In forming technology, similar ideas have been successfully applied to predict the resulting shape of sheet metals after deep drawing [6]. All these approaches have in common that the information about the material anisotropy is mapped into the definition of the equivalent stress, while they differ in the amount of material parameters that is required to describe the anisotropy in the material's flow behavior. To determine these parameters, a series of experiments with different mutual orientations of loading axis and material axis is necessary. Alternatively, micromechanical models, in which discrete representations of the material's microstructure are used together with quite fundamental crystal plasticity models, can be used to calculate the anisotropic flow behavior of a polycrystalline metal [7,8].

To model non-linear material behavior, in more recent approaches, the method of data-based mechanics has been introduced by Kirchdoerfer and Ortiz [9], in which stress-strain data from experimental tests are used directly, rather than constitutive rules. While the first approaches have been limited to elastic structures under static loads, recently, this concept has been extended to dynamics [10] and to inelastic material behavior [11]. These methods are based on a fundamental re-formulation of the basic equations of mechanics and thus require completely new mechanical solvers. Other data-driven methods in plasticity are formulated as process models, e.g. for air-bending [12], or focus on the application of data-oriented methods as constitutive models in computational plasticity [13]. The latter idea allows the use of existing FEA solvers for mechanical problems, and is also followed in this work, where conventional constitutive models – formulated in a mathematical closed form – are replaced by machine learning (ML) algorithms, which provide a great flexibility to describe arbitrary mathematical functions, and at the same time, they offer the possibility to handle large data sets and multi-dimensional feature vectors as input. Hence, using ML algorithms as constitutive rules for plastic material behavior offers the possibility to explicitly take into account the microstructural information of the material in constitutive modeling. Furthermore, data resulting from experiment and micromechanical simulations can be hybridized to generate training data sets. An overview on applications of ML and data-mining methods in continuum mechanical simulations of material behavior has been provided by Bock *et al.* [14].

ML algorithms can be classified into algorithms for supervised and unsupervised learning. The former group can be further categorized into classification algorithms, which divide a multi-dimensional feature space into regions with similar properties, and into regressors, which provide linear or non-linear regression functions for the given multi-dimensional data set. Support Vector Machines (SVM) are successfully applied both as classifiers (SVC) [15] and regressors (SVR) [16,17]; an overview on both applications is given in a technical report by Gunn [18]. Since yield functions in continuum plasticity are also employed to subdivide stress space into elastic and plastic regions, this work aims at investigating the possibility of using SVC for the purpose of constitutive modeling in plasticity. An overview on data-mining methods and statistical learning, also covering the SVM method, is given by Hastie *et al.* [19].

The present paper is organized as follows: In the next section, the basic concepts of continuum plasticity are briefly summarized, and a new mathematical formulation is introduced, which enables a data-oriented approach to constitutive modeling. Subsequently, a consistent formulation of the SVC method to serve as yield function in continuum plasticity is introduced, which is then trained with artificial data resulting from a Hill-like reference material. Using such data has the advantage of being able to judge the quality of the approximations in an objective way. In the next step, trained ML yield function is applied as constitutive model in simple finite element simulations to demonstrate its applicability for this purpose. Finally, the results of ML and conventional flow rules are compared and the conclusions drawn from this comparison are presented.

2. Methods

2.1. Anisotropic Continuum Plasticity

In order to describe the elastic-plastic deformation of a material, we introduce the strain tensor $\boldsymbol{\epsilon}$ that describes the deformation of the material and the stress tensor $\boldsymbol{\sigma}$ that describes the forces acting on the surface of the material. Note that tensorial quantities with rank≥ 1 are typeset in bold letters, whereas scalar quantities are represented by standard characters. In the elastic regime, Hooke's law is used as constitutive relation between stress and strain, such that

$$\boldsymbol{\sigma} = \boldsymbol{C}\boldsymbol{\epsilon}, \tag{1}$$

where $\boldsymbol{C}$ is the fourth-rank elasticity tensor of the material. To describe plastic deformation, the yield function of the material is introduced as

$$f(\boldsymbol{\sigma}) = \sigma_{\text{eq}} - \sigma_y, \tag{2}$$

which takes negative values if the equivalent stress σ_{eq} is smaller than the yield strength σ_y of the material, i.e. in the elastic regime. When $f = 0$ plastic yielding sets in, and in case of work hardening, σ_y should be considered as flow stress after this point. Since this work only deals with the onset of plastic yielding, ideal plasticity will be assumed throughout, such that σ_y is a constant, irrespective of the deformation history of the material. Denoting the principal stresses of the stress tensor $\boldsymbol{\sigma}$ as σ_j with $(j = 1, 2, 3)$, the equivalent stress takes the form

$$\sigma_{eq}^{\text{J2}} = \sqrt{\frac{1}{2}\left[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2\right]}, \tag{3}$$

which – following the definition of von Mises (see e.g. the translation of the original work by D. H. Delphenich [20]) – is based on the second invariant of the stress deviator (J2). In conjunction with the yield function of Eq. (2), it describes the onset of plastic yielding for isotropic materials. Note that the formulation in Eq. (3) is intrinsically independent of hydrostatic stress components $p = 1/3\text{Trace}(\boldsymbol{\sigma})$ and thus does not require to explicitly calculate the deviatoric stress

$$\boldsymbol{\sigma}' = \boldsymbol{\sigma} - p\boldsymbol{U}, \tag{4}$$

where $\boldsymbol{U}$ is the unit tensor. By this definition of the equivalent stress, it is inherently assumed that hydrostatic stress components do not affect the plastic flow behavior of the considered material, which is typically fulfilled for metals, but not for polymers or rocks, such that the method formulated here, will mainly apply to metallic materials or, more generally, to materials, where hydrostatic stresses do not influence the plastic behavior.

As described in the introduction, many materials exhibit a directionally dependent yield strength, such that anisotropic flow rules need to be introduced. A first definition of such anisotropic flow rules was introduced by Hill [2], who used a generalized definition of the equivalent stress to achieve a directionally dependent mapping of the equivalent stresses to maintain a constant yield strength. Hence, in this formulation, the anisotropy is considered in the stress rather than in the yield strength. Since the

mathematical formulations in this work are purely based on principal stresses, we use a simplified version of the Hill definition and introduce an Hill-like anisotropic definition of the equivalent stress as

$$\sigma_{\text{eq}} = \sqrt{\frac{1}{2}\left[H_1\left(\sigma_1 - \sigma_2\right)^2 + H_2\left(\sigma_2 - \sigma_3\right)^2 + H_3\left(\sigma_3 - \sigma_1\right)^2\right]}, \tag{5}$$

with only three material parameters H_1, H_2 and H_3, whereas in his original work, Hill introduced three more parameters to scale also the shear components of the stress. Hence, the formulation introduced here, will apply for materials with orthotropic flow anisotropy. The definition of the equivalent stress following Hill can be considered as a generalization of the J2 equivalent stress, because for isotropy, i.e., $H_1 = H_2 = H_3 = 1$, both definitions are equal.

   Note that in this work, the mathematical notation is simplified by only considering principal stresses. This formulation is still seen as general, because for any stress state, there exists a coordinate system in which the given stress tensor becomes a diagonal tensor composed of the principal stresses σ_j. This coordinate system is given by the eigenvectors of the stress, such that the coordinate system of the original stress tensor – and with it the material axis – can be rotated into the coordinate system of the eigenvectors of the stress tensor. In this orientation the stress tensor becomes a diagonal tensor, and Eq. (5) can be evaluated with parameters H'_j in the rotated state of the material axis.

   The thus defined yield function can be used to determine whether a given stress state results in a purely elastic or rather in an elastic-plastic deformation of a material. The condition $f(\boldsymbol{\sigma}) = 0$ relates stresses lying on a specific hyperplane in stress space, the so-called the yield-locus. Since a material does not sustain any stresses larger than the yield stress (for ideal plasticity) or the flow stress (in case of work hardening), acceptable stress states either produce a negative value of the yield function (elasticity) or lie on the yield locus (plasticity), which should be a convex hull of the elastic stress states. Hence, if a predictor step in finite element analysis (FEA) produces a stress outside the yield locus, a plastic strain increment must be calculated that leads again to an accepted stress state on the yield locus. The return mapping algorithm to calculate such strain increments has been described in many text books on continuum plasticity and non-linear FEA, such that here only a very brief summary based on [1] is reproduced. According to the Prandtl-Reuss flow rule, the plastic strain increment for a given time step can be calculated as

$$\dot{\boldsymbol{\epsilon}}_p = \dot{\lambda}\frac{\partial f}{\partial \boldsymbol{\sigma}} = \dot{\lambda}\boldsymbol{n}, \tag{6}$$

where $\boldsymbol{n}$ is the normal vector to the yield locus, defined by the gradient of the yield function $\partial f/\partial \boldsymbol{\sigma}$, and $\dot{\lambda} > 0$ is the so-called plastic strain multiplier that can be evaluated as

$$\dot{\lambda} = \frac{\boldsymbol{n}\cdot \boldsymbol{C}\dot{\boldsymbol{\epsilon}}}{\boldsymbol{n}\cdot \boldsymbol{C}\boldsymbol{n}}, \tag{7}$$

where $\dot{\boldsymbol{\epsilon}}$ is the total strain increment of the FEA predictor step that leads to a stress state outside the yield locus and which is consequently decomposed into the plastic strain increment, given by Eq. (6), and the elastic strain increment or stress increment given by

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{C}_t\dot{\boldsymbol{\epsilon}} \tag{8}$$

with the tangent stiffness tensor

$$\boldsymbol{C}_t = \boldsymbol{C} - \frac{\boldsymbol{C}\boldsymbol{n}\otimes \boldsymbol{C}\boldsymbol{n}}{\boldsymbol{n}\cdot \boldsymbol{C}\boldsymbol{n}} \tag{9}$$

where "$\otimes$" denotes the tensorial product in the form $a_i \otimes b_j = a_i b_j$.

The gradient of the yield function with respect to the principal stresses can be evaluated analytically as

$$
\begin{aligned}
\frac{\partial f}{\partial \sigma_1} &= \frac{\partial \sigma_{eq}}{\partial \sigma_1} = \frac{(H_1 + H_3)\,\sigma_1 - H_1 \sigma_2 - H_3 \sigma_3}{\sigma_{eq}} \\
\frac{\partial f}{\partial \sigma_2} &= \frac{\partial \sigma_{eq}}{\partial \sigma_2} = \frac{(H_2 + H_1)\,\sigma_2 - H_1 \sigma_1 - H_2 \sigma_3}{\sigma_{eq}} \\
\frac{\partial f}{\partial \sigma_3} &= \frac{\partial \sigma_{eq}}{\partial \sigma_3} = \frac{(H_3 + H_2)\,\sigma_3 - H_3 \sigma_1 - H_2 \sigma_2}{\sigma_{eq}}
\end{aligned}
\tag{10}
$$

Note that in the case of isotropic plasticity ($H_1 = H_2 = H_3 = 1$), the gradient takes the simple form

$$
\frac{\partial f}{\partial \boldsymbol{\sigma}} = 3 \frac{\boldsymbol{\sigma} - p\boldsymbol{U}}{\sigma_{eq}} = 3 \frac{\boldsymbol{\sigma}'}{\sigma_{eq}}.
\tag{11}
$$

This section served the purpose to introduce the main physical quantities in the notation used in this work. For further details of continuum plasticity or FEA, the reader is referred to standard textbooks, as for example [1]. In the following, the formalism for the data-oriented constitutive model based on a machine learning (ML) yield function is laid out.

2.2. Stress space in cylindrical coordinates

Since plastic deformation in most metals does not depend on hydrostatic stress components, it is useful to transform principal stresses from their representation as a 3-dimensional (3D) Cartesian vector of principal stresses $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \sigma_3)$ into a cylindrical coordinate system with $\boldsymbol{s} = (\sigma_{eq}, \theta, p)$, where the equivalent stress σ_{eq} represents the norm of the stress deviator $\boldsymbol{\sigma}'$, and the polar angle θ lies in the deviatoric plane normal to the hydrostatic axis p, which has already been used by Hill [2]. This coordinate transformation improves the efficiency of the training, because only two-dimensional data for the equivalent stress and the polar angle need to be used as training features, whereas the hydrostatic component is disregarded. Hence, by exploiting basic physical principles, we effectively reduce the dimensionality of the problem from 6 independent components of an arbitrary stress tensor to 2 degrees of freedom, without loosing the generality of the formulation. As the polar angle θ can be considered a generalized Lode angle [21], it is noted that the Lode angle, by definition, describes the axiality of a loading state in a way that uniaxial loads in different directions result in the same Lode angle. Since our formulation aims at describing anisotropy in the plastic deformation, uniaxial stresses in different directions must possess different angles. To achieve this, we introduce a complex-valued deviatoric stress

$$
\sigma_c' = \boldsymbol{\sigma} \cdot \boldsymbol{a} + i\,\boldsymbol{\sigma} \cdot \boldsymbol{b} = \sqrt{2/3}\,\sigma_{eq}\, e^{i\theta},
\tag{12}
$$

where i is the imaginary unit, such that the polar angle

$$
\theta = \arg \sigma_c' = -i\,\ln \frac{\boldsymbol{\sigma} \cdot \boldsymbol{a} + i\,\boldsymbol{\sigma} \cdot \boldsymbol{b}}{\sqrt{2/3}\,\sigma_{eq}}.
\tag{13}
$$

with the unit vectors $\boldsymbol{a} = (2, -1, -1)/\sqrt{6}$ and $\boldsymbol{b} = (0, 1, -1)/\sqrt{2}$ that span the plane normal to the hydrostatic axis $\boldsymbol{c} = (1, 1, 1)/\sqrt{3}$.

To transform the gradient of the yield function from this cylindrical stress space back to the principle stress space, in which form it is used to calculate the direction of the plastic strain increments in the

return mapping algorithm of the plasticity model, we introduce the Jacobian matrix for this coordinate transformation as

$$
\boldsymbol{J} = \frac{\partial \boldsymbol{s}}{\partial \boldsymbol{\sigma}} = \begin{pmatrix} \frac{\partial \sigma_{eq}}{\partial \sigma_1} & \frac{\partial \theta}{\partial \sigma_1} & \frac{\partial p}{\partial \sigma_1} \\ \frac{\partial \sigma_{eq}}{\partial \sigma_2} & \frac{\partial \theta}{\partial \sigma_2} & \frac{\partial p}{\partial \sigma_1} \\ \frac{\partial \sigma_{eq}}{\partial \sigma_3} & \frac{\partial \theta}{\partial \sigma_3} & \frac{\partial p}{\partial \sigma_1} \end{pmatrix} \tag{14}
$$

where $\partial \sigma_{eq} / \partial \sigma_j$ is given in Eqs. (10), $\partial p / \partial \sigma_j = 1/3$ and

$$
\frac{\partial \theta}{\partial \boldsymbol{\sigma}} = -i \left(\frac{\boldsymbol{a} + i\boldsymbol{b}}{\boldsymbol{\sigma} \cdot \boldsymbol{a} + i\boldsymbol{\sigma} \cdot \boldsymbol{b}} - \frac{3\boldsymbol{\sigma}'}{\sigma_{eq}^2} \right). \tag{15}
$$

With this Jacobian, the gradient can be calculated as

$$
\frac{\partial f}{\partial \boldsymbol{\sigma}} = \boldsymbol{J} \frac{\partial f}{\partial \boldsymbol{s}}. \tag{16}
$$

2.3. Data-oriented yield function

While the concept of mapping the equivalent stress in describing anisotropic flow behavior has been applied successfully in the approaches of Hill [2,3] and Barlat [4], for a data-oriented yield function, it is impracticable to calculate the necessary parameters for this stress mapping explicitly. Hence, it is of advantage to reformulate the flow rule in such a way that the yield strength is considered to be directionally dependent, whereas the equivalent stress is formulated in an objective way, without prior knowledge of the material behavior. This is achieved by using the J2 equivalent stress in the flow rule and formally considering the flow stress to be a function of the polar angle in the deviatoric plane, such that

$$
f_d(\boldsymbol{s}) = s_1 - \sigma_y(s_2) = \sigma_{eq}^{J2} - \sigma_y(\theta). \tag{17}
$$

A further advantage of this formulation is that the dependence of the yield function on the two degrees of freedom of the cylindrical stress notation is separated into two independent terms. Furthermore, for symmetry reasons, it is required that the yield strength is a periodic function of the polar angle. The gradient of the ML yield function w.r.t. the cylindrical coordinates reads

$$
\frac{\partial f_d}{\partial s_1} = \frac{\partial f_d}{\partial \sigma_{eq}^{J2}} = 1
$$

$$
\frac{\partial f_d}{\partial s_2} = \frac{\partial f_d}{\partial \theta} = \frac{d\sigma_y}{d\theta}
$$

$$
\frac{\partial f_d}{\partial s_3} = \frac{\partial f_d}{\partial p} = 0.
$$

It is seen that in the cylindrical stress space $\partial f_d / \partial \sigma_{eq} = 1$ and $\partial f_d / \partial p = 0$, under the condition that plasticity is independent of hydrostatic stress components. Hence, the only non-constant component of the gradient is $\partial f_d / \partial \theta$, which simplifies the numerical implementation of the method. For isotropic plasticity, $\partial f_d / \partial \theta = 0$, and in this case it is particularly easy to calculate the gradient and to see that the formulations in both coordinate systems result in the same gradient. The transformation of this gradient into the principal stress space is achieved by multiplication with the Jacobian, according to Eq. (16).

To establish a data-oriented formulation, we introduce a yield function in the form of a machine learning (ML) algorithm, rather than in a mathematically closed form with a number of model parameters that need to be fitted to the data. This enables us to use the available data directly for the training of the

ML algorithm. Furthermore, ML methods allow for the use of higher dimensional feature vectors, such that – in future work – information about the material properties and the microstructure of the material can be directly used as input into one single ML yield function able to handle different microstructures.

In this work, Support Vector Classification (SVC) is applied to categorize data sets consisting of principal stresses into the classes "elastic" and "plastic". During training, SVC constructs a hyperplane in stress space, which separates the two regions from each other. This hyperplane, consequently, is the equivalent to the yield locus, defined by the zeros of the yield function, and it is constructed such that it has the largest distance to the nearest training data points of both classes. It is, therefore, essential to have sufficiently many training data points in close proximity to the yield locus. Data points that lie deeper within the elastic and plastic regions are required to prevent the decision function from falling back to zero, which would produce erroneous results. Such data points, however, can be constructed from available raw data lying close to the yield locus by linearly scaling principal stresses in the elastic region towards smaller values, such that they stay within the elastic region, and, likewise in the plastic region, scaling principal stress data towards higher values. Thus, the raw data can be spread throughout the stress space, without knowing the strain value associated with each constructed data point. Only the knowledge of its class "elastic" or "plastic" is required for this data extension step necessary for the training purpose. The SVC decision function is defined as [15]

$$f_{\text{SVC}}(\boldsymbol{s}) = \sum_{k=1}^{n} y_k \alpha_k K(\boldsymbol{s}_{\text{sv}}^{(k)}, \boldsymbol{s}) + \rho \,, \tag{18}$$

where n is the number of support vectors identified during the training process and

$$K(\boldsymbol{s}_{\text{sv}}^{(k)}, \boldsymbol{s}) = \exp\left(-\gamma \|\boldsymbol{s} - \boldsymbol{s}_{\text{sv}}^{(k)}\|^2\right) \tag{19}$$

is the Radial Basis Function (RBF) kernel of the SVC, which is well suited for non-linear problems, with the parameter γ that determines how fast the influence of one support vector decays in stress space. The support vectors $\boldsymbol{s}_{\text{sv}}^{(k)}$, the dual coefficients $y_k \alpha_k$, and the intercept ρ are determined during the training. There are essentially two parameters that control the training process and thus the quality of the obtained decision function: (i) $\gamma > 0$, which is a parameter of the RBF kernel function and controls how far-reaching the influence of each support vector is: the larger the value of γ, the more short-ranged and local the influence. (ii) $C > 0$, which is a parameter used during the training to regularize the decision function. The larger the value of C, the more flexible but irregular the decision function will become by approximating the shape of the training data more accurately. The choice if these training parameters is critical for the successful use of the decision function in a flow rule. In short, the larger both values are, the more flexible and sensitive to local values the resulting decision function will become. Thus, too small values will result in a smooth but not accurate approximation of the true yield function, whereas too large values will result in a noisy yield function that cannot be used in FEA. The numerical example presented later on will demonstrate this effect and provide examples for values producing accurate yet sufficiently smooth results for the yield function.

For the supervised training, a set of n_t feature vectors $\boldsymbol{s}_{\text{train}}^{(j)}$ has to be provided together with the result vector $y^{(j)}$ with $j = (1, ..., n_t)$ and with values in two categories: $y^{(j)} = -1$ for those training data points $\boldsymbol{s}_{\text{train}}^{(j)}$ in the "elastic" regime and $y^{(j)} = +1$ for training data in the "plastic" regime. This training data are within the core of the method outlined here, because they are directly used to define the plastic properties of the material. It is noted again, that for each principal stress used as training data point, only the resulting material behavior in the categories "elastic" or "plastic" needs to be known but no other information associated with this data point, as for example the plastic strain at this stress is required.

As seen from Eq. (18), the decision function is a continuous function constructed in an way to reproduce this category, i.e., the sign of the training data in the respective regions in the optimal way. To make predictions about the elastic or plastic material behavior at any given stress, the sign of the value of the decision function $f_{\text{SVC}}(\boldsymbol{s})$ at the given stress is evaluated. Furthermore, the yield locus of the material can be obtained in the same way as for traditional yield functions, simply by finding the zeros of the continuous function. In this way, furthermore, the distance of any point in stress space to the yield locus can be evaluated, which is important for making efficient predictor steps during FEA and for calculating plastic strain increments for the return mapping algorithm. In particular, as described in the previous section, the gradient to the yield locus needs to be known in order to calculate plastic strain increments, that bring the stress back to the yield surface, because the plastic material does not support any stresses outside. Due to the definition of the ML yield function as convolution sum over the support vectors, the gradient to the SVC decision function can be calculated analytically as

$$\frac{\partial f_{\text{SVC}}}{\partial \boldsymbol{s}} = \sum_{k=1}^{n} y_k \alpha_k \frac{\partial K(\boldsymbol{s}_{\text{sv}}^{(k)}, \boldsymbol{s})}{\partial \boldsymbol{s}} \tag{20}$$

with

$$\frac{\partial K(\boldsymbol{s}_{sv}^{(k)}, \boldsymbol{s})}{\partial \boldsymbol{s}} = -2\gamma \exp\left(-\gamma \|\boldsymbol{s} - \boldsymbol{s}_{\text{sv}}^{(k)}\|^2\right)\left(\boldsymbol{s} - \boldsymbol{s}_{\text{sv}}^{(k)}\right). \tag{21}$$

The gradient of the yield function in the 3D principle stress space is obtained by multiplication of the gradient in the cylindrical stress space with the Jacobian defined in Eq. (14). Thus, the formulation of the data-oriented yield function based on the SVC algorithm can be used directly as ML yield function in FEA.

3. Results

In the following, it will be demonstrated how the derived formulation of the ML yield function can be trained with data obtained from a reference material and used in FEA as constitutive law for plasticity. All numerical examples are conducted with the tools provided on the open-source platform Sci-Kit Learn [22] and a Python library for FEA written by the author of this work. The Python code used for generating the results presented here is provided as supplementary material in form of a Jupyter notebook.

3.1. Training of ML yield function

A reference material with Hill-type anisotropy is defined with the material parameters as given in Table 1. The yield locus of this reference material for plane-stress conditions with $\sigma_3 = 0$ is plotted in Fig. 1, in which the yield locus of the reference material with Hill-like anisotropy is compared with that of an isotropic material with the same yield strength σ_y.

Table 1. Elastic and plastic material parameters defining the reference material with Hill-like anisotropy in its plastic flow behavior. For simplicity, isotropic elastic behavior and ideal plasticity without work hardening are assumed in this work.

Quantity	Symbol	Value
Young's modulus	E	200 GPa
Poisson's number	ν	0.3
Yield strength	σ_y	150 MPa
Hill parameters	H_1, H_2, H_3	0.7, 1, 1.4
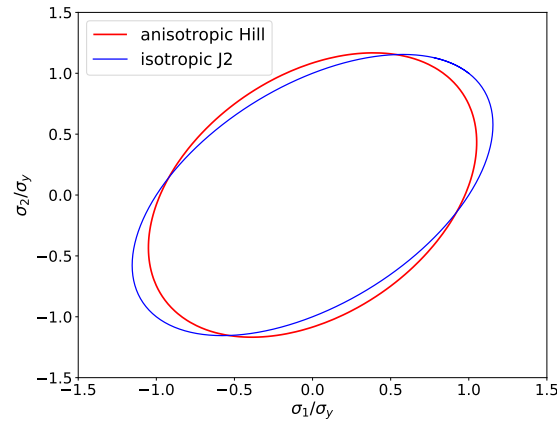
Figure 1. Yield locus for plane-stress conditions ($\sigma_3 = 0$) and Hill-like anisotropy with parameters given in Table 1 (red line) and for an isotropic material with the same yield strength, but $H_1 = H_2 = H_3 = 1$ (blue line). The values of the principal stresses are normalized by the yield strength σ_y.

The thus-defined reference material is used to produce training data – and later also test data – for the machine learning algorithm. To accomplish this, a set of stress values in form of principal deviatoric stresses is produced in a way to cover the complete space of polar angles and also sufficiently many equivalent stresses in the elastic and plastic regimes. This is conveniently achieved by creating a set of n_ang equally distributed polar angles $\theta^{(k)}$ in the range of $[-\pi, \pi]$ and a set of n_s equivalent stresses $\sigma_{eq}^{\text{J2}\,(l)}$ in the range $[0.1\sigma_y, 5\sigma_y]$. Note that for the entire procedure, the yield strength σ_y of the material is assumed to be known. This is not a restriction, because the yield strength of an unknown material can be easily determined from the input data in a pre-analysis step.

The transformation into principal stresses is performed as

$$\boldsymbol{\sigma}_\text{train}^{(j)} = \sqrt{2/3}\, \sigma_{eq}^{\text{J2}\,(l)} \left(\boldsymbol{a} \cos \theta^{(k)} + \boldsymbol{b} \sin \theta^{(k)} \right) \tag{22}$$

with

$$j = k + (l-1) n_\text{ang} \qquad (k = 1, ..., n_\text{ang}; \; l = 1, ..., n_s) \tag{23}$$

and the unit angles $\boldsymbol{a}$ and $\boldsymbol{b}$ spanning the deviatoric stress plane as given above. This produces a set of $n_t = n_\text{ang} n_s$ principal stresses with which, finally, the set of result vectors

$$y^{(j)} = \text{sgn} \left(f(\boldsymbol{\sigma}_\text{train}^{(j)}) \right) \qquad \text{with} \qquad j = (1, ..., n_t), \tag{24}$$

is generated by evaluating the yield function f of the reference material, as defined in Eqs. (2) and (5), with the material parameters given in Table 1.

In the numerical example given here, the full training data set comprises $n_\text{ang} = 36$ values for the polar angle and $n_s = 28$ values for the equivalent stress for each angle, resulting in a total of $n_t = 1008$ training data sets. Concerning the effort to create this data, it is noted here that only the number of angles n_ang is relevant for the number of experiments or micromechanical simulations necessary to generate the training data, because each angle defines a load case from which several stresses in the elastic and plastic regime will result, and other training data points can be easily constructed from this raw data by linear scaling, as described above. The implications of the number of load cases required to achieve an accurate representation of the ML yield function will be further discussed in section 4. A graphical representation

of the training data is shown in Fig. 2, where also the different ways of representing the anisotropy of the yield function with Hill-type equivalent stresses and von Mises (J2) equivalent stresses are demonstrated. The actual training data comprise four additional sets of polar angles associated with larger equivalent stresses scaled to values of up to $\sigma_{\mathrm{eq}} = 5\sigma_y$ to prevent the decision function from falling back to zero in this regime, which might cause erroneous results in FEA. To enforce the periodicity of the training ML yield function and its gradient, the training data is periodically repeated within the training algorithm, such that the polar angle covers a range $-1.3\pi < \theta < 1.3\pi$.
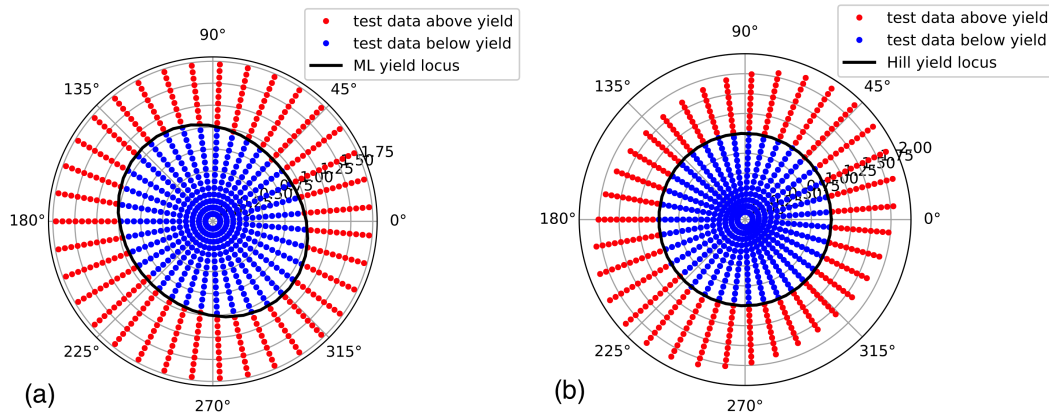

Figure 2. Polar plots of a subset of the training data produced from the anisotropic yield function of the reference material: (**a**) Von Mises (J2) equivalent stresses according to Eq. (3) are used, such that the yield strength, rather than the equivalent stress, is a function of the polar angle θ. (**b**) Equivalent stresses are calculated according to the Hill definition in Eq. (5) to achieve a constant yield strength by mapping the equivalent stresses accordingly. In both figures, the yield locus is indicated by a solid black line, data points in the elastic regime are plotted in blue color and data in the plastic regime in red color. Both figures represent the same stress data – only mapped in a different way; all stresses are normalized with the reference yield strength σ_y.

With this data set, the training of the SVC algorithm is performed. Using the training parameters $C = 10$ and $\gamma = 4$ results in a very good training score of above 99%. However, to evaluate the true quality of the training procedure and to judge whether overfitting has occurred, it is necessary to verify the results with an independent set of test data, which has not been used for training purposes. The error produced on such test data sets with 480 random deviatoric stresses as data points is below 1%, and the R^2-correlation coefficient between test data and training data is above 98%, which leads to the conclusion that the trained ML yield function has a very high accuracy and robustness. A variation of the training parameters revealed that the results are rather insensitive to the parameter C, which can be varied between $2 < C < 20$ without having a pronounced influence on the results, whereas changing the parameter γ by more then 20% causes a significant deterioration of the training results. The resulting SVC decision function, defined in Eq. (18), is plotted together with the training data in Fig. 3 in the deviatoric stress space.
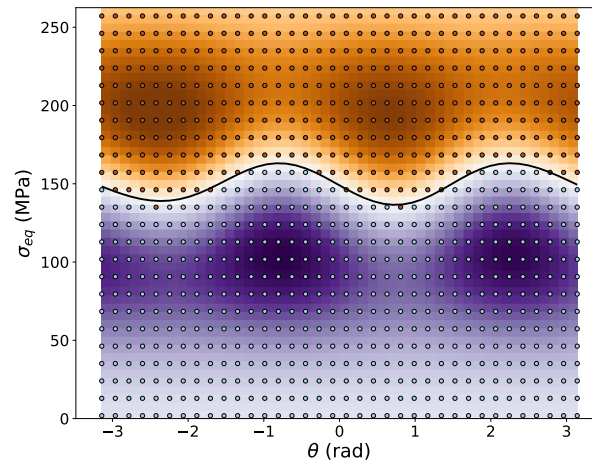
Figure 3. Field plot of the trained SVM decision function defined in Eq. (18), where areas in purple color shades represent negative values and brown shades represent positive values. The numerical value of the decision function is not relevant because only its sign is taken into account in the flow rule. The isoline for $f_{\mathrm{SVC}} = 0$ is represented as a black line. Training data are plotted in light blue color for data with negative values (elastic) and in brown color for positive values (plastic).

Finally, to demonstrate the quality of the ML yield function in the full principal stress space, the predicted categories are plotted in three different slices corresponding to different plane-stress conditions, together with the yield function of the reference material in Fig. 4. It is seen that the training data in the deviatoric space covers in fact only a single line in each slice, which demonstrates the power of reducing the dimensionality of the data-oriented yield function by exploiting basic physical principles.
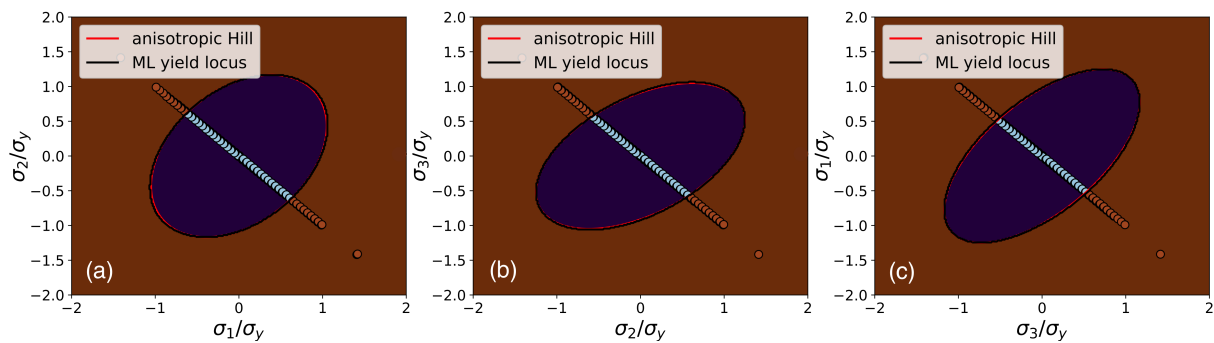
Figure 4. Color map of the trained SVC prediction of the yield function in slices through the principal stress space defined by plane-stress conditions: (**a**) $\sigma_3 = 0$; (**b**) $\sigma_1 = 0$; (**c**) $\sigma_2 = 0$. Brown regions indicate values of "+1" (plasticity) and purple regions values of "-1" (elasticity). The ML yield locus, corresponding to the isoline for $f_{\mathrm{SVC}} = 0$, is represented as a black line; the yield locus of the Hill-like anisotropic reference material is indicated as a red line. The training data points are plotted with the same color code as in Fig. 3.

3.2. Application of the trained ML yield function in FE analysis

The ML yield function trained and analyzed in the previous step shall now be applied in FEA to demonstrate its usefulness for this purpose. The numerical examples provided here have been conducted with the Python library "pyLab-FE" created by the author, which is provided in the supplementary materials together with a Jupyter notebook following the work-flow defined in this work. The known

parameters and support vectors resulting from the training process of the ML yield function together with the mathematical formalism laid out in section 2 allow a rather straightforward evaluation of the yield function as sum over the support vectors convoluted with the kernel function, such that they can also be used for implementing a user material subroutine (UMAT) for common commercial FEA tools in any compiler language.

As numerical examples, four different load cases are simulated with FEA: (i) uniaxial stress in horizontal direction, (ii) uniaxial stress in vertical direction, (iii) equibiaxial strain under plane-stress conditions, and (iv) pure shear strain under plane-stress conditions. These four load cases are applied to the reference material as well as to the material with the ML yield function, and the resulting yield stresses and plastic strains at the end of each load step are compared in Table 2. In Fig. 5 the resulting global equivalent stresses and equivalent total strains

$$\epsilon_{\text{eq}} = \sqrt{\frac{2}{3}\boldsymbol{\epsilon} : \boldsymbol{\epsilon}} \qquad (25)$$

for each load case are plotted for both materials, where the different definitions of the equivalent stress have been applied to the reference material.

Table 2. Yield stresses (YS) obtained for Hill-like yield function, with parameters given in Table 1, and machine learning (ML) yield function under the specified load cases. The relative errors in yield stress and equivalent plastic strain (PE) at maximum load are also specified.

Load Case	YS-Hill (MPa)	YS-ML(MPa)	rel. error YS	rel. error PE
uniaxial stress, horizontal	146.4	147.4	0.69%	4.56%
uniaxial stress, vertical	162.7	161.5	-0.73%	0.49%
equibiaxial strain	136.9	138.5	1.17%	-0.02%
pure shear strain	161.1	160.3	-0.54%	-0.40%
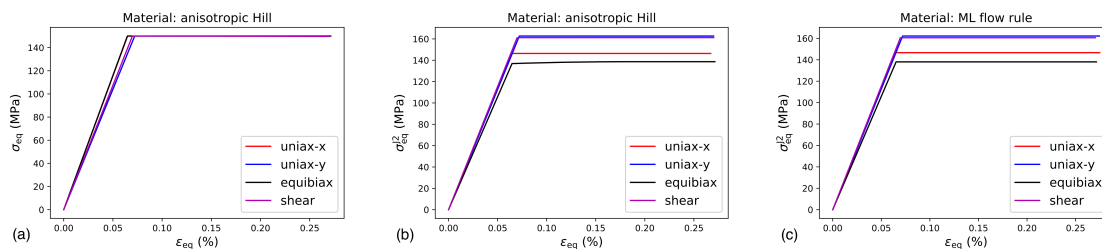
Figure 5. Stress strain curves obtained for elastic-ideal plastic material behavior under the loading conditions specified in the legend: (**a**) Equivalent total strain vs. equivalent Hill-stress, (**b**) equivalent total strain vs. equivalent J2-stress for Hill-like yield function, and (**c**) equivalent total strain vs. equivalent J2-stress for ML yield function.

With these numerical examples, the applicability of the ML yield function developed in this work has been demonstrated. The new formulation has proven to be numerically stable. The numerical effort is somewhat higher than that for the mathematical closed-form yield function, because the calculation of the predictor step requires a higher effort in numerically evaluating the distance of a given point in stress space to the yield locus. However, in conjunction with the implementation of the ML yield function in a compiled computer code, FEA even for large engineering models seems to be feasible with the new ML yield functions.

4. Discussion

In this section the requirements on training data for the ML yield function will be more closely examined. For the first application of the ML yield function, training data for $n_{\mathrm{ang}} = 36$ load cases have been constructed from a reference material with Hill-like anisotropy, and it has been demonstrated that this number of load cases produces rather accurate results. Numerical studies with larger data sets reveal that the accuracy of the results increases slightly for data sets of up to $n_{\mathrm{ang}} = 200$. The accuracy as well as the numerical stability and efficiency of the method remain stable for even larger data sets, which has been tested for up to $n_{\mathrm{ang}} = 600$, however, the accuracy does not improve further. It is also interesting to see that the accuracy of the method is only slightly reduced when rather small data sets of $n_{\mathrm{ang}} = 8$ load cases are used. Even producing the training data in plane-stress conditions with $\sigma_3 = 0$ does not change the quality of the results significantly, which is a consequence of mapping all stresses onto the deviatoric plane and extending the results by assuming that the material's flow behavior does not depend on hydrostatic stresses, which is fulfilled to a very good degree for metals. If it is assumed, furthermore, that the material under consideration shows a symmetric flow behavior under tension and compression, only one half-space of the deviatoric plane needs to be characterized, and the results can be mirrored into the other half-plane. Thus, only four mechanical tests are required to produce the raw data for the training of the ML yield function. It can be verified that only knowing the full stress tensors at the onset of plastic yielding in the four uniaxial or plane-stress load cases given in Table 2 is sufficient to produce training data from which a useful ML yield function results. To accomplish this, the assumptions of tension-compression symmetry and of independence of the flow behavior on the hydrostatic stress component are exploited. This example is also provided in the supplementary material.

From these considerations, it can be concluded that for material with an anisotropic flow behavior that can be described with a Hill-like formulation, a small number of experiments under plane-stress conditions, as they can be performed on a bi-axial test rig, are fully sufficient to produce enough raw data to train the ML yield function. Of course, this experimental data could also be used to calculate the Hill parameters. However, the ML yield function offers a larger flexibility, and the training process for machine learning methods like support vector characterization (SVC) is highly efficient. For materials with a more irregular anisotropy in their flow behavior, these advantages become even more prominent, as the effort to determine a larger number of material parameters, like the 18 Barlat parameters [4], increases dramatically, whereas the training effort for the ML yield function remains the same, independent of the volume of training data.

Data-oriented constitutive modeling thus requires only a limited amount of data, as compared to other approaches in data-driven mechanics [9,11,13]. Large data volumes, of course, help to increase the accuracy of the resulting ML yield function, but the results achieved even with small amounts of data provide already a very good estimate of the material's anisotropic flow behavior even for load cases that have not been tested. These comparatively moderate requirements on the volume of training data are a consequence of exploiting physical symmetry conditions on the material's flow behavior in the formulation of the data-oriented yield function. For materials exhibiting a significant influence of the hydrostatic stress component on the plastic behavior, the method is still applicable, but the requirements on the training data will be higher.

Another aspect to be discussed here is the use of micromechanical models to produce training data. With such models, the mechanical behavior of realistic microstructures can be simulated with crystal plasticity methods [7], providing an accurate description of the plastic properties of polycrystaline metals with different microstructures and crystallographic textures. One disadvantage of such micromechanical models is their tremendously high numerical effort making them unsuited for FEA applications of engineering structures that are much larger than the grain size of a material. However, it is possible to

employ relatively small micromechanical models, validated by experimental data, for creating a sufficiently large data volume describing the mechanical properties of the real material under various loading conditions with a high accuracy. By purposefully varying the microstructure or the texture of the model material, micromechanical simulations also serve the purpose of extending experimental data. This hybrid experimental and numerical data can then be used for the training of the ML yield function presented in this work. In this way, material parameters like grain size, grain morphology, and crystallographic texture can be explicitly included into the feature vector of the ML yield functions, in addition to the purely mechanical data used currently as input for the yield function. This microstructure-sensitive ML yield function can then be used in large-scale FEA for the simulation of engineering structures, which holds the possibility to consider the trained ML flow rule as a "digital twin" of the material, containing all relevant information on the material properties. The data-oriented constitutive model developed in this work will, hence, also pave the way for new approaches to scale-bridging materials modeling.

5. Conclusions

In this work, a new formulation of a data-oriented constitutive model for plasticity has been derived and applied within finite element analysis. The central element of this new constitutive model is a Support Vector Characterization (SVC) algorithm serving as yield function. This SVC algorithm is trained by using deviatoric stresses as input data and the information whether a given stress state leads to purely elastic or rather to elastic-plastic deformation of the material as result data. In this way, a machine learning (ML) yield function is obtained, which can determine whether a given stress state lies inside or outside of the elastic regime of the material. Furthermore, the yield locus, i.e., the hyperplane in stress space on which plastic deformation occurs, can be reconstructed from the SVC, and the gradient on this yield locus can be conveniently calculated. Therefore, the standard formulations of continuum plasticity, as the return mapping algorithm, can be applied in finite element analysis in the usual way. Thus, it has been demonstrated that the new ML yield function can replace conventional yield functions in finite element analysis. The main advantage of such data-oriented constitutive models over the conventional ones is that they can be used with higher-dimensional feature vectors combining mechanical stresses with microstructural parameters of a material. In forthcoming work, it will thus be demonstrated how a single ML yield function can be trained to be used as a constitutive rule for a material in different microstructural states. The production of training data by micromechanical models, based on crystal plasticity and a discrete representation of the material's microstructure, allows the ML flow rule to serve as efficient homogenization scheme, which offers new possibilities in scale-bridging material modeling.

Conflicts of Interest: The author declares no conflict of interest.

References

1.   De Borst, R.; Crisfield, M.A.; Remmers, J.J.; Verhoosel, C.V. *Nonlinear finite element analysis of solids and structures*; John Wiley & Sons, 2012.

2. Hill, R. A variational principle of maximum plastic work in classical plasticity. *The Quarterly Journal of Mechanics and Applied Mathematics* **1948**, *1*, 18–28.

3. Hill, R. Constitutive modelling of orthotropic plasticity in sheet metals. *Journal of the Mechanics and Physics of Solids* **1990**, *38*, 405–417.

4. Barlat, F.; Aretz, H.; Yoon, J.W.; Karabin, M.; Brem, J.; Dick, R. Linear transfomation-based anisotropic yield functions. *International Journal of Plasticity* **2005**, *21*, 1009–1039.

5. Soyarslan, C.; Klusemann, B.; Bargmann, S. The effect of yield surface curvature change by cross hardening on forming limit diagrams of sheets. *International journal of mechanical sciences* **2016**, *117*, 53–66.

6. Güner, A.; Soyarslan, C.; Brosius, A.; Tekkaya, A. Characterization of anisotropy of sheet metals employing inhomogeneous strain fields for Yld2000-2D yield function. *International Journal of Solids and Structures* **2012**, *49*, 3517–3527.

7. Vajragupta, N.; Ahmed, S.; Boeff, M.; Ma, A.; Hartmaier, A. Micromechanical modeling approach to derive the yield surface for BCC and FCC steels using statistically informed microstructure models and nonlocal crystal plasticity. *Physical Mesomechanics* **2017**, *20*, 343–352. doi:10.1134/S1029959917030109.

8. Biswas, A.; Prasad, M.R.; Vajragupta, N.; Hassan, H.u.; Brenne, F.; Niendorf, T.; Hartmaier, A. Influence of Microstructural Features on the Strain Hardening Behavior of Additively Manufactured Metallic Components. *Advanced Engineering Materials* **2019**, *21*, 1900275.

9. Kirchdoerfer, T.; Ortiz, M. Data-driven computational mechanics. *Computer Methods in Applied Mechanics and Engineering* **2016**, *304*, 81–101.

10. Kirchdoerfer, T.; Ortiz, M. Data-driven computing in dynamics. *International Journal for Numerical Methods in Engineering* **2018**, *113*, 1697–1710.

11. Eggersmann, R.; Kirchdoerfer, T.; Reese, S.; Stainier, L.; Ortiz, M. Model-free data-driven inelasticity. *Computer Methods in Applied Mechanics and Engineering* **2019**, *350*, 81–99.

12. Vorkov, V.; García, A.T.; Rodrigues, G.C.; Duflou, J.R. Data-driven prediction of air bending. *Procedia Manufacturing* **2019**, *29*, 177–184.

13. Chinesta, F.; Ladeveze, P.; Ibanez, R.; Aguado, J.V.; Abisset-Chavanne, E.; Cueto, E. Data-driven computational plasticity. *Procedia engineering* **2017**, *207*, 209–214.

14. Bock, F.E.; Aydin, R.C.; Cyron, C.J.; Huber, N.; Kalidindi, S.R.; Klusemann, B. A Review of the Application of Machine Learning and Data Mining Approaches in Continuum Materials Mechanics. *Frontiers in Materials* **2019**, *6*, 110. doi:10.3389/fmats.2019.00110.

15. Suykens, J.A.; Vandewalle, J. Least squares support vector machine classifiers. *Neural processing letters* **1999**, *9*, 293–300.

16. Vapnik, V. The support vector method of function estimation. In *Nonlinear Modeling*; Springer, 1998; pp. 55–85.

17. Reimann, D.; Chandra, K.; Vajragupta, N.; Glasmachers, T.; Junker, P.; Hartmaier, A.; others. Modeling macroscopic material behavior with machine learning algorithms trained by micromechanical simulations. *Frontiers in Materials* **2019**, *6*, 181.

18. Gunn, S.R. Support vector machines for classification and regression. *ISIS technical report* **1998**, *14*, 5–16.

19. Hastie, T.; Tibshirani, R.; Friedman, J. *The elements of statistical learning: data mining, inference, and prediction*; Springer Science & Business Media, 2009.

20. Mises, R.v. Mechanics of solid bodies in the plastically-deformable state. *Göttin Nachr Math Phys* **1913**, *1*, 582–592.

21. Lode, W. Versuche über den Einfluß der mittleren Hauptspannung auf das Fließen der Metalle Eisen, Kupfer und Nickel. *Zeitschrift für Physik* **1926**, *36*, 913–939.

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