



# Optimal Data-Generation Strategy for Machine Learning Yield Functions in Anisotropic Plasticity

Ronak Shoghi<sup>†</sup> and Alexander Hartmaier<sup>\*†</sup>

ICAMS, Ruhr-Universität Bochum, Bochum, Germany

Trained machine learning (ML) algorithms can serve as numerically efficient surrogate models of sophisticated but numerically expensive constitutive models of material behavior. In the field of plasticity, ML yield functions have been proposed that serve as the basis of a constitutive model for plastic material behavior. If the training data for such ML flow rules is gained by micromechanical models, the training procedure can be considered as a homogenization method that captures essential information of microstructure-property relationships of a given material. However, generating training data with micromechanical methods, as for example, the crystal plasticity finite element method, is a numerically challenging task. Hence, in this work, it is investigated how an optimal data-generation strategy for the training of a ML model can be established that produces reliable and accurate ML yield functions with the least possible effort. It is shown that even for materials with a significant plastic anisotropy, as polycrystals with a pronounced Goss texture, 300 data points representing the yield locus of the material in stress space, are sufficient to train the ML yield function successfully. Furthermore, it is demonstrated how data-oriented flow rules can be used in standard finite element analysis.

**Keywords:** plasticity, data-driven methods, machine learning, data generation, uniform distribution, hypersphere, homogenization

## OPEN ACCESS

### Edited by:

Norbert Huber,  
Helmholtz-Zentrum Hereon, Germany

### Reviewed by:

Francisco Chinesta,  
Arts et Metiers Institute of Technology,  
France

Christian Johannes Cyron,  
Hamburg University of Technology,  
Germany

### \*Correspondence:

Alexander Hartmaier  
alexander.hartmaier@rub.de

<sup>†</sup>These authors have contributed  
equally to this work

### Specialty section:

This article was submitted to  
Computational Materials Science,  
a section of the journal  
Frontiers in Materials

**Received:** 02 February 2022

**Accepted:** 22 March 2022

**Published:** 27 April 2022

### Citation:

Shoghi R and Hartmaier A (2022)  
Optimal Data-Generation Strategy for  
Machine Learning Yield Functions in  
Anisotropic Plasticity.  
Front. Mater. 9:868248.  
doi: 10.3389/fmats.2022.868248

## INTRODUCTION

The finite element method is one of the most popular methods used in solid mechanics to solve the nonlinear partial differential equations describing mechanical equilibrium of a solid under arbitrary boundary conditions. The solution of a solid mechanics problem must satisfy equations of equilibrium, compatibility of strains and displacements, and obey the constitutive laws for the materials represented in the model (Chen and Saleeb, 1994; Pian and Wu, 2005). These constitutive equations describe the relationships between stresses and strains, i.e., they quantify the material response under given distortions. In the simplest case of elastic materials, this relationship assumes a linear form in which stress and strain are proportional to each other, as described by Hooke's law. For nonlinear and irreversible material behavior, as it must be considered for plastic materials, elastic and plastic strains need to be treated by separate constitutive models (Chen and Saleeb, 1994; Bonet and Wood, 1997). While for elastic strains, Hooke's law is still valid, plastic strains need to be calculated in a way that is consistent with the yield strength and the flow stress observed in tensile tests of the given material. To accomplish this, a yield function is formulated that indicates whether a given stress state results in a linear-elastic material response or whether plastic deformation needs to be considered. In the linear elastic regime, the yield function takes negative values, and it reaches the value zero for those stress tensors for which plastic yielding starts. Hence, the zeros of the yield function can be represented

by a hypersurface in stress space, which embodies the convex hull of the stress tensors leading to a linear-elastic material response. In the Voigt notation, the six-dimensional (6D) stress space itself is spanned by the six independent components of the stress tensor, including normal and shear components. In the case of ideal plasticity, this hypersurface in stress space, the so-called yield-locus, remains unchanged by plastic deformation (Chen and Saleeb, 1994). In contrast, in the case of strain hardening, it can change its volume (isotropic hardening), location (kinematic hardening), or shape (distortive hardening) (Helling and Miller, 1987; Kurtyka and Życzkowski, 1996).

This concept of the yield locus as geometrical representation of the yield criterion can also be applied in cases of anisotropic plastic deformation, as it occurs, for example, in polycrystalline metals with specific crystallographic textures. This plastic anisotropy of polycrystals can be described with the crystal plasticity finite element method (CPFEM) (Roters et al., 2011), for an overview, in which plastic slip on the discrete crystallographic slip planes of each grain is considered. Such methods also allow us to study the evolution of textures under large plastic strains (Peranio et al., 2010), where certain crystallographic planes tend to rotate concerning the direction of the highest principal strain and, thus, cause a dynamic change of the crystallographic texture and the resulting plastic anisotropy, which is the origin of distortive hardening in these materials.

While CPFEM methods represent a fundamental way of describing plastic anisotropy of materials, they cause a high numerical effort, limiting their application to relatively small volumes. In the work of Vajragupta et al. (2017), it has been shown that the results of CPFEM calculations can be homogenized by fitting the parameters of an anisotropic yield criterion to data obtained from CPFEM calculation of specific textures. Various formulations of such anisotropic yield criteria have been published in the literature (Karafillis and Boyce, 1993; Cazacu and Barlat, 2001; Banabic et al., 2004). In this work, we refer to the formulation of Hill (Hill, 1948) with six parameters describing the state of anisotropy, which has been suggested as a generalization of the isotropic yield criterion after Huber-Von Mises (Huber, 1904; v Mises, 1913), based on the second invariant of the stress deviator ( $J_2$ ). The corresponding equivalent stress can be compared to the scalar yield stress of the material determined in a uniaxial test. Barlat et al. (2005) suggested a yield function that contains 18 parameters (Yld 2004-18p) based on a linear transformation of the stress deviator.

In more recent approaches, data-driven computational methods have been suggested in the literature to describe anisotropic material behavior. In the data-driven method suggested by Kirchdoerfer and Ortiz (2016), instead of constitutive models for finite element analysis, experimental material data can be used directly to satisfy the required constraints and conservation laws for mechanical equilibrium. In a similar approach, Eggersmann et al. (2019), Eggersmann et al. (2021) have shown that a model-free data-driven formulation of mechanical problems can be achieved. In the work of Chinesta et al. (2017) the data-driven strategy was extended for nonlinear material behavior, which includes not only the plastic strain rate and the rate of accumulated plastic

deformation but also the kinematic hardening rate. Another popular approach to develop a data-driven material model is using machine learning algorithms that are capable of handling large data sets. At the same time, they provide the possibility of describing arbitrary mathematical functions, thus relieving the restrictions for closed-form mathematical descriptions of anisotropic yield criteria. Liu et al. (2018) employ a clustering technique to solve the equilibrium equation on clusters of material points with similar mechanical responses. In another approach suggested by Ibañez et al. (2018), manifold learning methods were used to define the constitutive manifold of a given material, allowing the extraction of relevant information directly from large experimental data sets. In Linka et al. (2021), a machine learning-based hyperelasticity model is suggested using a feed-forward neural network trained using experimental data. (Linka et al. (2021) introduced constitutive artificial neural networks to describe hyperelastic material behavior.

Following the method developed in (Hartmaier, 2020), the present work uses Support Vector Classification (SVC) of the elastic and plastic domains in the stress space as a data-oriented yield function. The SVC algorithm is trained by using input data in the form of critical stresses that mark the onset of plastic yielding, thus representing the yield locus in a data-based manner. In this way, a machine learning (ML) yield function is obtained to determine whether a given stress state lies inside or outside the material's elastic domain. Getting a proper data-based representation of the yield locus as a hypersurface in the 6D stress space can be challenging. This holds in particular if the training data is generated by numerically expensive methods as CPFEM. Hence, in this work, we will use simpler methods for the generation of various data sets that are the basis for the development of an optimal strategy to distribute the training data points over the entire yield locus with as small as possible data sets. Based on these training data, an accurate ML yield locus, i.e., the hypersurface in the stress space on which plastic deformation occurs, can be reconstructed from the SVC in the form of a convolutional sum over a kernel function, from which 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 the usual way to finite element analysis (FEA). Thus, it is demonstrated that the new ML yield function can replace conventional FEA flow rules.

## METHODS

### Machine Learning Flow Rule

The elastic-plastic deformation of a material can be described using stress and strain tensors denoted with  $\sigma$  and  $\epsilon$ , respectively. The stress tensor describes the force acting on the surface of a material, and the strain tensor describes the deformation of the material.

In the Voigt notation, the symmetric tensor is defined by its six independent components as

$$\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6) \quad (1)$$

where  $\sigma_1 = \sigma_{11}, \sigma_2 = \sigma_{22}, \sigma_3 = \sigma_{33}, \sigma_4 = \sigma_{23}, \sigma_5 = \sigma_{13}, \sigma_6 = \sigma_{12}$ . The yield function of a material is defined as

$$f(\boldsymbol{\sigma}) = \sigma_{eq}(\boldsymbol{\sigma}) - \sigma_y \quad (2)$$

and plastic deformation sets in at  $f = 0$ , i.e., when the equivalent stress  $\sigma_{eq}$  equals the yield strength  $\sigma_y$  of the material. The equivalent stress used here follows the definition of Hill for anisotropic materials in the form

$$\sigma_{eq}(\boldsymbol{\sigma}) = \frac{1}{\sqrt{2}} [H_1(\sigma_1 - \sigma_2)^2 + H_2(\sigma_2 - \sigma_3)^2 + H_3(\sigma_3 - \sigma_1)^2 + 6H_4\sigma_4^2 + 6H_5\sigma_5^2 + 6H_6\sigma_6^2]^{1/2} \quad (3)$$

In this yield function, the anisotropy of the material's flow behavior is described in a Hill-like approach, where the parameters  $H_1, \dots, H_6$  control the anisotropic flow behavior of the material. Note that these parameters correspond to the parameters used in the conventional Hill yield criterion (Hill, 1948).

$$E(\sigma_2 - \sigma_3)^2 + G(\sigma_3 - \sigma_1)^2 + H(\sigma_1 - \sigma_2)^2 + 2L\sigma_4^2 + 2M\sigma_5^2 + 2N\sigma_6^2 = 1 \quad (4)$$

when  $H = H_1\sigma_y^2, E = H_2\sigma_y^2, G = H_3\sigma_y^2, L = 3H_4\sigma_y^2, M = 3H_5\sigma_y^2, N = 3H_6\sigma_y^2$ . The special case  $H_1 = H_2 = \dots = H_6 = 1$  describes isotropic plastic behavior, where the equivalent stress defined here is identical to the von Mises (J2) equivalent stress.

The gradient of the yield function with respect to the stress components is needed for calculating the plastic strain increments in the return mapping algorithm of continuum plasticity and can be evaluated analytically as

$$\frac{\partial f}{\partial \sigma_1} = \frac{\partial \sigma_{eq}}{\partial \sigma_1} = \frac{(H_1 + H_3)\sigma_1 - H_1\sigma_2 - H_3\sigma_3}{2\sigma_{eq}} \quad (5)$$

$$\frac{\partial f}{\partial \sigma_2} = \frac{\partial \sigma_{eq}}{\partial \sigma_2} = \frac{(H_2 + H_1)\sigma_2 - H_1\sigma_1 - H_2\sigma_3}{2\sigma_{eq}} \quad (6)$$

$$\frac{\partial f}{\partial \sigma_3} = \frac{\partial \sigma_{eq}}{\partial \sigma_3} = \frac{(H_3 + H_2)\sigma_3 - H_3\sigma_1 - H_2\sigma_2}{2\sigma_{eq}} \quad (7)$$

$$\frac{\partial f}{\partial \sigma_4} = \frac{\partial \sigma_{eq}}{\partial \sigma_4} = 3H_4 \frac{\sigma_4}{\sigma_{eq}} \quad (8)$$

$$\frac{\partial f}{\partial \sigma_5} = \frac{\partial \sigma_{eq}}{\partial \sigma_5} = 3H_5 \frac{\sigma_5}{\sigma_{eq}} \quad (9)$$

$$\frac{\partial f}{\partial \sigma_6} = \frac{\partial \sigma_{eq}}{\partial \sigma_6} = 3H_6 \frac{\sigma_6}{\sigma_{eq}} \quad (10)$$

In the case of isotropic plasticity, i.e.,  $H_1 = H_2 = \dots = H_6 = 1$  the gradient takes the simple form

$$\frac{\partial f}{\partial \boldsymbol{\sigma}} = \frac{3}{2} \frac{\boldsymbol{\sigma}^{dev}}{\sigma_{eq}} \quad (11)$$

Where  $\boldsymbol{\sigma}^{dev} = (\sigma_1 - p, \sigma_2 - p, \sigma_3 - p, \sigma_4, \sigma_5, \sigma_6)$  is the deviatoric stress tensor and  $p = (\sigma_1 + \sigma_2 + \sigma_3)/3$  is the hydrostatic stress.

## Data-Oriented Yield Function

In the data-oriented approach followed in this work, the yield function  $f_{ML}(\boldsymbol{\sigma})$  is described in the form of a machine learning (ML) algorithm, which uses Support Vector Classification (SVC) for categorizing any given stress tensor  $\boldsymbol{\sigma}$  into the categories

“elastic” ( $f_{ML}(\boldsymbol{\sigma}) = -1$ ) and “plastic” ( $f_{ML}(\boldsymbol{\sigma}) = +1$ ) (Hartmaier, 2020). The purpose is to find the optimal hypersurface which separates these two regions from each other. This hypersurface is the yield locus defined by the zeros of the yield function. Based on the SVM algorithm, the optimal hypersurface is the one in which the margin between training data points of the respective classes “elastic” and “plastic” is maximum. This margin is defined as the distance between the separator and the closest data points to it from both classes. These data points in the vicinity of the separator are called support vectors. Given a training set of  $N$  data points  $\{y_i, \boldsymbol{\sigma}_i\}_{i=1}^N$  where  $\boldsymbol{\sigma}_i \in \mathbb{R}^6$  is the  $i$ th input stress and  $y_i = f_{ML}(\boldsymbol{\sigma}_i)$  is the  $i$ th output term required for the supervised training algorithm. Note that this data-oriented yield function  $f_{ML}$  can be considered as the signum function of the physical yield function defined in Eq. 2, thus  $f_{ML}(\boldsymbol{\sigma}) = \text{sgn}(f(\boldsymbol{\sigma}))$ . This has the advantage that the input data can be given in terms of critical stresses marking the onset of plastic yielding. Each of these stresses can then simply be scaled proportionally into the elastic or plastic region of the stress space during the training procedure. Furthermore, once the categorial yield function  $f_{ML}(\boldsymbol{\sigma})$  is known, the value of the true yield function can be reconstructed by calculating the distance of the given stress tensor to the yield locus in stress space.

The support vector method aims to construct a classifier in form

$$f_{ML}(\boldsymbol{\sigma}) = \left[ \sum_{k=1}^{N_{SV}} \alpha_k y_k \psi(\boldsymbol{\sigma}, \boldsymbol{\sigma}_k) + b \right] \quad (12)$$

where  $N_{SV}$  is the number of the support vectors  $\boldsymbol{\sigma}_k$  and  $\alpha_k y_k$  are the so-called dual coefficients and  $b$  is an offset. These parameters are defined during the training procedure of the SVC. For nonlinear problems  $\psi(\boldsymbol{\sigma}, \boldsymbol{\sigma}_k)$  should be chosen as the radial basis function (RBF) kernel, which is defined as (Suykens and Vandewalle, 1999).

$$\psi(\boldsymbol{\sigma}, \boldsymbol{\sigma}_k) = \exp[-\gamma \|\boldsymbol{\sigma} - \boldsymbol{\sigma}_k\|^2] \quad (13)$$

Since the ML yield function is defined as convolution sum over support vectors, the gradient to the SVC decision function can be calculated as

$$\frac{\partial f_{ML}(\boldsymbol{\sigma})}{\partial \boldsymbol{\sigma}} = \left[ \sum_{k=1}^{N_{SV}} -2\gamma \alpha_k y_k \exp(-\gamma \|\boldsymbol{\sigma} - \boldsymbol{\sigma}_k\|^2) (\boldsymbol{\sigma} - \boldsymbol{\sigma}_k) \right] \quad (14)$$

When using the RBF kernel for training,  $\gamma$  and  $C$  are the two most important hyperparameters to be defined.  $\gamma$  indicates the width of the kernel function and, thus, the extent to which a single training point has an impact. A smaller value of  $\gamma$  leads to a short-ranged influence. Any misclassified data point is penalized by parameter  $C$ . When  $C$  is small, the penalty for misclassified points is also small such that a wide-margin decision function on the boundary is chosen at the expense of a larger number of misclassifications. If  $C$  is large, the training algorithm limits the number of misclassified cases by using a high penalty and a smaller decision boundary. Thus, a higher value of  $C$  produces a “softer” boundary of the classifier, i.e., a function with more

undulations and an irregular gradient, whereas a smaller value of  $C$  results in a “stiffer” classifier function, i.e., in a rather straight boundary, which might, however, be less accurate.

A high training quality relies on providing sufficient training points on the yield locus, as the SVM algorithm creates support vectors only in the areas covered by training points. While in previous work (Hartmaier, 2020) only the three-dimensional space of principal stresses was considered, where the coverage of the yield locus with data points is a rather trivial task, it is quite a challenge to create data points efficiently in the full 6D stress space. An optimal strategy to create as few as possible data points representing the yield locus in the best possible way is necessary, particularly when data points are created with numerically expensive methods such as CPFEM.

## Data-Generation Model

Creating a set of yield stresses in the full stress space that serves as the ground truth is required to train the ML yield function. In a first step, unit stresses need to be generated that define the load cases, i.e., the directions in which the load is applied. Then, each unit stress is proportionally increased until the value of the yield function  $f(\sigma)$  defined in Eq. 2 is zero, indicating the start of plastic yielding for the stress tensor  $\sigma$ .

Finally, the full set of stress tensors at the onset of plastic yielding represents the ground truth for the training of the ML yield function (Hartmaier, 2020). The task of finding the zeros of the yield function, i.e., the critical stress tensor at the onset of plastic yielding, for each load case needs to be done by a fundamental method, such as CPFEM or mechanical testing that captures the yielding behavior of the material under investigation. Hence, the effort for funding this ground truth is a considerable task that represents the major effort for training an ML flow rule. Thus, finding an optimal strategy for creating the unit stresses is the primary goal of this work. The starting point is creating different distributions of unit stresses on the surface of a unit sphere in a full 6D stress space. After this, the quality of the ML yield criterion resulting from the training with these data points is verified to identify the optimal strategy for data generation.

There are different methods for distributing points on the surface of a unit sphere, but not all of them can be extended to dimensions higher than 3. Different algorithms have been proposed to solve the problem of distributing points uniformly over the surface of a unit sphere in higher dimensional spaces because it is a common but highly non-trivial task with many applications in science and engineering. In many proposed solutions, the initial idea is to uniformly distribute points over a rectangular area that is then mapped to a sphere using the cylindrical projection (Hannay and Nye, 2004). In the next part, four different methods are introduced and the dimensions where they can be used are investigated.

One of the most common and simple methods for distributing points on a 3-dimensional sphere is based on the Fibonacci lattice. Based on the work of Marques et al. (2013), Fibonacci lattice points are constructed through a mapping process from the unit square to the unit sphere. A Fibonacci lattice in the unit square is a set  $Q_m$  of  $F_m$  points  $(x, y)$  defined as

$$\begin{aligned} x_j &= \left\{ j \frac{F_{m-1}}{F_m} \right\} \\ y_j &= \frac{j}{F_m} \\ 0 &\leq j < F_m \end{aligned} \quad (15)$$

where  $F_m$  and  $F_{m-1}$  are the two last members of the Fibonacci sequence for a given  $m > 1$ , as defined by the recurrence equation  $F_m = F_{m-1} + F_{m-2}$ , with the starting numbers  $F_0 = 0$  and  $F_1 = 1$ . In this equation,  $\{x\} = x - [x]$  is the fractional part for non-negative real numbers  $x$ , where  $[x]$  is the integer part of  $x$  (Marques et al., 2013). Mapping this lattice to the unit sphere is done based on the Lambert mapping or equal area north pole projection which is visualized in Figure 1. In this projection, the points

$$\begin{aligned} \theta_j &= \arcsin(2j/2F_m + 1) \\ \phi_j &= 2\pi \left\{ j \frac{F_{m-1}}{F_m} \right\} \end{aligned} \quad (16)$$

given by the polar angle  $\theta_j$  and the azimuthal angle  $\phi_j$  are first moved to  $(\frac{1}{2}\theta, \phi)$  on the northern hemisphere and then projected perpendicularly on to the equatorial plane. It can be shown that equal areas on the sphere’s surface transform into equal areas on the projection. The south pole becomes the whole circumference (Hannay and Nye, 2004). The resulting equatorial plane is shown in Figure 1 with 378 points using the Fibonacci number  $F_{15} = 377$ .

As  $m$  increases, the Fibonacci ratio  $F_m/F_{m+1}$  approaches the golden ratio  $\varphi = (1 + \sqrt{5})/2$  and, as a result, the asymptotic azimuthal angle.

$$\lim_{m \rightarrow \infty} \phi_j = 2j\pi\varphi^{-1} \quad (17)$$

is obtained due to the periodicity of the spherical coordinates (Marques et al., 2013). This relation can be exploited to release the requirement that the number of points has to be a Fibonacci number by defining the coordinates of a spherical point set with an arbitrary, but sufficiently large number  $N$  of points as:

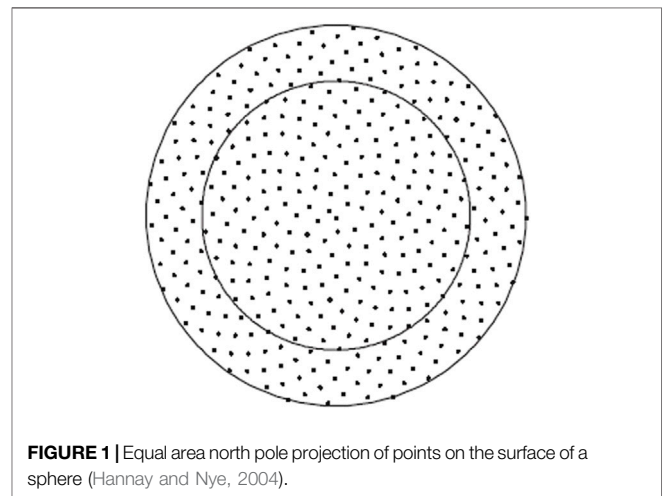


FIGURE 1 | Equal area north pole projection of points on the surface of a sphere (Hannay and Nye, 2004).



$$\begin{aligned} \theta_j &= \arcsin(2j/2N + 1) \\ \phi_j &= 2j\pi\varphi^{-1} \\ 0 &\leq j < N \end{aligned} \tag{18}$$

Based on these angles, the cartesian coordinates of the set of equally distributed points on the 3D unit sphere can be calculated as

$$(x_j, y_j, z_j) = (\cos \theta_j \sin \phi_j, \sin \theta_j \sin \phi_j, \cos \phi_j) \tag{19}$$

The even arrangement of points divides the sphere into equal-area spherical rings due to the area-preserving property of the Lambert map. In this arrangement, each ring contains a single lattice node (Swinbank and James Purser, 2006; González, 2010; Marques et al., 2013). A major disadvantage of this formulation for finding equally distributed points on a sphere based on Fibonacci lattice points is that it cannot be extended directly to dimensions higher than 3.

To overcome this limitation, Marsaglia (1972) has suggested a method that allows the selection of uniformly distributed points on the surface of a 4D-sphere. This method is conducted by picking two points as  $(x_1, x_2)$  and  $(x_3, x_4)$  and rejecting any points if  $(x_1^2 + x_2^2) \geq 1$  and  $(x_3^2 + x_4^2) \geq 1$ . In this case the points

$$\begin{aligned} x &= x_1 \\ y &= x_2 \\ z &= x_3 \sqrt{\frac{1 - x_1^2 - x_2^2}{x_3^2 + x_4^2}} \\ w &= x_4 \sqrt{\frac{1 - x_1^2 - x_2^2}{x_3^2 + x_4^2}} \end{aligned} \tag{20}$$

have a uniform distribution on the surface of a 4D hypersphere. However, this method does not generalize to higher dimensions like the Fibonacci lattice.

Another method that can be used even in higher dimensions is the spherical code problem. In this method, the main goal is distributing  $N$  points on the unit sphere  $\mathbb{S}^{d-1}$  in a way that the minimal distance between any two points is maximized. Any set of points on the unit sphere is called a spherical code (Nurmela, 1995). In the literature, different solutions for the problem of maximizing the mutual distance between any two points have been suggested based on energy minimization techniques. In the work of Buddenhagen and Kottwitz (2001) up to 90 points were suggested in three dimensions. Nurmela (1995) suggested proper spherical codes up that distribute uniformly on unit spheres up to five-dimensional space. Sloane et al. (2000) collected the most extensive spherical codes in various dimensions. The major restriction with spherical codes is their restriction to specific solutions with fixed coordinates and can only be calculated for a specific number of points in different dimensions. Although there are spherical codes that give the coordinates of a uniform distribution of points on the surface of a 6D unit sphere, the limitation in the available number of points makes them unsuited to select points for the training process.

The most promising method which can be used for distributing points on the  $d$ -dimensional surface of the sphere  $\mathbb{S}^d$  embedded in  $d + 1$  dimensions is an inverse sampling

technique that can generate samples from any distribution (Kroese and Rubinstein, 2012). This method can be used for uniform sampling and relies on repeated random sampling and statistical analysis to compute the result. Based on this theory, a random variable that is uniformly distributed in the range  $(0,1)$  can be used to generate a value of a desired random variable with the given distribution. To sample a function uniformly, the first step is to find the PDF (probability distribution function) of that function and then to compute its cumulative probability distribution function (CDF). As the final step, the inverse function of the CDF must be calculated. This method can generate random points on the surface of a unit sphere, but it can also be implemented in combination with the Fibonacci lattice concept described before to generate uniform points. In this case when a random ensemble is generated instead of taking a random independent point from the hypersphere, Fibonacci-like points can be selected to have uniform even distribution on the  $d$ -dimensional surface of the sphere  $\mathbb{S}^d$ .

For introducing this method, a polar coordinate system is defined by  $(r, \theta^1, \theta^2, \dots, \theta^d)$ , which can be converted to cartesian coordinates  $(x^1, x^2, \dots, x^{d+1})$  with the relations

$$\begin{aligned} x^{d+1} &= r \cos \theta^d \\ x^d &= r \sin \theta^d \cos \theta^{d-1} \\ &\vdots \\ x^1 &= r \sin \theta^d \cos \theta^1 \end{aligned} \tag{21}$$

For calculating the PDF, we assume that the data drawn from a particular distribution are independent and identically distributed. If we consider a vector  $\theta$ , containing all angles, as the parameter vector for  $\rho$  which is the probability density function, the PDF is denoted as  $\rho_\theta$  (Raychaudhuri, 2008). Since the points are on  $\mathbb{S}^d$  and  $r = 1$ , the distribution over  $\theta$  coordinates must introduce a uniform surface measure which can be written as

$$\rho_\theta(\theta) = \rho_{\theta^d}(\theta^d) \rho_{\theta^{d-1}|\theta^d}(\theta^{d-1}) \dots \rho_{\theta^1|\theta^2 \dots \theta^d}(\theta^1) \tag{22}$$

Considering Eqs 21, 22 and based on the assumption that the angles are independently distributed, the  $d$ -dimensional PDF can be written as (Raychaudhuri, 2008).

$$\rho_\theta(\theta) = \prod_{\alpha=1}^d \rho_\alpha(\theta^\alpha) \tag{23}$$

Based on the work of Cai et al. (2013), when the variable  $\alpha$  is fixed, the density function or  $\rho_\alpha$  is given by

$$\begin{aligned} \rho_\alpha(\theta^\alpha) &= \frac{1}{\sqrt{\pi}} \frac{\Gamma(\frac{\alpha+1}{2})}{\Gamma(\frac{\alpha}{2})} \sin^{\alpha-1}(\theta^\alpha) \\ \theta^\alpha &\in [0, \pi] \end{aligned} \tag{24}$$

Once the distribution is known for each  $\theta^\alpha$  the next step is to calculate their respective CDFs and finally find their inverse functions.

If we consider  $X$  as the continuous random variate that we want to generate, it will follow  $\rho(\theta)$  as PDF. The CDF for the

variant  $F$  is continuous and increasing in  $(0,1)$ , which can be seen from its definition

$$Y^\alpha(\theta^\alpha) = \int_0^{\theta^\alpha} \rho_\alpha(u) du \tag{25}$$

which satisfies  $Y^\alpha(0) = 0$  and  $Y^\alpha(\pi) = 1$ .

If  $Y^\alpha$  is a random number generated from a continuous uniform distribution between 0 and 1, then  $X$  is a random number from a distribution with a CDF  $F$ , and can be defined as

$$X = F^{-1}(Y^\alpha) \tag{26}$$

where  $F^{-1}$  is the inverse of the CDF (Raychaudhuri, 2008). As the final step, mapping from  $Y$  hypercube to hypersphere should be done based on the Lambert mapping as discussed before. Now a random point ensemble on the sphere has been established. To make it uniform, instead of taking random independent points from the hypercube of  $Y$ 's, we generate points  $Y(n)$  as

$$\begin{aligned} Y_n^d &= \frac{n}{N+1} \\ Y_n^{\alpha-1} &= \{na_1\} \\ Y_n^{\alpha-2} &= \{na_2\} \\ &\vdots \\ Y_n^1 &= \{na_{d-1}\} \end{aligned} \tag{27}$$

where  $a_i$  can be irrational numbers like the Fibonacci sequence that satisfies  $\frac{a_i}{a_j} \notin \mathbb{Q} \forall i \neq j$ . Using the inverse transform method in combination with the Fibonacci lattice concept, in the case of 4-dimensional space ( $d = 4$ ) the points are generated as

$$\begin{aligned} w &= \cos \theta^1 \\ z &= \sin \theta^1 \cos \theta^2 \\ y &= \sin \theta^1 \sin \theta^2 \cos \theta^3 \\ x &= \sin \theta^1 \sin \theta^2 \sin \theta^3 \end{aligned} \tag{28}$$

And the angles are given by

$$\begin{aligned} F(x) &\equiv x - \frac{1}{2} \sin 2x, \theta_n^1 = F^{-1}\left(\frac{n\pi}{N+1}\right) \\ \theta_n^2 &= \arccos(1 - 2\{n\sqrt{2}\}) \\ \theta_n^3 &= 2\pi\{n\sqrt{3}\} \end{aligned} \tag{29}$$

This method can be used for recursively creating uniformly distributed points in 6- or even higher-dimensional spaces. The points are distributed such that the distance to the nearest neighbors for each point is maximum. Since the points are constructed to lie on the surface of a unit sphere, the distance of each point to the center of the sphere is unity.

### Validation of Uniform Distribution of Data Points

The inverse transform method mentioned in the previous section has been implemented in a python code that is freely available within the Python package ‘‘pyLabFEA’’ (Hartmaier et al., 2022) from a public repository. Using this algorithm,

any desired number of points can be distributed uniformly on the surface of a  $d$ -dimensional sphere. In **Figure 2**, 400 points were distributed uniformly on the surface of a unit sphere in 3D space.

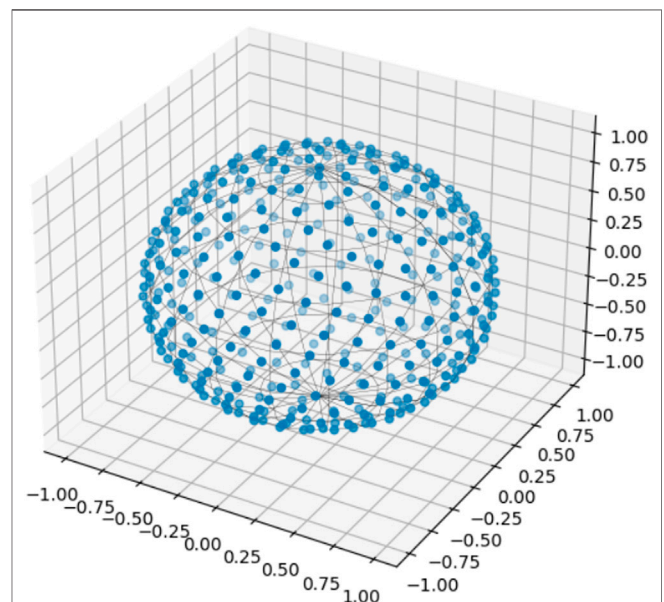
To test the uniformity of this data generation method, 400 points were distributed on the surface of a hypersphere in 6D space. The average distance from each point to its five nearest neighbors was calculated. The distribution of these distances can be seen in **Figure 3A**. For comparison, 400 points were generated randomly, and the same average neighboring distance was calculated and is shown in **Figure 3B**. The random points were generated using the random function available in the NumPy (Harris et al., 2020) package in python.

It is seen that the method laid out here can create any number of data points that are uniformly distributed on a surface of a  $d$ -dimensional unit sphere in the sense that the mutual distance of any two points is maximized and the average distances for  $K$  nearest neighbors are at the same range. This method will be used in the following to create the data points for training an ML yield criterion.

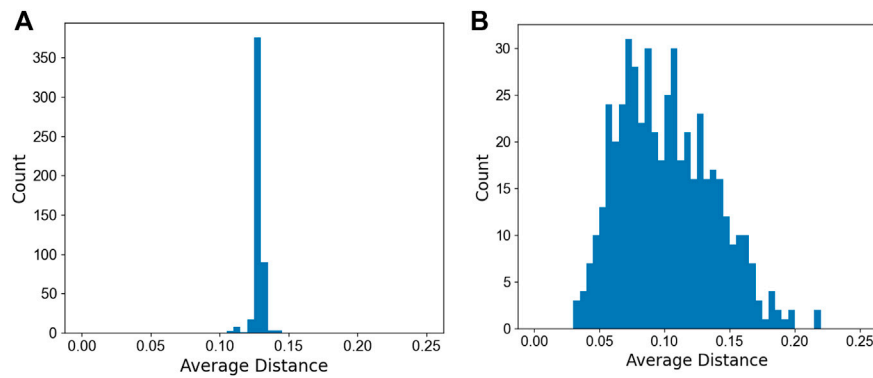
## OPTIMAL STRATEGY FOR DATA GENERATION

In order to find an optimal strategy for selecting a set of training data points, in the following, various data sets will be created with uniform and random distributions and for different sub-spaces of the full stress space. The training results for the different sets are evaluated concerning different error measures, and thus, a strategy to find the smallest possible training data set that provides the desired accuracy of the result is developed.

Since the training data points need to be generated as stress tensors lying on the yield locus of material, we define a reference



**FIGURE 2** | 400 points generated uniformly on the surface of a sphere in 3-dimensional space using the combination of Monte Carlo theory and Fibonacci spiral principle.



**FIGURE 3** | Distribution of average distances for five nearest neighboring points of (A) uniform and (B) randomly distributed points on the surface of the unit sphere in 6D space. The total number of points is 400.

material with Hill-type anisotropy with the parameters given in **Table 1**. Using such a relatively simple material model with a simple yield criterion defined in **Eq. 2**, allows us to generate the training data sets with a minimum effort. This is beneficial for developing an optimal strategy for generating data sets for materials with significant plastic anisotropy. Afterward, it is verified that the training strategy also produces good results for more severe cases of anisotropy as they can be seen in CPFEM results or described by Barlat-type yield criteria to demonstrate the general applicability of the developed method.

The described reference material is used for creating training data for machine learning algorithms. This is accomplished by first creating unit stresses according to various schemes. Then each unit stress is increased proportionally until the yield function of this stress tensor is zero, i.e., when plastic yielding starts for this specific load case. The full set of stress tensors at the onset of plastic yielding represents the yield function in a data-oriented way and, thus, forms the ground truth for the training of the ML yield function. After this, the ML yield function in form of a Support Vector Classifier (SVC) is trained based on this data set.

As will be seen later, it is necessary to create unit stresses that combine load cases in the full 6D stress space and load cases with purely normal stresses, representing a 3D subspace of the full stress space. For training sets with only 6D load cases, the normal stress space is grossly under-represented with yields a very poor training result. This reflects the fact that the Voigt representation of the symmetric ( $3 \times 3$ ) stress tensor as vector of its six independent components does not fully represent the tensorial properties of the stress as, for example, the existence of a transformation into a diagonal tensor by a rotation of the coordinate system. Hence, the combination of 3D and 6D stresses reflects the tensorial properties of the stress tensor better. The optimal combination of these load cases will be investigated in the second step together with the optimization of the size of the training set.

## Uniform Versus Random Training Data Sets

In the first step to develop an optimal training strategy, we compare the results from training with uniformly and

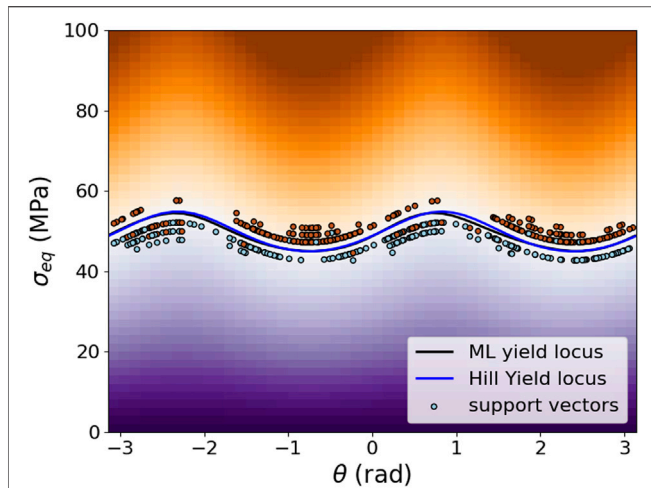
**TABLE 1** | Elastic and plastic material parameters define the reference material with Hill-like anisotropy in plastic flow behavior. For simplicity, ideal plasticity with no work hardening is considered in this work.

Quantity	Symbol	Value
Yield strength	$\sigma_y$	50 MPa
Young's modulus	$E$	200 GPa
Poisson's ratio	$\nu$	0.3
Hill parameters	$H_1, H_2, H_3, H_4, H_5, H_6$	1.4, 1, 0.7, 1.3, 0.8, 1

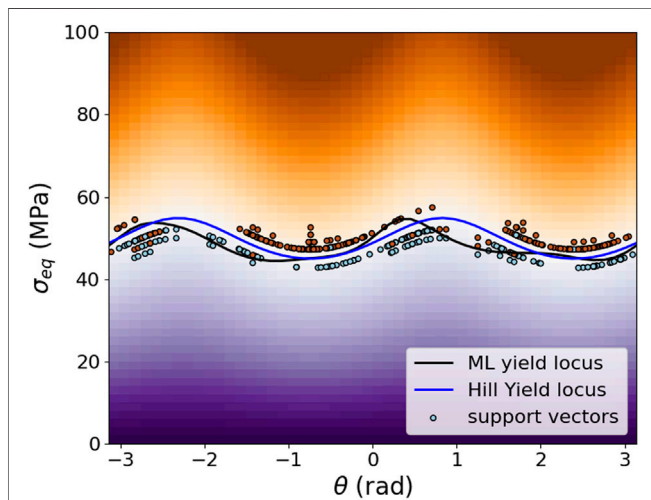
randomly distributed training points. For training, 400 load cases have been generated, including 350 load cases in full stress space and 50 load cases in principal space. Since the success of the training procedure depends critically on the hyperparameters, the optimal hyperparameters for the SVC were selected using a grid search algorithm for each data set.

For uniformly distributed load cases, the result of SVC training with the hyperparameters  $C = 5$  and  $\gamma = 2.5$  is shown in **Figure 4**, where the J2 equivalent stress of the stress tensors at the onset of plastic yielding is plotted over the polar angle of the stress tensor in the  $\pi$ -plane, i.e., the space of principal deviatoric stresses, **Appendix A** for the definition of these quantities. A good agreement between trained ML function (black line) and the Hill yield locus (blue line) can be observed. In this figure, the support vectors identified during the training procedure are also represented with a color indicating their location in the elastic or plastic domain of the stress space. It is seen that all support vectors lie in the vicinity of the yield locus. At first sight, it appears that many support vectors are misclassified, i.e., are lying in the wrong domain. However, this is a mere artifact from projecting the 6D stresses onto the  $\pi$ -plane for a material with a significant plastic anisotropy.

For comparison, the training was done with the same number of load cases, i.e., a total of 400 with 350 in 6D stress space and 50 in normal stress space, that were generated from a random distribution. The optimal hyperparameters for this case were identified as  $C = 12$  and  $\gamma = 2.5$  by a grid search algorithm. The result of the training is shown in **Figure 5**, where it can be seen that the ML yield locus resulting from training data points



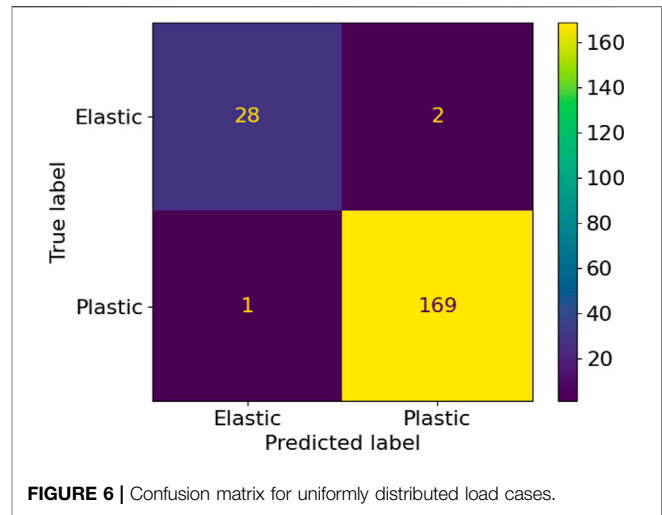
**FIGURE 4 |** The plot of trained SVM classification with uniformly distributed training data in cylindrical coordinates on the  $\pi$ -plane (space of principal deviatoric stresses). Orange colors represent positive yield function values, i.e., they indicate plastic yielding. Purple colors represent negative values of the yield function, where the stress lies in the elastic regime. No color scheme is given because the absolute value of the ML yield function has no physical meaning. The blue line indicates the stresses where the yield function is zero for the reference material with an analytically formulated Hill-like yield criterion and a black line for the ML yield criterion. The support vectors identified during the training procedure are represented as open symbols.



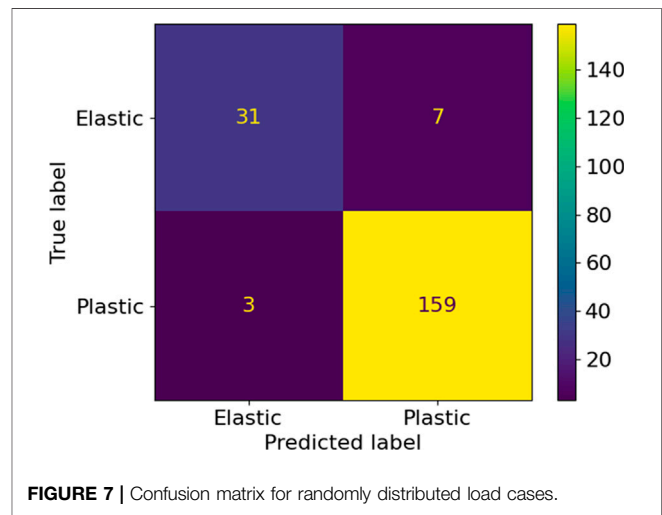
**FIGURE 5 |** The plot of trained SVM classification with randomly selected training data in cylindrical coordinates on the  $\pi$ -plane (space of principal deviatoric stresses). The figure annotations are identical to **Figure 4**.

representing randomly selected load cases does not have a good agreement with that of the reference material.

To further quantify the results of the training procedure, both ML yield functions are verified by comparison with the analytical yield function for random stress tensors in the vicinity of the yield locus. The confusion matrices in **Figures 6, 7** summarize the classification performance of both yield functions concerning the



**FIGURE 6 |** Confusion matrix for uniformly distributed load cases.



**FIGURE 7 |** Confusion matrix for randomly distributed load cases.

same test data in the form of a two-dimensional matrix, indexed in one dimension by the object’s true class and in the other dimension by the class assigned by the classifier. In this context, the four cells of the matrix are designated as true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) (Shultz et al., 2011).

The classification performance can further be quantified in terms of four classification results, as

$$\begin{aligned}
 \text{Recall} &= TP / (TP + FN) \\
 \text{Precision} &= TP / (TP + FP) \\
 \text{Accuracy} &= (TP + TN) / (TP + FP + FN + TN) \\
 \text{F1 Score} &= 2 * \text{Precision} * \text{Recall} / (\text{Precision} + \text{Recall})
 \end{aligned}
 \tag{30}$$

The summary of metrics for uniform and randomly distributed load cases are given in **Table 2**. It is concluded that uniformly distributed training data results in a significantly higher quality of the training of the ML yield function. Hence, in the remainder of this work, only this method will be further investigated.



**TABLE 2** | Summary of the metrics for uniform and random load cases indicate the quality of training.

Metrics	Uniform	Random
Precision	0.9885	0.9578
Accuracy	0.985	0.95
Recall	0.9942	0.9815
F1 Score	0.9913	0.9695

## Optimal Number and Structure of Load Cases

After finding the proper strategy for creating unit stresses uniformly distributed in the stress space, the next step is to find the optimal size for it. Besides the quality, the quantity of training data plays a vital role in the performance of any machine learning model. Small training data sets result in low training accuracy and lack of precision, and a very big training size may lead to overfitting and the problem that the model cannot generalize well to new data. In this context, it is also important to find the optimal ratio of unit stresses in the full 6D stress space and purely normal stresses in the 3D sub-space.

In the first step for finding the proper ratio between 6D and 3D load cases, training was done for a fixed number of 300 load cases and different ratios of 1:2, 1:1, 1.5:1, 2:1, 2.5:1, and 3:2. and the confusion matrix and the training metrics were compared in each case. The corresponding plots can be seen in **Figure 8**. After training, the precision and mean absolute error were calculated and compared in different ratios of unit load cases in full stress space and the extra cases at principal stress space, the ratio of 2:1 for 6D:3D load cases have highest precision and the lowest mean absolute error (MAE). Mean Absolute error measures the average magnitude of error which quantifies the difference between prediction and the actual observation which can be defined as:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - x_i| \quad (31)$$

At the next step for finding the optimal training size, the training was done for different total numbers of uniformly distributed load cases with a fixed ratio of 2:1 for 6D to 3D load cases. The comparison is again made based on the precision calculated from the confusion matrix and the MAE, **Figure 9**. It is seen that a number of 300 is the optimal size for the set of training data with the lowest mean absolute error and a high precision after training. It can be seen that an increasing number of data points leads to an increasing MAE, possibly due to overfitting. Furthermore, the precision does not have a uniform trend with an increasing number of data points such that we conclude that 300 data points is the optimal value, which still allows an efficient data generation process.

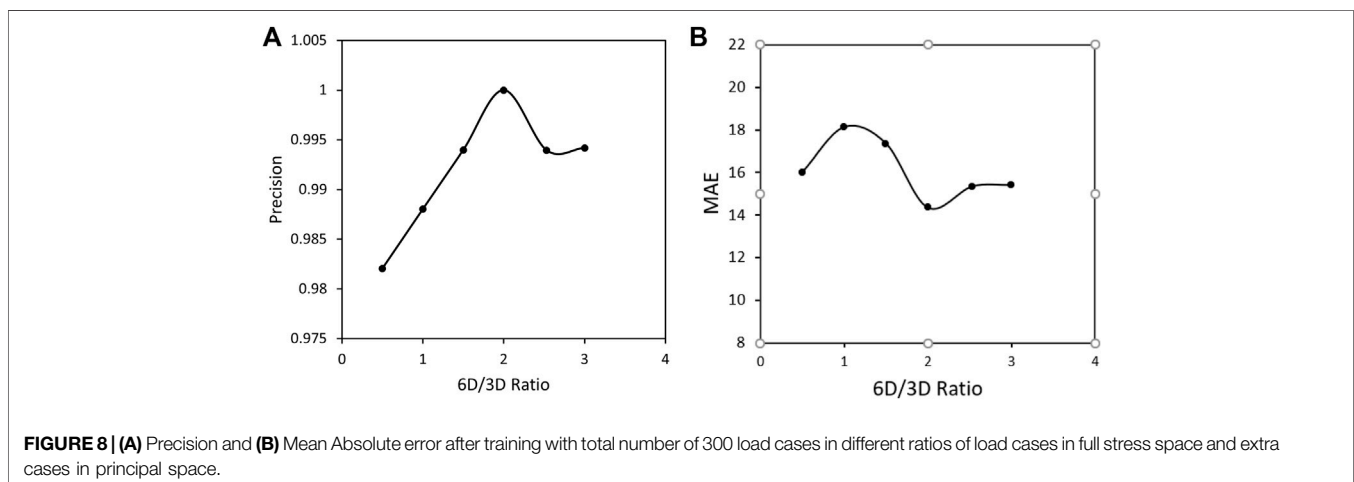
The resulting ML yield function, trained with the optimized training set of 300 uniformly distributed load cases, is visualized in **Figure 10**. The optimal hyperparameters for the SVC algorithm have been determined as  $C = 15$  and  $\gamma = 3$  using grid search. A comparison to **Figure 4**, where a total of 400 data points have been used in the training set, clearly reveals the importance of the optimization of the ratio between 6D and 3D stresses in the training data, as the optimized result is more accurate with 25% less training data.

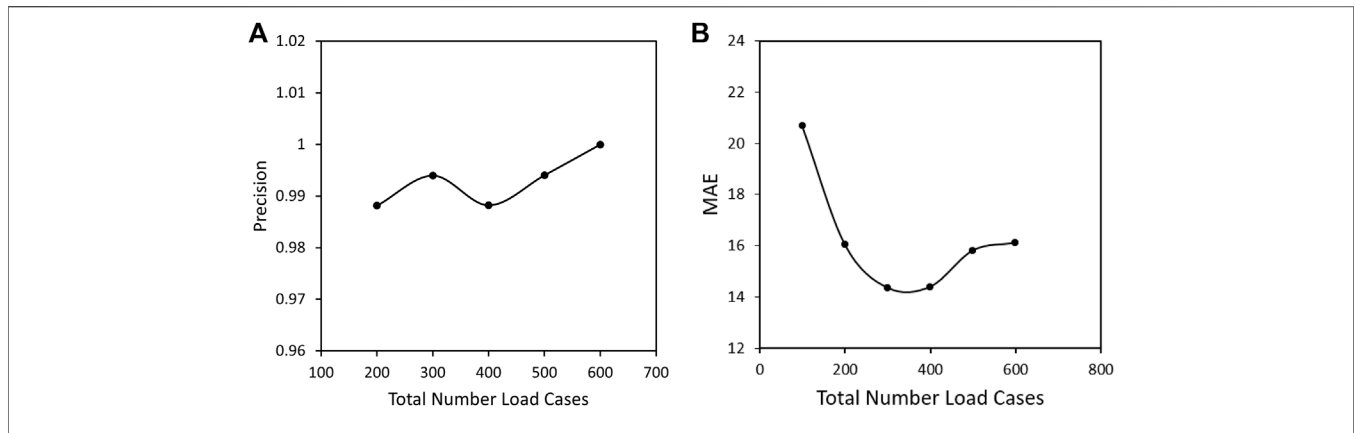
The quality of the ML flow rule training with the optimized data set is further verified by comparing the results of the ML yield function to the known reference values for random stresses in the vicinity of the yield locus. The confusion matrix and the performance metrics were calculated and shown in **Figure 11**, and **Table 3**. Comparing to **Figures 6, 7** and **Table 2** again indicates a better performance after optimization.

## APPLICATION OF TRAINED MACHINE LEARNING YIELD FUNCTIONS IN FINITE ELEMENT ANALYSIS

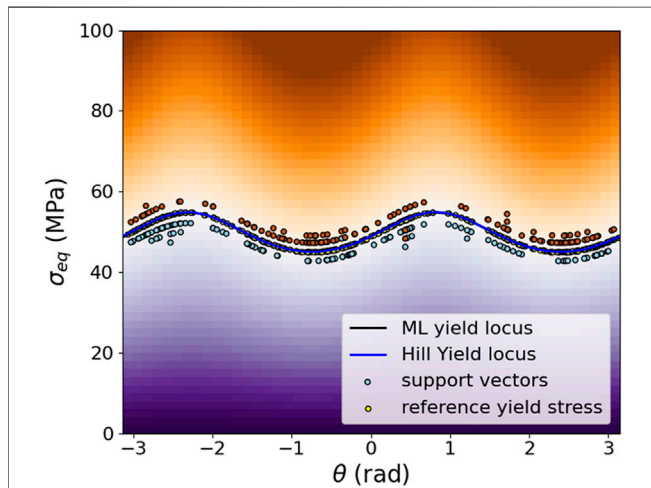
### Hill-Type Anisotropy

To demonstrate the capabilities and the accuracy of the trained ML yield function, in the first step a finite element





**FIGURE 9 | (A)** Precision and **(B)** Mean Absolute error after training with different number of load cases and a fixed ratio of 2:1 for 6D to 3D load cases in full stress space and extra cases in principal space.



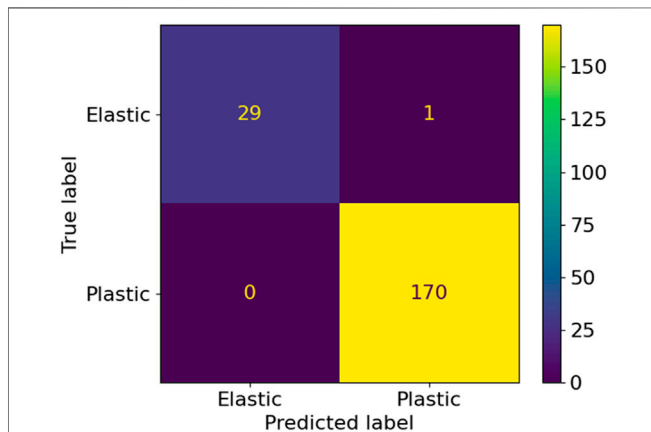
**FIGURE 10 |** Plot of trained SVM classification with optimized training size and ratio in cylindrical coordinates on the  $\pi$ -plane (space of principal deviatoric stresses). The figure annotations are identical to **Figure 5**.

**TABLE 3 |** Summary of the metrics for 300 with the ratio of 2 between 3D and 6D load cases.

Metrics	Uniform
Precision	0.994
Accuracy	0.995
Recall	1
F1 Score	0.997

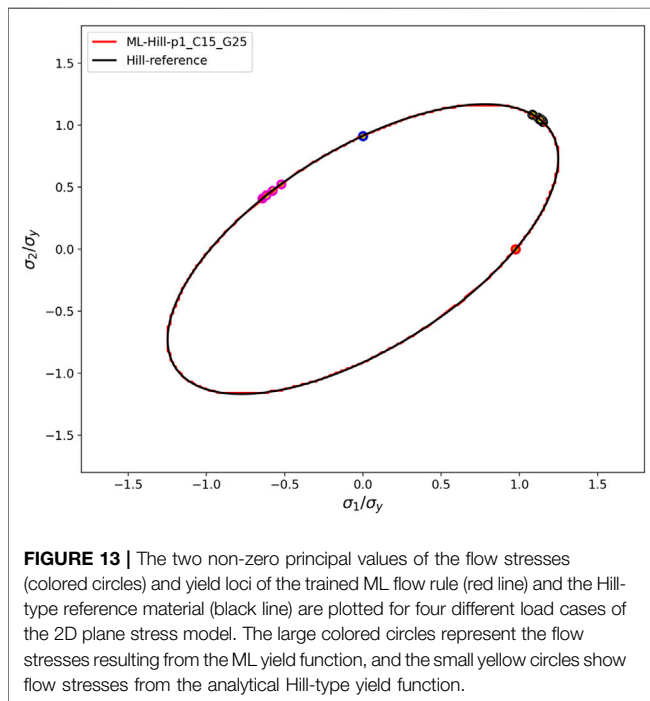
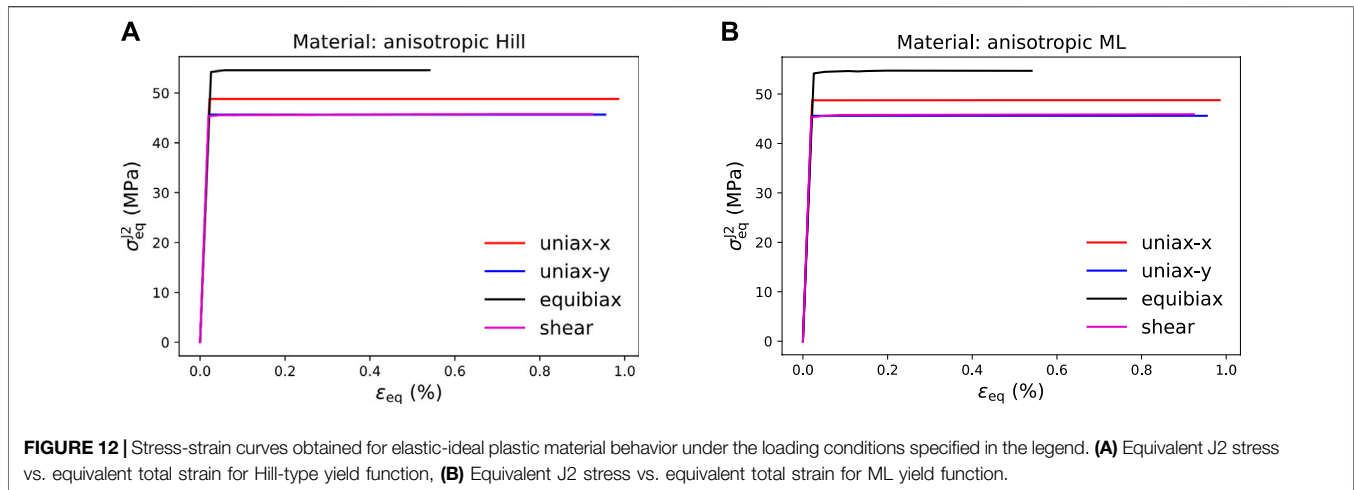
**TABLE 4 |** Yield stress obtained for a Hill-type reference material with analytical yield function and a material with an ML yield function trained to data from the reference material under four specified load cases.

Load case	Hill yield stress (MPa)	ML yield stress (MPa)	Rel. difference (%)
Uniaxial stress, horizontal	48.744	48.795	0.10
Uniaxial stress, vertical	45.596	45.644	0.11
Equibiaxial strain	54.171	54.233	0.11
Pure shear strain	45.283	45.33	0.10

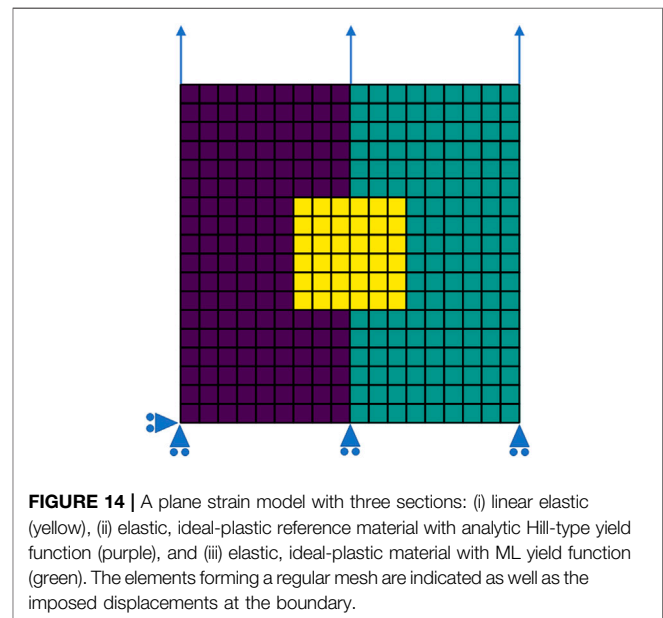


**FIGURE 11 |** Confusion matrix for 300 load cases and the ratio of 2:1 between 6D and 3D load cases.

analysis (FEA) of a simple 4-element 2D plane-stress model under four load cases has been performed: 1) uniaxial stress in the vertical direction, 2) uniaxial stress in the horizontal direction, 3) equibiaxial strain, and 3) pure shear strain. The numerical simulations shown in this work have been completed with the open-source package pyLabFEA (Hartmaier et al., 2022). All examples of this work are also provided as python code and a Jupyter notebook in this public repository. The return mapping algorithm used to calculate the plastic strain increments based on the ML yield function and its gradient, defined in Eqs 12, 14, respectively, has been described in detail in (Suykens and Vandewalle, 1999). In that work, also the full details of the finite element model to evaluate the given load cases are provided. To estimate the accuracy of the ML yield function, its results are compared to

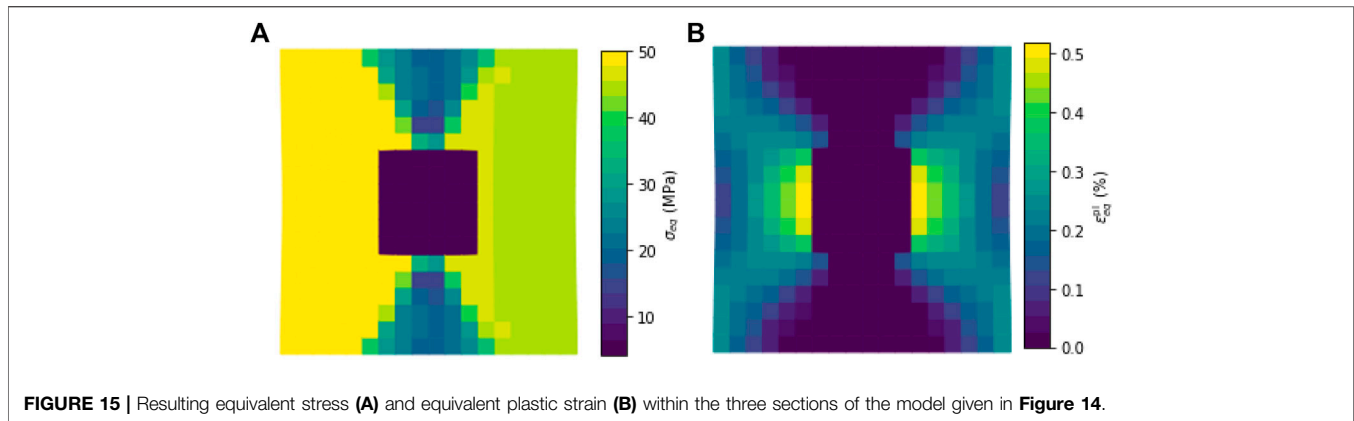


the ones obtained from the same model with a Hill-type reference material with elastic-ideal plastic behavior and an analytically defined yield function, Eq. 2. The material parameters for this reference material are given in Table 1. As described above, the ML yield function has been trained to a data set of 300 stress tensors lying on the yield locus of this reference material. The equivalent yield stresses resulting from the FEA under the four given load cases are summarized in Table 4, where it is seen that the error of the ML yield function is only on the order of 0.1%. The resulting curves for equivalent stress versus equivalent strain for each material under the different load cases are plotted in Figure 12, where the elastic-ideal plastic material behavior of both materials is verified for total equivalent



strains of up to 1%. Note that the equivalent J2 stress is plotted for both materials because the Hill-type equivalent stress defined in Eq. 3 depends on material parameters that are typically unknown when working with data-based yield functions.

During the plastic deformation of materials with ideal plasticity, it is expected that the flow stresses remain on the yield locus, as the material does not support higher equivalent stresses. In Figure 13, the flow stresses for Hill and ML cases are plotted in the space of the non-zero principal stresses of the plane stress model, together with the yield locus in this slice of the stress space. It is seen that all flow stresses, in fact, remain on the yield locus. Furthermore, it is seen that under stress-controlled boundary conditions, i.e., uniaxial stress in the horizontal or vertical direction, the flow stresses remain constant, whereas, under strain-controlled boundary conditions, the response stress of the anisotropic material is subject to changes. A comparison of



**FIGURE 15** | Resulting equivalent stress **(A)** and equivalent plastic strain **(B)** within the three sections of the model given in **Figure 14**.

this evolution of the flow stress tensor under strain-controlled deformation for the analytical Hill-type yield function and the ML yield function exhibits that both solutions are in very good agreement, demonstrating the accuracy and numerical stability of the ML flow rule even under finite plastic strains.

In a second step, the ML yield function is tested under a more complex state of deformation. This is accomplished by performing the FEA of a plane-strain model under uniaxial strain with laterally free boundaries. The three sections of the model are given by a square-shaped elastic inclusion in the middle of a matrix that is vertically split up into the reference material and the ML material, **Figure 14**. As the ML material is trained to have identical material properties as the reference material, the matrix is expected to show a mirror-symmetric state of deformation. The linear-elastic inclusion is rather compliant, with Young's modulus of  $E_{inc} = 1$  GPa and a Poisson ratio of  $\nu_{inc} = 0.27$ . The boundary conditions at the top surface are such that a total strain of 0.2% in the vertical direction is applied at the end of the load step.

**Figure 14A** shows the resulting equivalent stress for each finite element. It is apparent that qualitatively a symmetrical stress state in the different materials is reached. However, quantitatively, the equivalent stresses differ because in the region with the Hill-type material (left-hand-side), the equivalent stress as defined in **Eq. 3** is plotted, whereas in the region with the ML material (right-hand-side) the J2 equivalent stress is plotted, which can also be calculated from **Eq. 3** when the Hill parameters are set  $H_1 = H_2 = \dots = H_6 = 1$ . Note that for a data-based constitutive model, properties like the Hill parameters are typically not known and also not necessary for the training

process. Hence, it is best to use a material independent definition for the equivalent stress, as the J2 equivalent stress. In **Figure 15B**, the equivalent plastic strains are plotted, demonstrating a completely symmetric deformation between both regions, as expected for materials with identical plastic properties. Thus, this example verifies that the ML yield function can, in fact, be trained to possess the same plastic properties as the Hill-type reference material and produce the same plastic strain increments even in complex loading situations.

## Barlat-Type Anisotropy

After verifying the accuracy and robustness of the ML yield function for cases of Hill-type plastic anisotropy, it is tested here under a more demanding kind of anisotropic behavior, given by a material with a Barlat-type yield function (Yld 2004-18p) (Barlat et al., 2005). The material parameters given in **Table 5** have been chosen to mimic a polycrystal with a strong Goss texture, which represents a rather severe case of anisotropic yielding behavior.

Following the workflow developed above, in the first step, a set of 300 stress tensors on the yield locus of the Barlat-type reference model is generated. These training stresses are again produced by generating 200 unit stresses as load cases that are uniformly distributed in the 6D stress space and 100 load cases uniformly distributed in the 3D sub-space of normal stresses. Then, these unit stresses are increased proportionally until the zero of the Barlat yield function Yld 2004-18p (Barlat et al., 2005) is reached. The python code for this example is also provided in the pyLabFEA package (Hartmaier et al., 2022). With this training data set, representing the yield locus of the Barlat-type reference

**TABLE 5** | Material parameters for the Barlat-type reference material, mimicking the plastic anisotropy of a Goss-textured polycrystal.

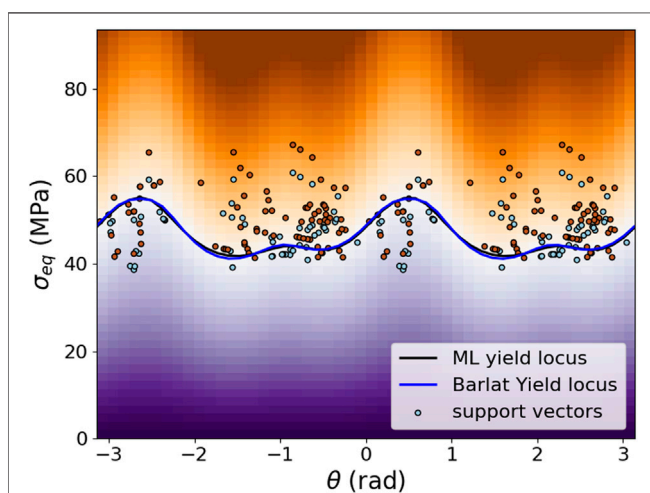
Quantity	Symbol	Value
Yield strength	$\sigma_y$	46.76 MPa
Young's modulus	$E$	151.22 GPa
Poisson's ratio	$\nu$	0.3
Barlat parameters	$B_1, B_2, \dots, B_6$	0.818, -0.364, 0.312, 0.843, -0.018, 0.832
	$B_7, B_8, \dots, B_{12}$	0.360, 0.081, 1.293, 1.096, 0.909, 0.277
	$B_{13}, B_{14}, \dots, B_{18}$	1.090, 1.183, -0.019, 0.905, 1.883, 0.013



material, the ML yield function is trained with the same hyperparameters as given above, i.e.,  $C = 15$  and  $\gamma = 2.5$ . The training result is shown in **Figure 16** by projecting yield stresses and support vectors to the  $\pi$ -plane, i.e., the plane of principal deviatoric stresses, **Appendix A** for a definition of the plotted quantities. It is seen that even though the Barlat-type yield function is much more irregular than the Hill-type yield function, the trained ML yield function provides a very accurate representation of its yield locus.

The quality of the ML yield function training is further quantified by comparing the signs of the values of the analytical yield function and the ML yield function for 200 random stress tensors in the vicinity of the yield locus. The corresponding confusion matrix is plotted in **Figure 17**, and the summary of the metrics indicating the high quality of the training result is summarized in **Table 6**. This analysis confirms that the ML yield function describes the Barlat-type reference material with very high accuracy.

With this trained ML yield function, the same FEA cases as above have been performed to demonstrate the stability of the ML yield function even for a more severe plastic anisotropy. After training, the stress-strain curves are plotted as J2 equivalent stress vs. equivalent total strain in four load **Figure 18A** cases as shown in **Figure 18A**. Also, the flow stresses resulting from FEA are plotted in **Figure 18B**, and they all lie on the ML yield locus as expected for ideal plasticity. Furthermore, it is seen again that for the strain-controlled boundary conditions, the stress tensors change during the plastic deformation. This behavior has already been observed for the Hill-type plastic anisotropy, but it is even more pronounced in this case. Note that for the case of equibiaxial strain, the stresses evolve into a “corner” of the yield locus, which causes a significant increase in the equivalent stress that is also seen in the corresponding stress strain curve for this load case.



**FIGURE 16** | Plot of trained ML yield function (black line), the analytic Barlat yield function (blue line) and the support vectors (circles) in cylindrical coordinates on the  $\pi$ -plane. The same color scheme as in **Figure 5** has been applied.

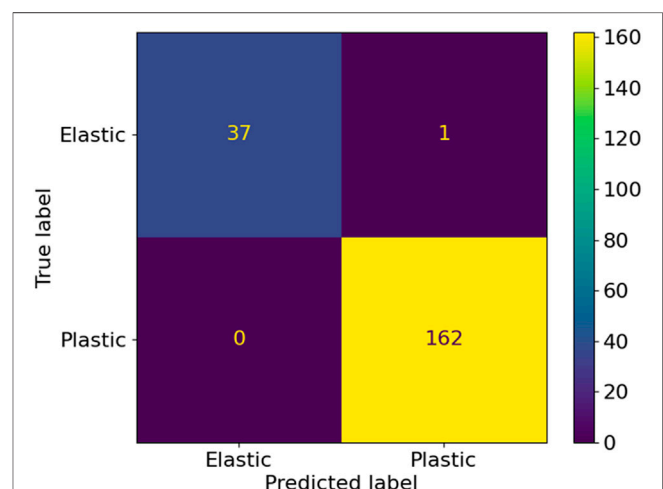
**TABLE 6** | Summary of the metrics for 300 with a ratio of 2 between 3D and 6D load cases for Trained ML yield function.

Metrics	Uniform
Precision	0.9939
Accuracy	0.995
Recall	1
F1 Score	0.9969

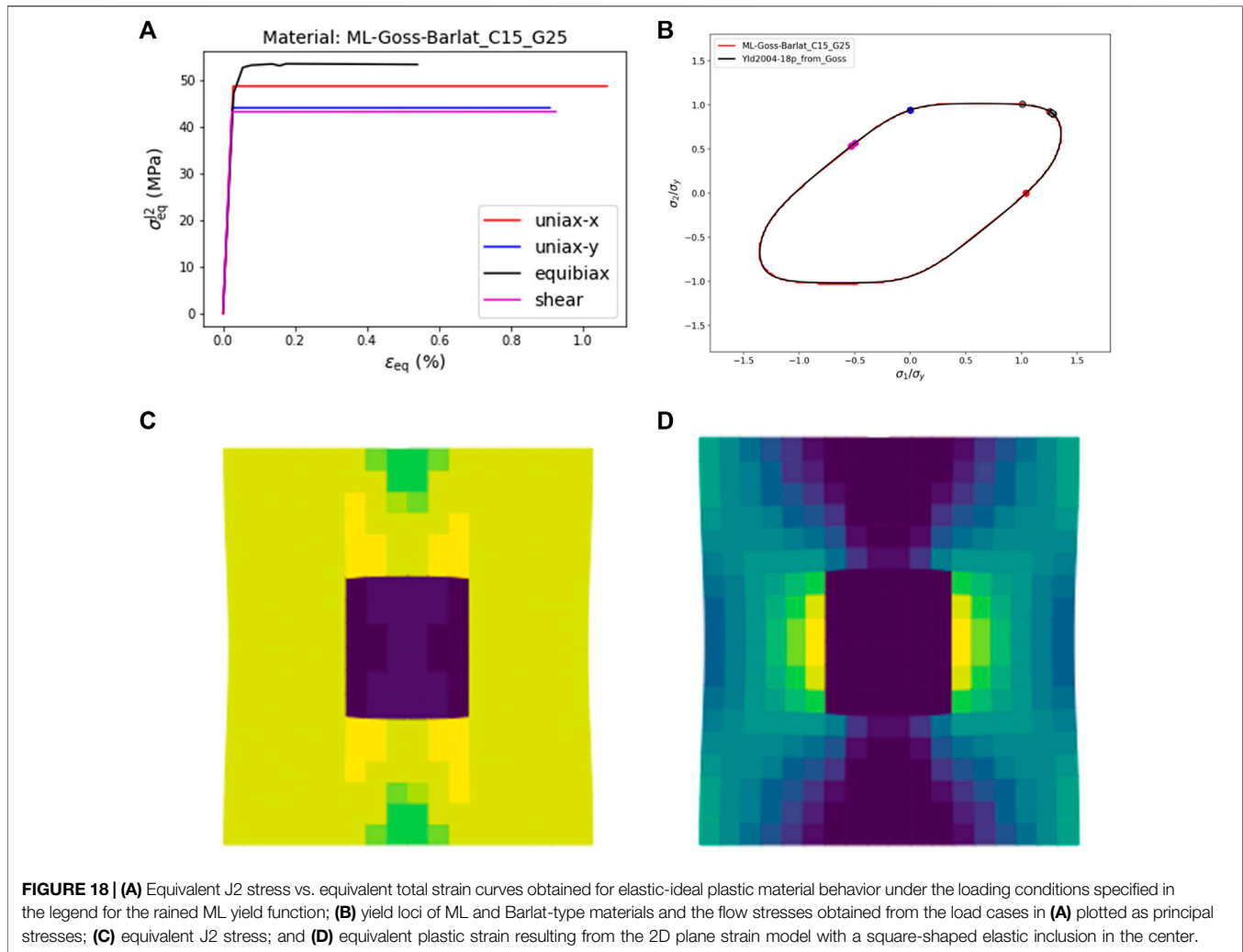
Furthermore, in **Figure 18**, the equivalent stress **Figure 18C** and the equivalent plastic strain **Figure 18D** for a model with a square-shaped elastic inclusion and an elastic-ideal plastic matrix represented by the ML yield function are illustrated to demonstrate the numerical stability of the ML flow rule even in cases of heterogeneous deformation patterns.

## CONCLUSION

In this work, an optimized procedure to generate a data-based description of the yield function of an arbitrary material has been developed. Conventionally, the yield function is based on the concept of the equivalent stress and indicates whether the material response to an applied stress tensor results in linear elastic material response, indicated by a negative value of the yield function, or rather in plastic yielding of the material when the yield function is zero or positive. Mathematically, the zeros of the yield function constitute a hypersurface in stress space, the so-called yield locus, that separates elastic and plastic domains. Since the stress space is 6-dimensional and spanned by the six independent normal and shear components of the stress tensor, it is essential to find an optimal way for sampling this hypersurface with as few as possible data points. Each data point represents a stress tensor at which the material starts to yield plastically, and generating such data requires either numerically



**FIGURE 17** | Confusion matrix comparing the trained ML yield function and the reference Barlat-type yield function for 200 random stress tensors in the vicinity of the yield locus.



expensive approaches such as crystal plasticity finite element methods or laborious mechanical testing under multi-axial loading conditions or a combination of both.

In a first step, a numerical method has been introduced that allows a uniform sampling of the yield locus in a six-dimensional stress space. The main idea behind this method is to proportionally increase uniformly distributed unit stresses until the criterion for plastic yielding is reached for each loading direction. Finding a uniform distribution of arbitrarily many points on the surface of a unit sphere in more than three dimensions is, however, a highly non-trivial task. The solution suggested in this work combines ideas of Monte Carlo sampling and choosing regularly distributed sampling points in higher-dimensional spaces based on the Fibonacci sequence. To sample a higher dimensional unit sphere uniformly, the probability distribution function of its surface needs to be defined in Cartesian coordinates, and then the cumulative distribution function and its inverse need to be computed. Based on this inverse function, a mapping algorithm is defined by which random numbers from the unit interval can be distributed uniformly on the hypersphere. If Fibonacci-like points are mapped accordingly, instead of random numbers, it can be

shown that the mutual distance between any two sampling points on the surface of the hypersphere is rather constant and maximized compared to randomly distributed points. It is demonstrated in this work that such a uniform distribution of stress tensors in the 6-dimensional stress space is superior to purely random sampling.

In the next step, this data-oriented description of the yield locus is used as the basis for training of a support vector classifier (SVC) that takes an arbitrary stress tensor as input and predicts whether the material response to this stress is elastic or plastic. In earlier work (Hartmaier, 2020), it has been shown for purely normal stresses how plasticity in the framework of finite element analysis (FEA) can be described based on such trained SVC. Here, this formulation is generalized to the full 6-dimensional stress space, and it is demonstrated that SVC can be trained to predict the behavior of classical yield functions, like Hill or Barlat-type yield functions, with very high accuracy, even for a very significant plastic anisotropy.

As for any machine learning (ML) algorithm, providing high-quality training data is of great importance, as training with smaller sets of proper data will result in a better training

success than larger sets of poor data. After finding a suitable way to distribute training data uniformly in stress space, it was found that an appropriate ratio of normal stresses and stresses in the full 6-dimensional stress space is necessary because otherwise, the normal stresses are under-represented. It is demonstrated that a ratio of 1:2 for normal stresses and full stresses represents the tensorial properties of the yield stress in the best way. It is also concluded that 300 data points in the form of stress tensors on the yield locus are sufficient to train the ML yield function with high accuracy, even in cases of severe plastic anisotropy. The thus-trained ML yield functions have been shown to produce accurate and stable numerical solutions in FEA.

To further expand the applicability of ML yield functions in FEA, microstructural parameters like crystallographic texture or grain size and morphology can be included in the input data for the training of the ML yield function, besides the purely mechanical data used in this work. Furthermore, it is crucial to develop a proper data-oriented formulation of work hardening and history-dependent material behavior in the next step.

## REFERENCES

- Banabic, D., Kuwabara, T., Balan, T., and Comsa, D. S. (2004). An Anisotropic Yield Criterion for Sheet Metals. *J. Mater. Process. Technol.* 157–158, 462–465. doi:10.1016/j.jmatprotec.2004.07.106
- Barlat, F., Aretz, H., Yoon, J. W., Karabin, M. E., Brem, J. C., and Dick, R. E. (2005). Linear Transformation-Based Anisotropic Yield Functions. *Int. J. Plasticity* 21, 1009–1039. doi:10.1016/j.ijplas.2004.06.004
- Bonet, J., and Wood, R. D. (1997). *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge: Cambridge University Press. Available at: <https://books.google.de/books?id=ORMLdrq1f18C>.
- Buddenhagen, J., and Kottwitz, D. A. (2001). Multiplicity and Symmetry Breaking in (Conjectured) Densest Packings of Congruent Circles on a Sphere. *Preprint*. Available at: [https://www.researchgate.net/publication/248871642\\_Multiplicity\\_and\\_Symmetry\\_Breaking\\_in\\_Conjectured\\_Densest\\_Packings\\_of\\_Congruent\\_Circles\\_on\\_a\\_Sphere](https://www.researchgate.net/publication/248871642_Multiplicity_and_Symmetry_Breaking_in_Conjectured_Densest_Packings_of_Congruent_Circles_on_a_Sphere).
- Cai, T., Fan, J., and Jiang, T. (2013). Distributions of Angles in Random Packing on Spheres. *J. Mach. Learn. Res.* 14, 1837–1864.
- Cazacu, O., and Barlat, F. (2001). Generalization of Drucker's Yield Criterion to Orthotropy. *Maths. Mech. Sol.* 6, 613–630. doi:10.1177/108128650100600603
- Chen, W. F., and Saleeb, A. F. (1994). *Constitutive Equations for Engineering Materials*. Amsterdam, Netherlands: Elsevier. Available at: <https://books.google.de/books?id=P91IAQAIAAJ>.
- Chinesta, F., Ladeveze, P., Ibanez, R., Aguado, J. V., Abisset-Chavanne, E., and Cueto, E. (2017). Data-driven Computational Plasticity. *Proced. Eng.* 207, 209–214. doi:10.1016/j.proeng.2017.10.763
- Eggersmann, R., Kirchdoerfer, T., Reese, S., Stainier, L., and Ortiz, M. (2019). Model-free Data-Driven Inelasticity. *Comput. Methods Appl. Mech. Eng.* 350, 81–99. doi:10.1016/j.cma.2019.02.016
- Eggersmann, R., Stainier, L., Ortiz, M., and Reese, S. (2021). Model-free Data-Driven Computational Mechanics Enhanced by Tensor Voting. *Comput. Methods Appl. Mech. Eng.* 373, 113499. doi:10.1016/j.cma.2020.113499
- González, Á. (2010). Measurement of Areas on a Sphere Using Fibonacci and Latitude–Longitude Lattices. *Math. Geosci.* 42, 49–64.
- Hannay, J. H., and Nye, J. F. (2004). Fibonacci Numerical Integration on a Sphere. *J. Phys. A: Math. Gen.* 37, 11591–11601. doi:10.1088/0305-4470/37/48/005
- Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., et al. (2020). Array Programming with NumPy. *Nature* 585, 357–362. doi:10.1038/s41586-020-2649-2
- Hartmaier, A. (2020). Data-oriented Constitutive Modeling of Plasticity in Metals. *Materials* 13, 1600. doi:10.3390/ma13071600
- Hartmaier, A., Menon, S., and Shoghi, R. (2022). *Python Laboratory for Finite Element Analysis (PyLabFEA)*. doi:10.5281/zenodo.5913365
- Helling, D. E., and Miller, A. K. (1987). The Incorporation of Yield Surface Distortion into a Unified Constitutive Model, Part 1: Equation Development. *Acta Mechanica* 69, 9–23. doi:10.1007/bf01175711
- Hill, R. (1948). A Theory of the Yielding and Plastic Flow of Anisotropic Metals. *Proc. R. Soc. Lond. Ser. A. Math. Phys. Sci.* 193, 281–297.
- Huber, M. T. (1904). Przyczynek Do Podstaw Wytorymalosci. *Czasop Techn* 22, 81.
- Ibañez, R., Abisset-Chavanne, E., Aguado, J. V., Gonzalez, D., Cueto, E., and Chinesta, F. (2018). A Manifold Learning Approach to Data-Driven Computational Elasticity and Inelasticity. *Arch. Computat Methods Eng.* 25, 47–57. doi:10.1007/s11831-016-9197-9
- Karafillis, A. P., and Boyce, M. C. (1993). A General Anisotropic Yield Criterion Using Bounds and a Transformation Weighting Tensor. *J. Mech. Phys. Sol.* 41, 1859–1886. doi:10.1016/0022-5096(93)90073-o
- Kirchdoerfer, T., and Ortiz, M. (2016). Data-driven Computational Mechanics. *Comput. Methods Appl. Mech. Eng.* 304, 81–101. doi:10.1016/j.cma.2016.02.001
- Kroese, D. P., and Rubinstein, R. Y. (2012). Monte Carlo Methods. *Wires Comp. Stat.* 4, 48–58. doi:10.1002/wics.194
- Kurtyka, T., and Życzkowski, M. (1996). Evolution Equations for Distortional Plastic Hardening. *Int. J. Plasticity* 12, 191–213. doi:10.1016/s0749-6419(96)00003-4
- Linka, K., Hillgärtner, M., Abdolazizi, K. P., Aydin, R. C., Itskov, M., and Cyron, C. J. (2021). Constitutive Artificial Neural Networks: A Fast and General Approach to Predictive Data-Driven Constitutive Modeling by Deep Learning. *J. Comput. Phys.* 429, 110010. doi:10.1016/j.jcp.2020.110010
- Liu, Z., Kafka, O. L., Yu, C., and Liu, W. K. (2018). “Data-driven Self-Consistent Clustering Analysis of Heterogeneous Materials with crystal Plasticity,” in *Adv. Comput. Plast.* (Berlin, Germany: Springer), 221–242. doi:10.1007/978-3-319-60885-3\_11
- Marques, R., Bouville, C., Ribardiè, M., Santos, L. P., and Bouatouch, K. (2013). Spherical Fibonacci point Sets for Illumination Integrals. *Comput. Graphics Forum* 32, 134–143. doi:10.1111/cgf.12190
- Marsaglia, G. (1972). Choosing a point from the Surface of a Sphere. *Ann. Math. Statist.* 43, 645–646. doi:10.1214/aoms/1177692644
- Nurmela, K. J. (1995). *Constructing Spherical Codes by Global Optimization Methods*. Helsinki: Helsinki University of Technology.
- Peranio, N., Li, Y. J., Roters, F., and Raabe, D. (2010). Microstructure and Texture Evolution in Dual-phase Steels: Competition between Recovery, Recrystallization, and Phase Transformation. *Mater. Sci. Eng. A* 527, 4161–4168. doi:10.1016/j.msea.2010.03.028
- Pian, T. H. H., and Wu, C. C. (2005). *Hybrid and Incompatible Finite Element Methods*. Abingdon-on-Thames, Oxfordshire, UK: Taylor & Francis. Available at: <https://books.google.de/books?id=t595AgAAQBAJ>.

## DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding author.

## AUTHOR CONTRIBUTIONS

Both authors have contributed equally to the work.

## ACKNOWLEDGMENTS

RS gratefully acknowledges helpful discussions on the mathematical problems with Erik Brinkman. AH gratefully acknowledges funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—Project-ID 190389738—TRR 103.

- Raychaudhuri, S. (2008). "Introduction to Monte Carlo Simulation," in *2008 Winter Simul. Conf.* (Piscataway, NJ, USA: IEEE), 91–100. doi:10.1109/wsc.2008.4736059
- Roters, F., Eisenlohr, P., Bieler, T. R., and Raabe, D. (2011). *Crystal Plasticity Finite Element Methods: In Materials Science and Engineering*. Hoboken, NJ, USA: Wiley. Available at: <https://books.google.de/books?id=WFSNYRvc-ZYC>.
- Shultz, T. R., Fahlman, S. E., Craw, S., Andritsos, P., Tsaparas, P., Silva, R., et al. (2011). Confusion Matrix. *Encycl. Mach. Learn.* 61, 209. doi:10.1007/978-0-387-30164-8\_157
- Sloane, N. J. A., Hardin, R. H., and Smith, W. D. (2000). *Spherical Codes*. Available at: <http://www2.Res.Att.Com/~Njas/Packings>.
- Suykens, J. A. K., and Vandewalle, J. (1999). Least Squares Support Vector Machine Classifiers. *Neural Process. Lett.* 9, 293–300.
- Swinbank, R., and James Purser, R. (2006). Fibonacci Grids: A Novel Approach to Global Modelling. *Q.J.R. Meteorol. Soc.* 132, 1769–1793. doi:10.1256/qj.05.227
- v Mises, R. (1913). Mechanik der festen Körper im plastisch-deformablen Zustand, Nachrichten von Der Gesellschaft Der Wissenschaften Zu Göttingen. *Math. Klasse.* 1913, 582–592.
- Vajragupta, N., Ahmed, S., Boeff, M., Ma, A., and Hartmaier, A. (2017). Micromechanical Modeling Approach to Derive the Yield Surface for BCC and FCC Steels Using Statistically Informed Microstructure Models and Nonlocal crystal Plasticity. *Phys. Mesomech.* 20, 343–352. doi:10.1134/s1029959917030109

**Conflict of Interest:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

**Publisher's Note:** All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2022 Shoghi and Hartmaier. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



## APPENDIX A

Since in most metals hydrostatic stress components do not play a significant role in plastic deformation, it is useful to analyze their flow stresses in the deviatoric stress space. To accomplish this, oftentimes a transform to principal stresses is applied. All deviatoric principal stresses lie on a plane in the stress space, the so-called  $\pi$ -plane. In this work, we use a cylindrical coordinate system to plot stresses in this plane, where the equivalent stress  $\sigma_{eq}$  is plotted along the  $y$ -axis and the polar angle  $\theta$  is plotted along the  $x$ -axis. This projection is best introduced via defining a complex-valued stress deviator based on the vector of deviatoric principal stresses  $\boldsymbol{\sigma}' = (\sigma_1 - p, \sigma_2 - p, \sigma_3 - p)$ , where  $\sigma_1, \sigma_2, \sigma_3$  are the principal stresses and  $p$  is the hydrostatic stress. Furthermore, it is necessary to specify two arbitrary

orthogonal directions in the  $\pi$ -plane, for which in this work  $\mathbf{a} = (2, -1, -1)/\sqrt{6}$  and  $\mathbf{b} = (0, 1, -1)/\sqrt{2}$  are chosen. Note that these unit vectors span the plane normal to the hydrostatic axis  $\mathbf{c} = (1, 1, 1)/\sqrt{3}$ . Following the method developed in previous work (Hartmaier, 2020), the complex-valued deviatoric stress is defined as

$$\sigma'_c = \boldsymbol{\sigma} \cdot \mathbf{a} + i\boldsymbol{\sigma} \cdot \mathbf{b} = \sqrt{2/3} \cdot \sigma_{eq} \cdot e^{i\theta} \quad (32)$$

In this definition,  $i$  is the imaginary unit and the polar angle  $\theta$  can be evaluated by

$$\theta = -i \ln \frac{\boldsymbol{\sigma} \cdot \mathbf{a} + i\boldsymbol{\sigma} \cdot \mathbf{b}}{\sqrt{2/3}\sigma_{eq}} \quad (33)$$

This mapping of the stress is used in **Figures 4, 5, 10, 16**.