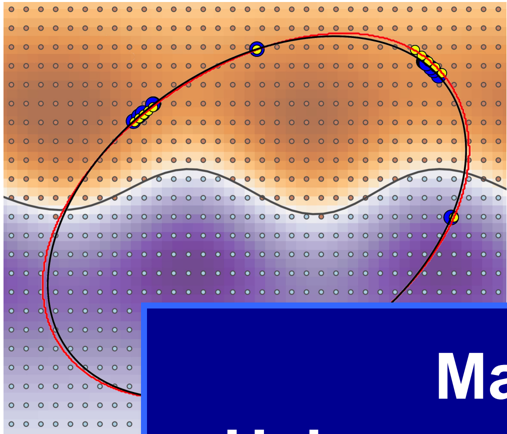


Spring School SFB 986 **25.03.2021**

Machine learning and data mining: Using machine learning methods to homogenize mechanical properties from micro- to macroscale

A. Hartmaier

Interdisciplinary Centre for Advanced Materials Simulation (ICAMS)

Ruhr-Universität Bochum

alexander.hartmaier@rub.de

Outline

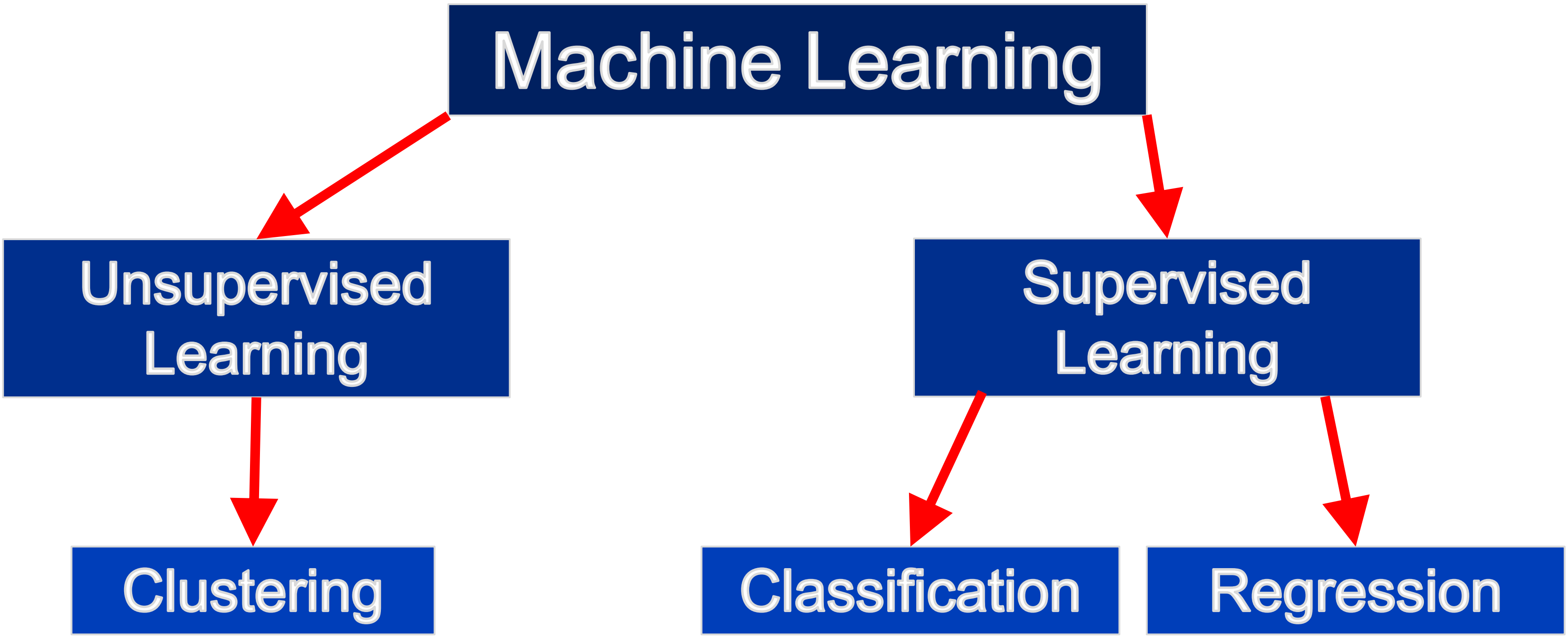
I. Lecture
   1. Machine Learning (ML) methods
   2. Applications as surrogate models
   3. Damage homogenization
   4. Microstructure-property relationships

II. Tutorials
   1. ML-Regression
   2. ML-Classification

Use tutorials on Binder:
https://mybinder.org/v2/gh/AHartmaier/ML-Tutorial.git/HEAD

Installation from GitHub repository:
https://github.com/AHartmaier/ML-Tutorial.git

Machine Learning



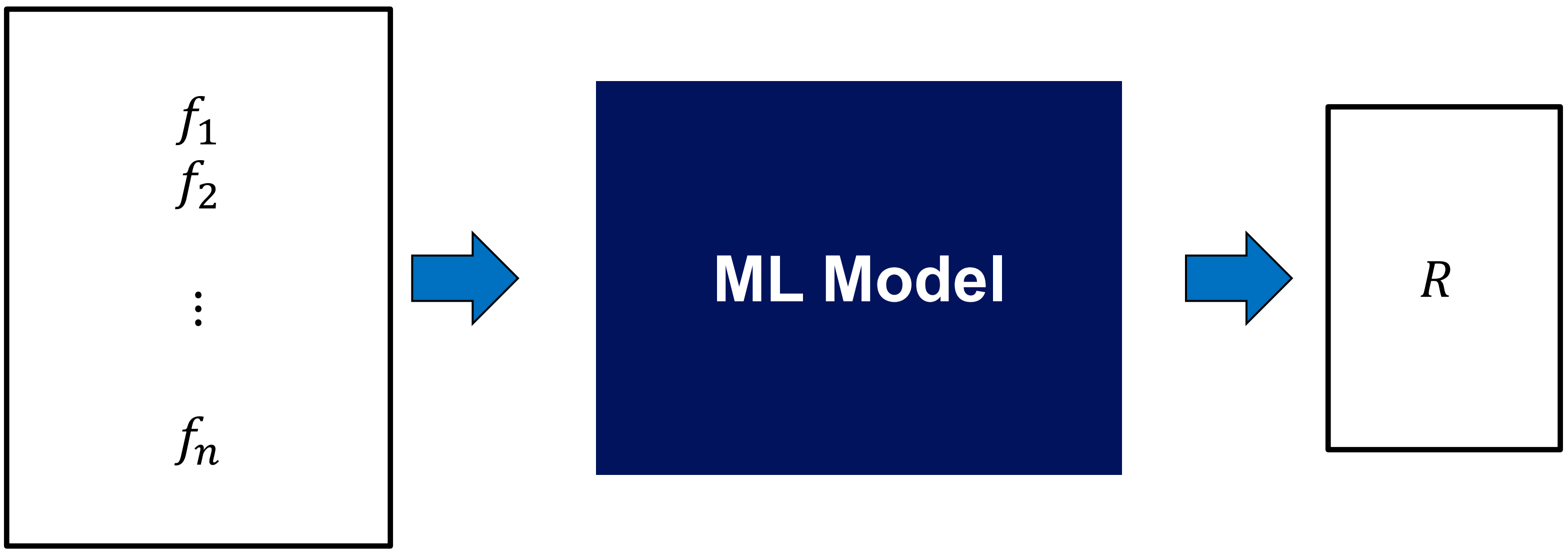
All examples of this lecture have been performed with scikit-learn (https://scikit-learn.org/stable/)

Machine Learning (ML) Modeling

input vector / "features"

Selection of features (or descriptors) determines the physics of the ML model

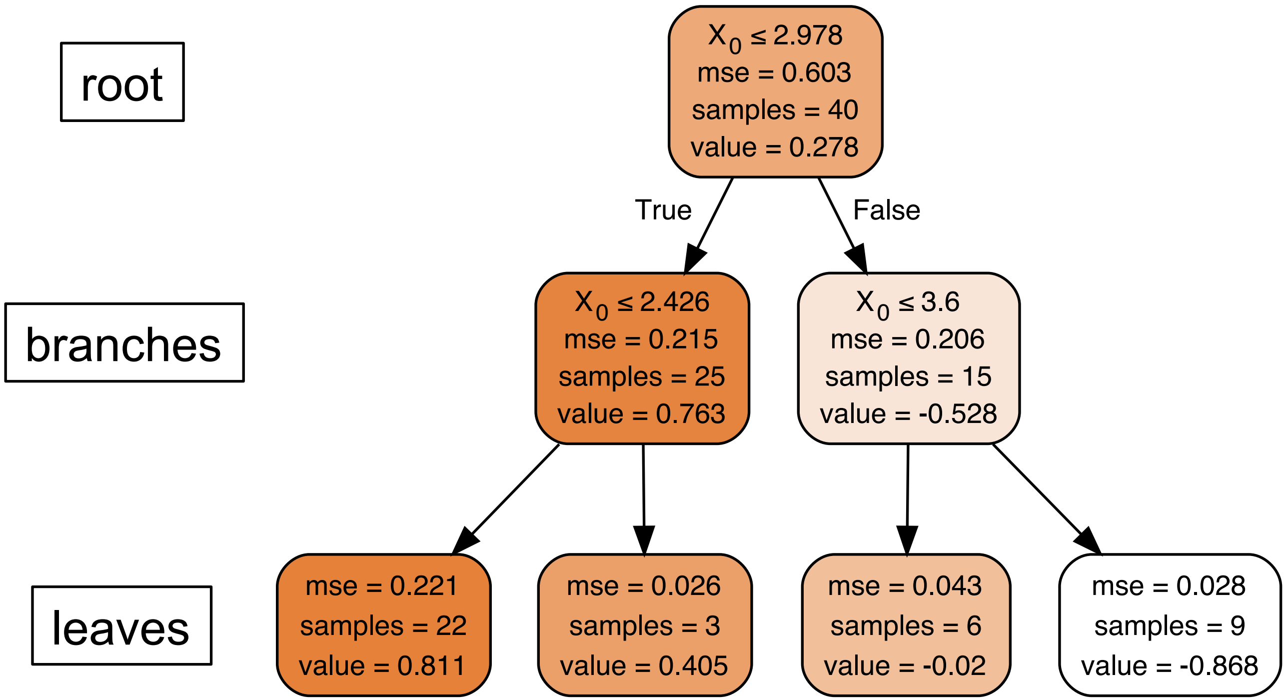
output vector / "labels"



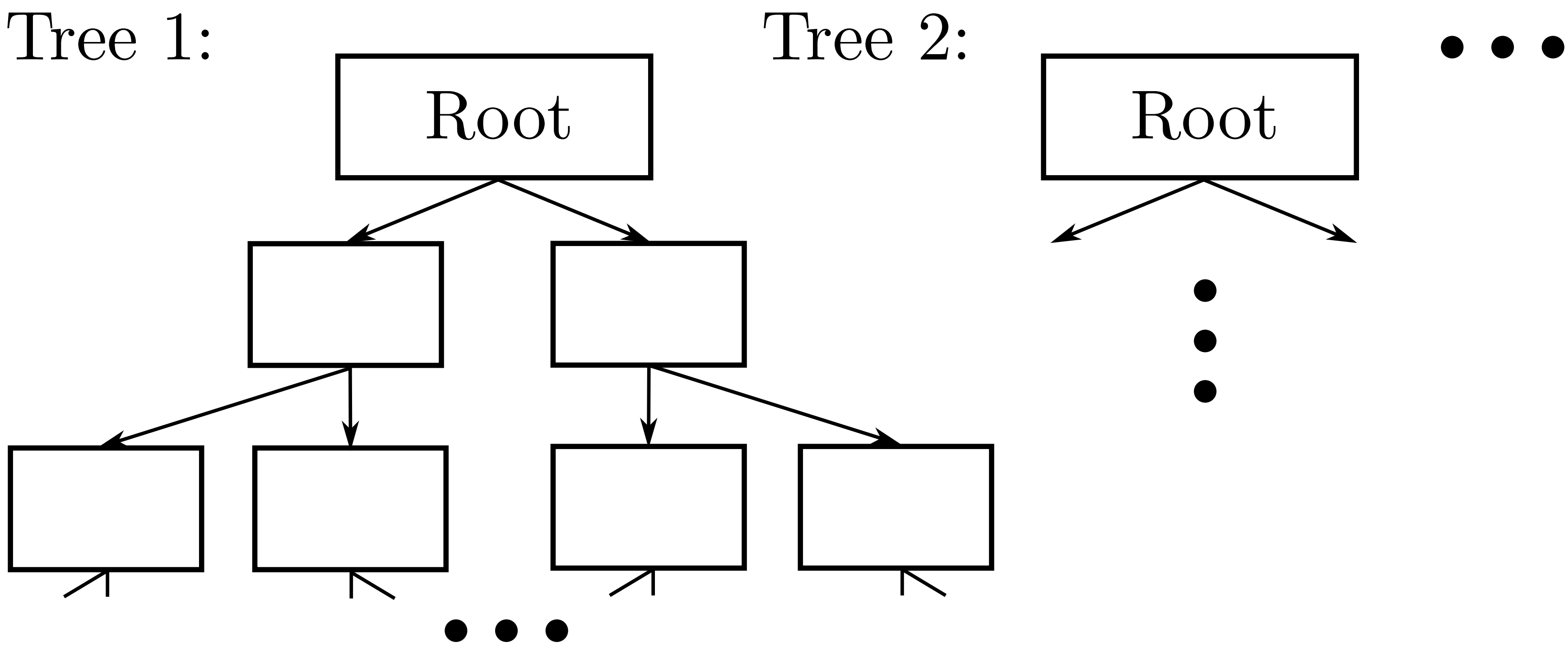
Supervised training
Find ML parameters that minimize deviation between result of ML model and known data point (ground truth).

Decision Tree Regression

Succession of if-clauses leads to final result in “leaves”



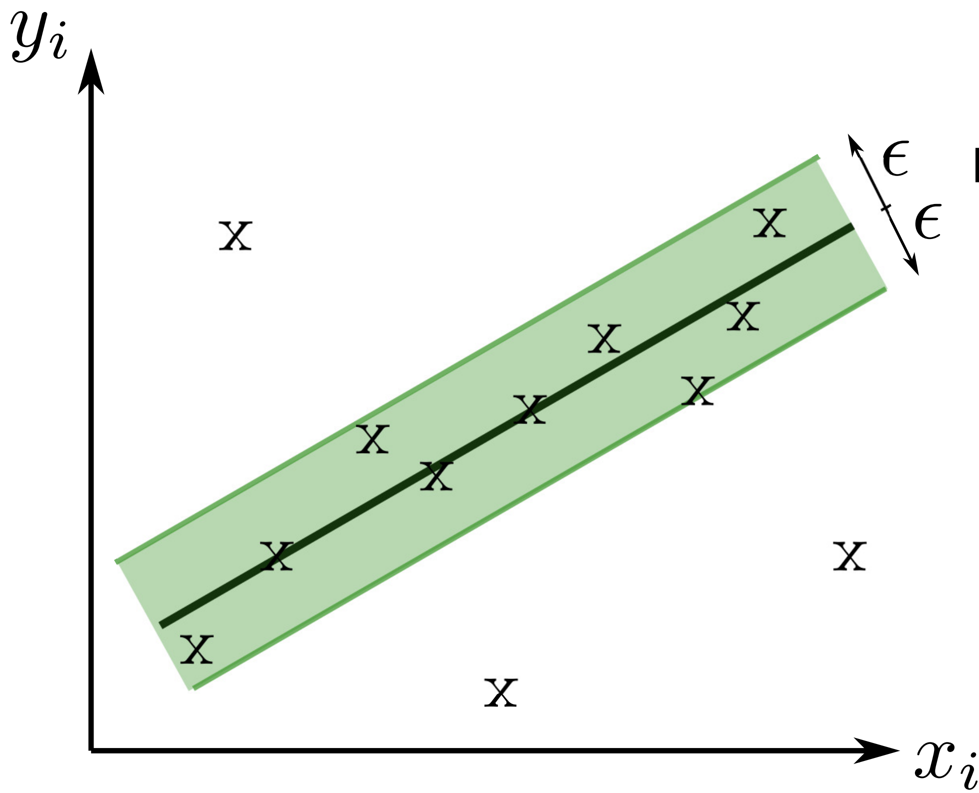
Random Forest Regression



Goal: Create model that predicts output value for given input data by learning simple decision rules

- Number of trees = 100 … 500
- Leaves contain either 1 or 0 samples
- Final result is average of all sample values

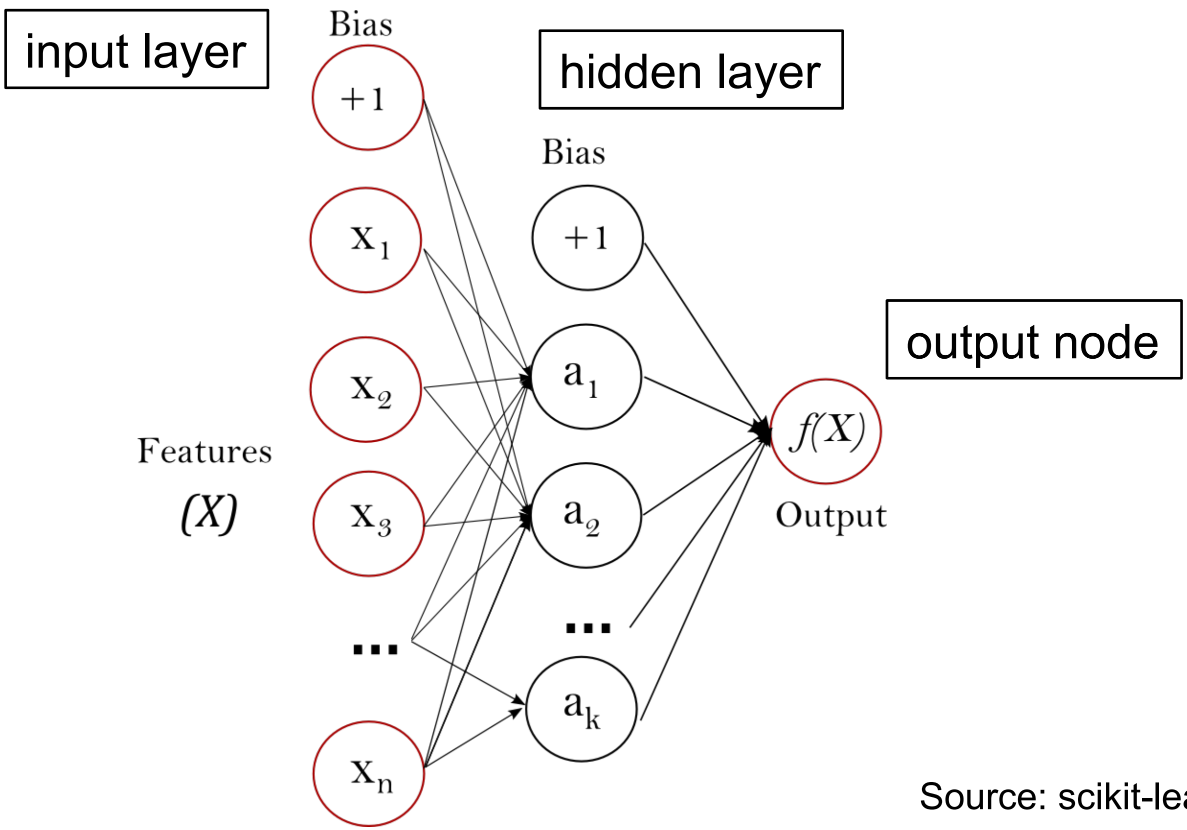
Support Vector Regression/Classification



Goal: Find a function such that data points lie within a corridor of $\pm\epsilon$ (function as flat as possible, actual error unimportant, penalty for outliers)

- Linear or Gaussian kernel for interpolation between support vectors
- Support vectors determined during training function (data points closest to delimiter line)

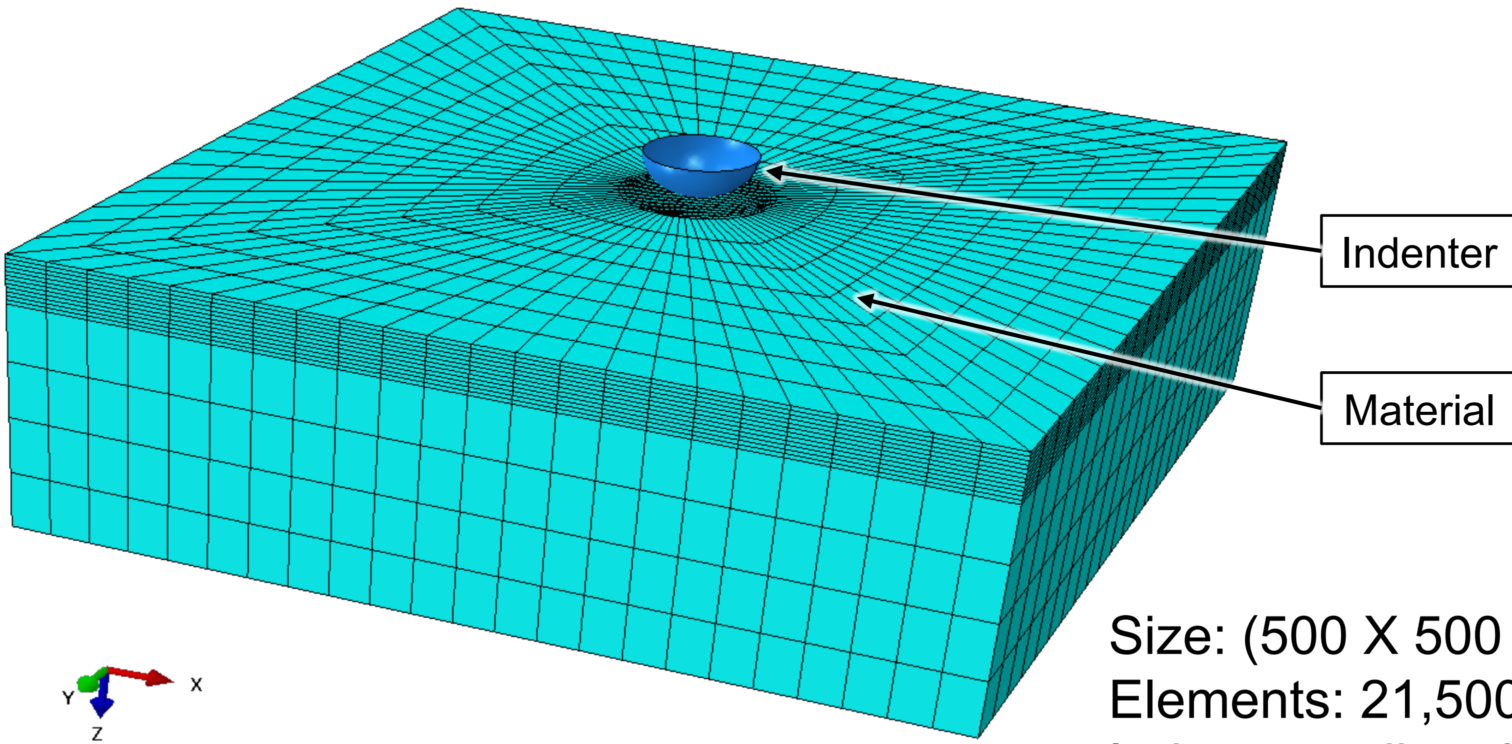
Neural Networks



Source: scikit-learn

Goal: Find bias values and activation functions that describe training data best. – *Deep learning*: multiple hidden layers.

Finite Element Model



Size: (500 X 500 X 200) μm
Elements: 21,500
Indenter radius: 30 μm

- Finite element model gives accurate description of indentation process
- Simulation times hours to days, depending on model size and constitutive model

Data generation for training of ML surrogate model

Isotropic hardening

$$R = Q(1 - e^{-b\varepsilon_{eq}})$$

Non-linear kinematic hardening

$$\boldsymbol{\kappa} = \sum_{i}^{n} \boldsymbol{\kappa}_i; \, d\boldsymbol{\kappa}_i = \frac{2}{3} C_i d\boldsymbol{\varepsilon}_p - g_i \boldsymbol{\kappa}_i d\varepsilon_{eq}$$

Creep/time-dependent deformation

$$\begin{aligned} \dot{\varepsilon}^{cr} &= \mathrm{A}\tilde{q}^n t^m \\ &= A_0 \left(\frac{q}{q_0}\right)^n \left(\frac{t}{t_0}\right)^m \end{aligned}$$

1000 different combinations of material parameters are generated randomly from defined ranges.

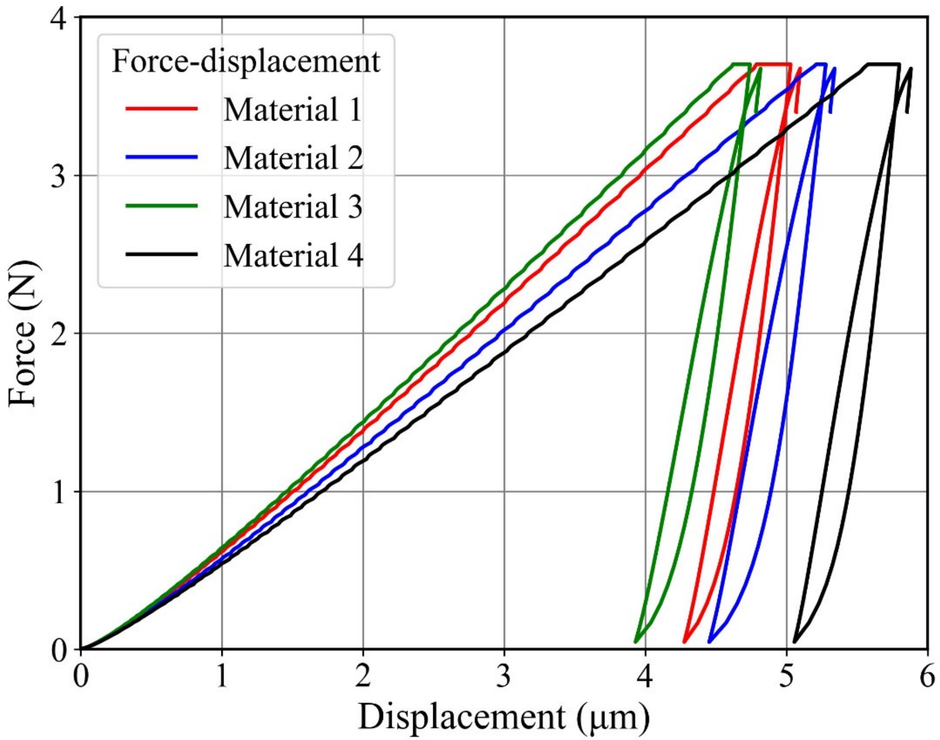
Material Parameter Ranges

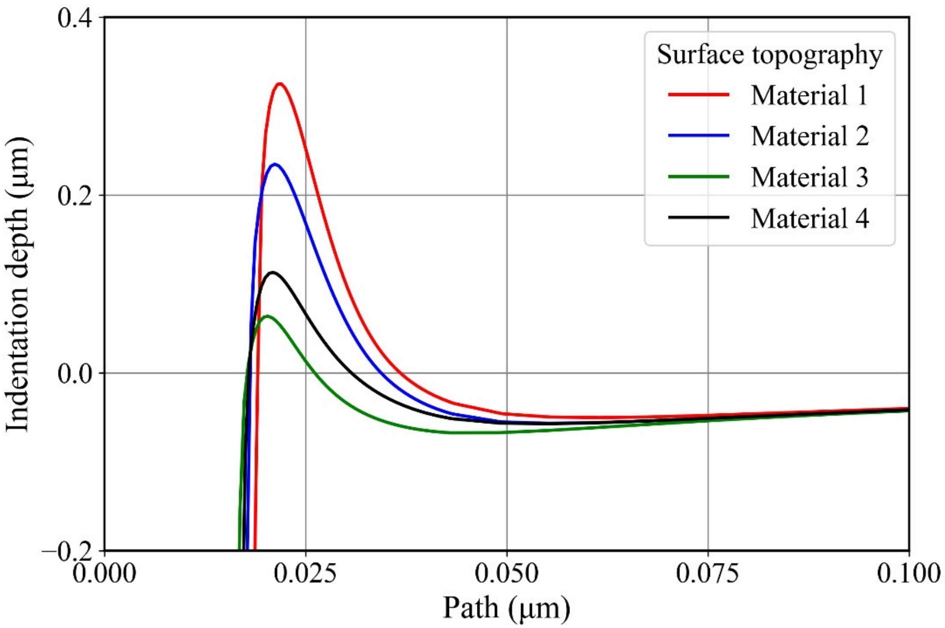
Parameter	Min	Max
A_0, 1/s	1E-07	1E-05
n, -	1.75	3.0
m, -	-0.95	-0.5
C_1, MPa	125000	225000
g_1, -	350	550
C_2, MPa	3000	5500
Q, MPa	-350	-1750
b, -	0.5	25

$$A = A_0 \, (750\mathrm{MPa})^{-n} \, (100\mathrm{s})^{-m}$$

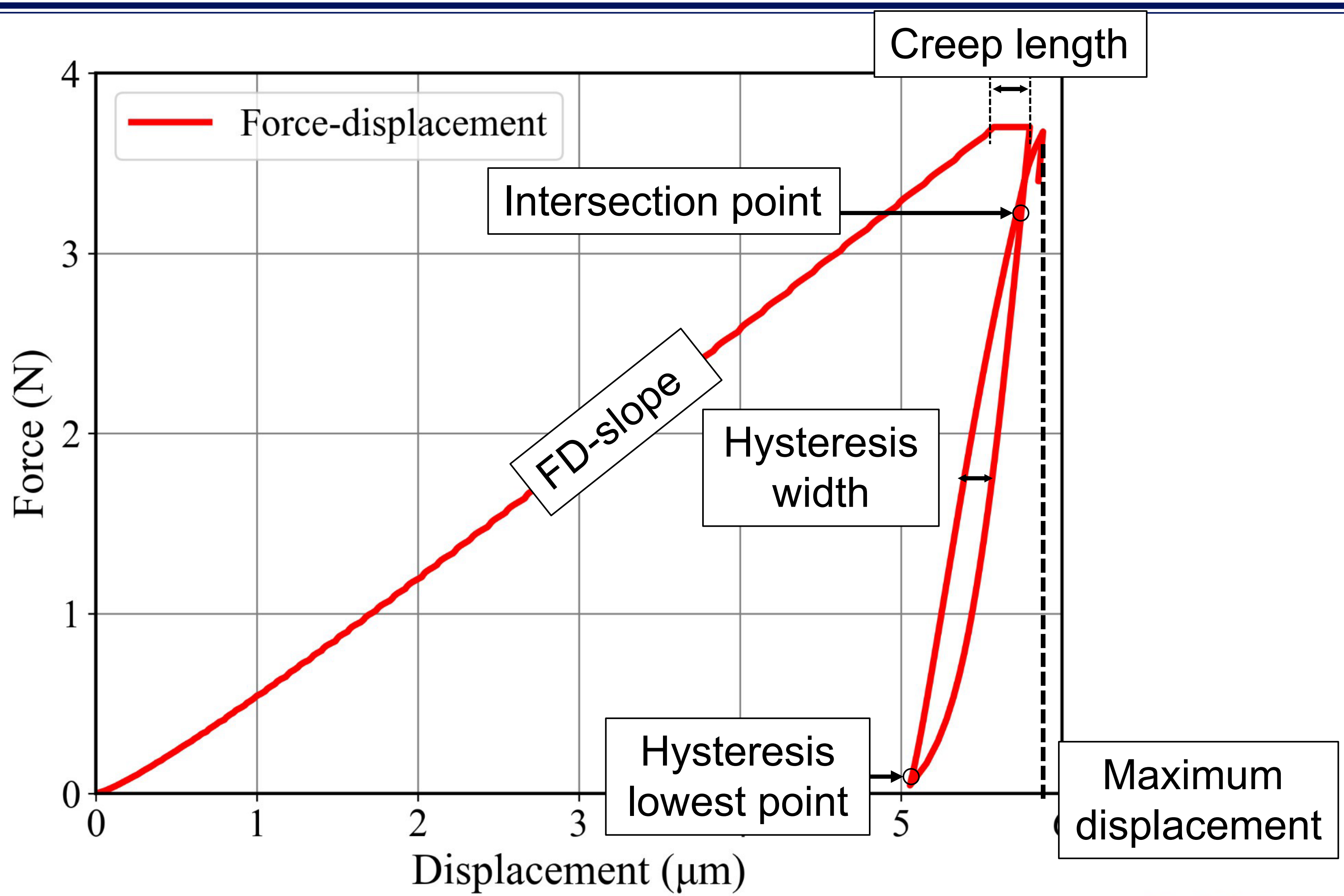
Database generation

Simulation output

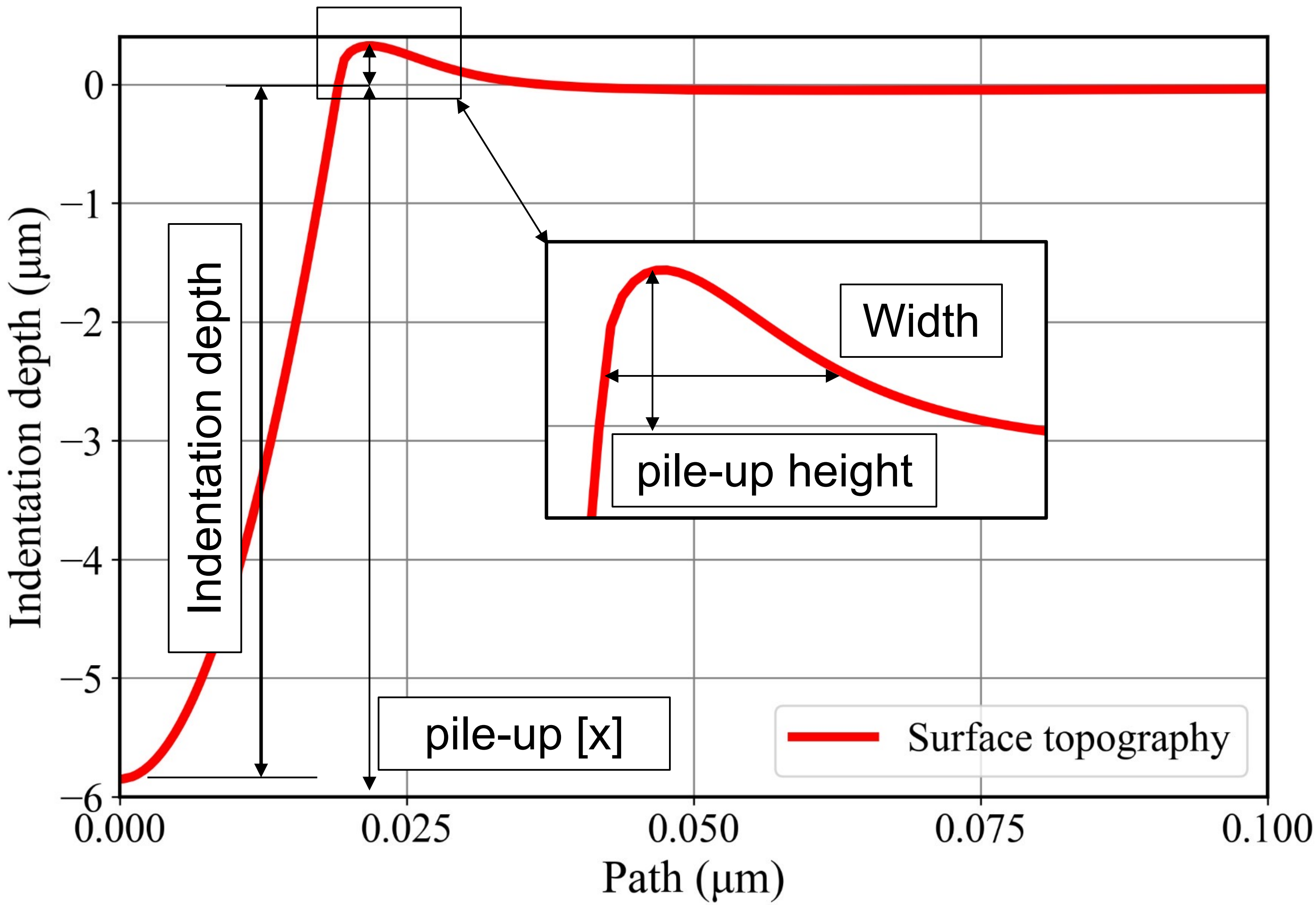




Definition of labels: Force-displacement curve

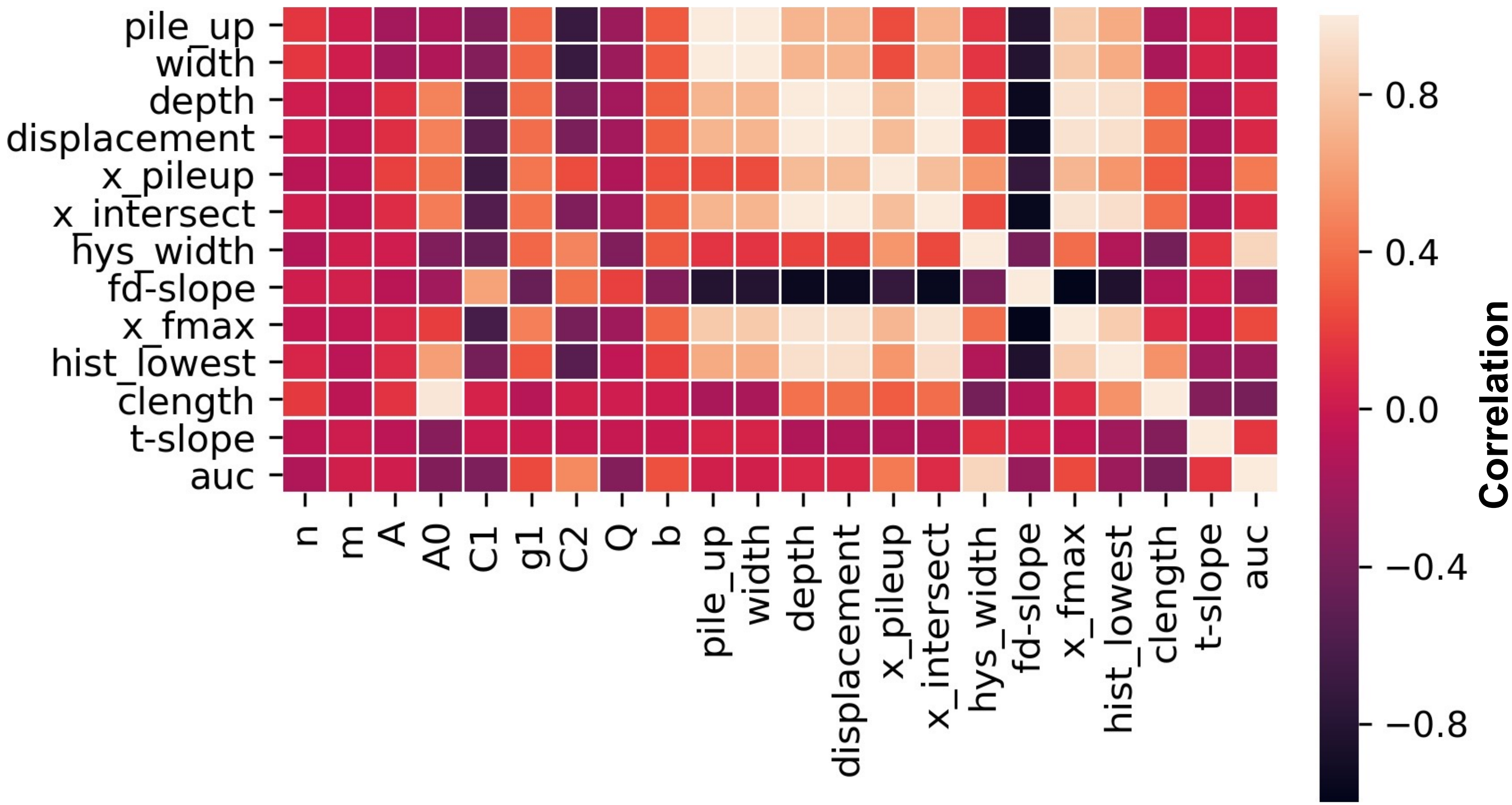


Definition of labels: Residual imprints

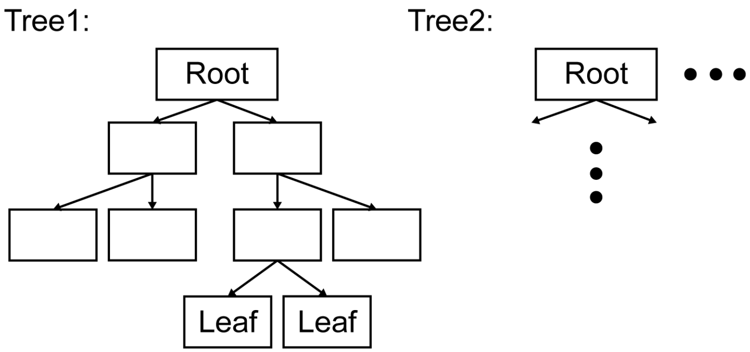


Feature selection

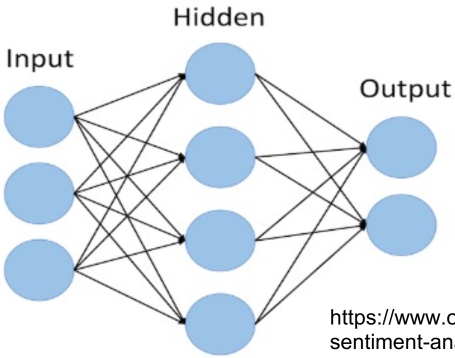
Heat map of extracted features and labels



Training and testing of ML algorithms

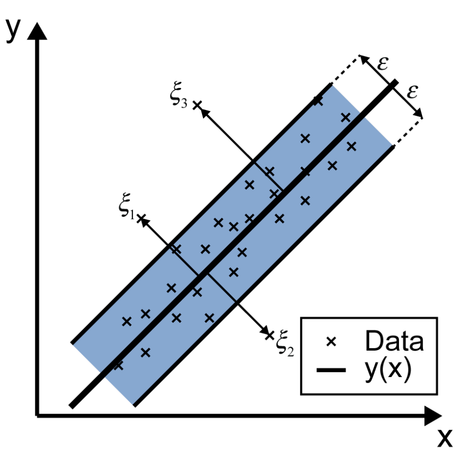


Random forest regression



https://www.oreilly.com/content/perform-sentiment-analysis-with-lstms-using-tensorflow/

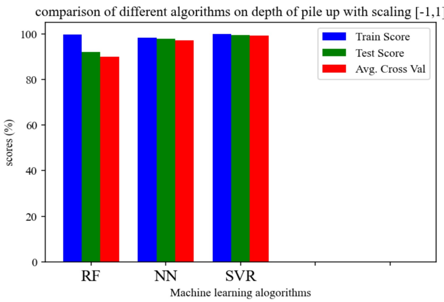
Artificial neural networks



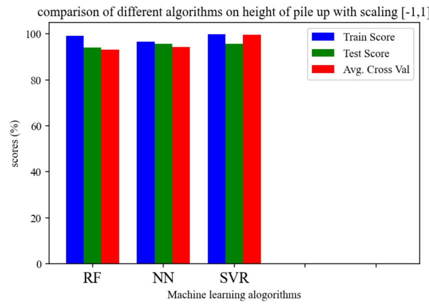
Support vector regression

- Machine learning library on Scikit Learn
- Random Forest Regression (RFR), Support Vector Regression (SVR), and Neural Networks (NN) are chosen.
- 75% training data and 25% testing data
- Grid search for determining hyperparameters

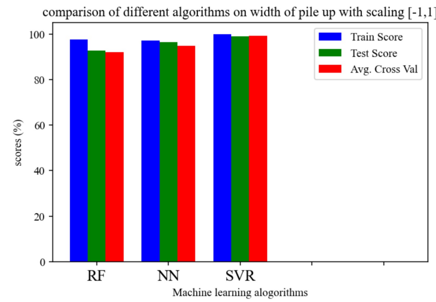
Training and testing results



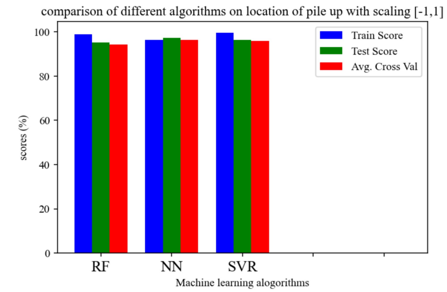
Indentation depth



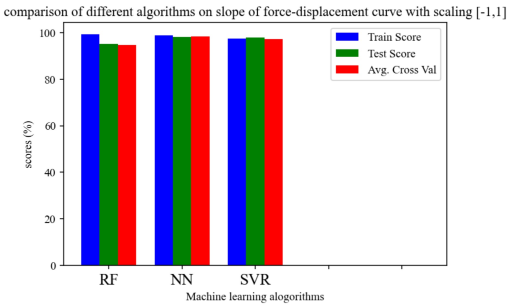
Pile-up height



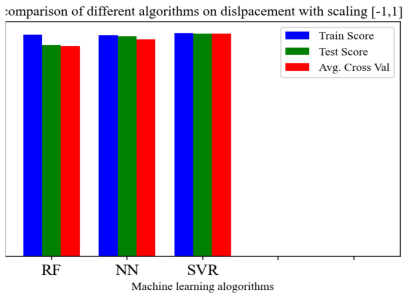
Pile-up width



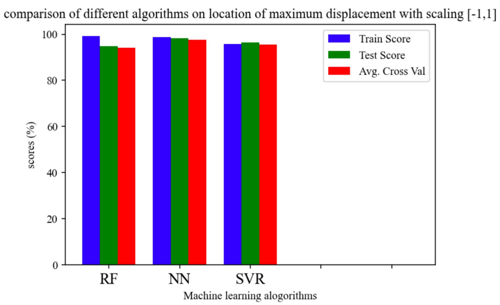
Pile-up location



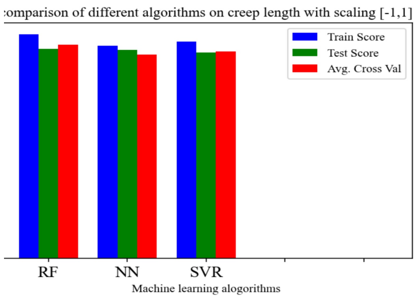
F-D slope



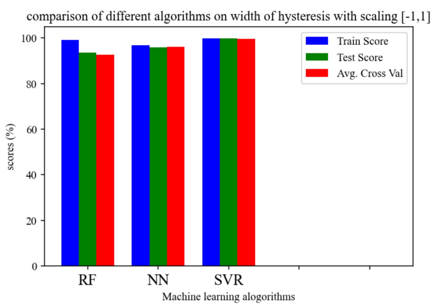
Displacement



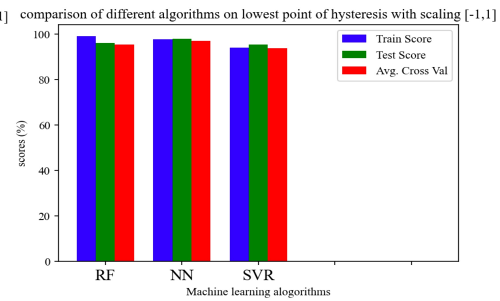
Max. displacement



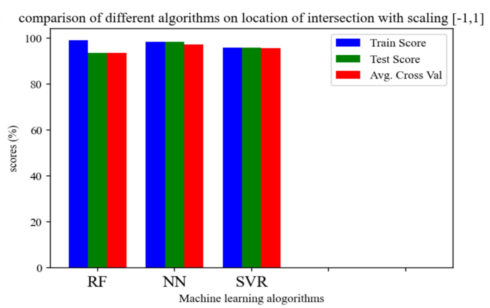
Creep length



Hysteresis width

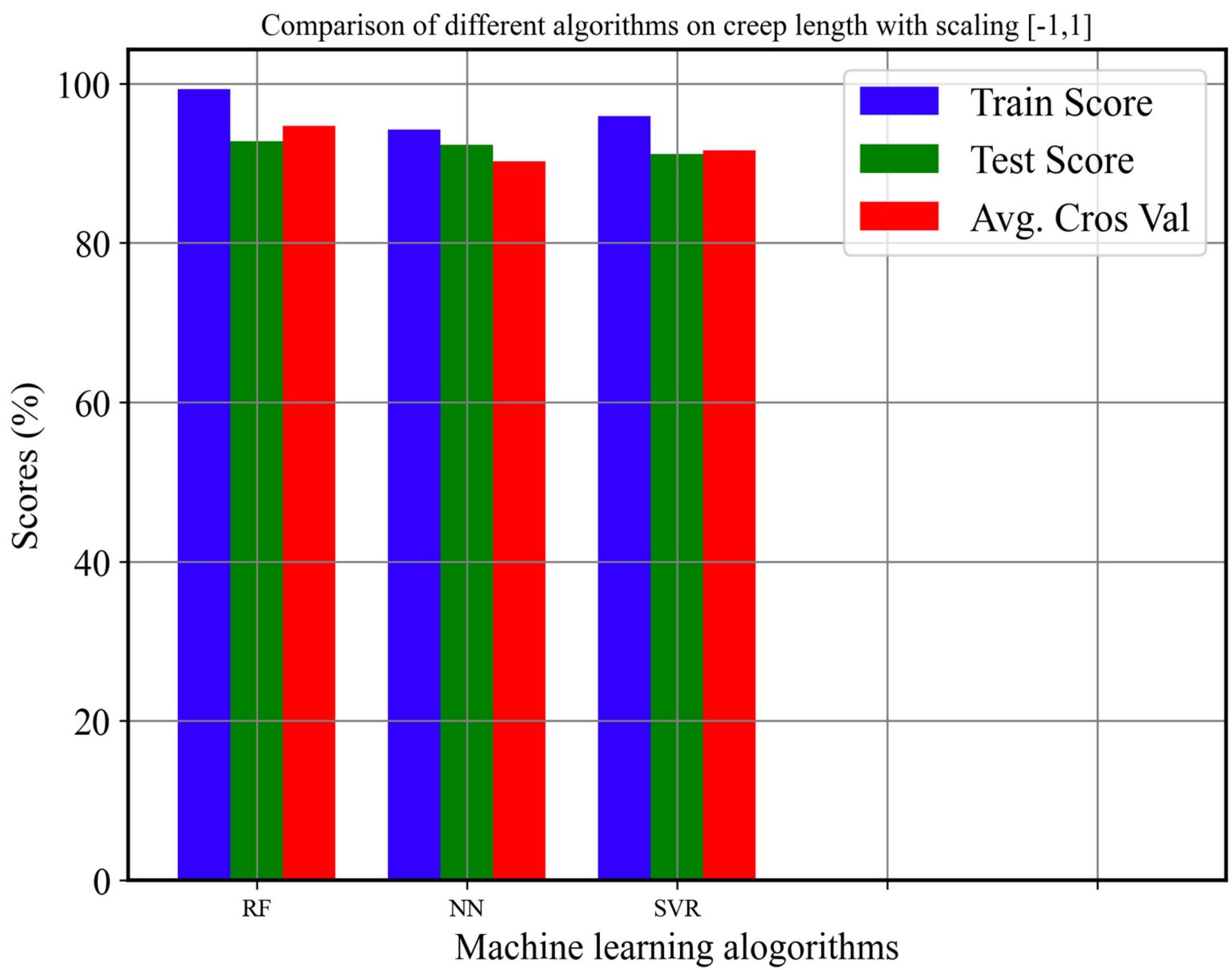


Hyst. lowest point

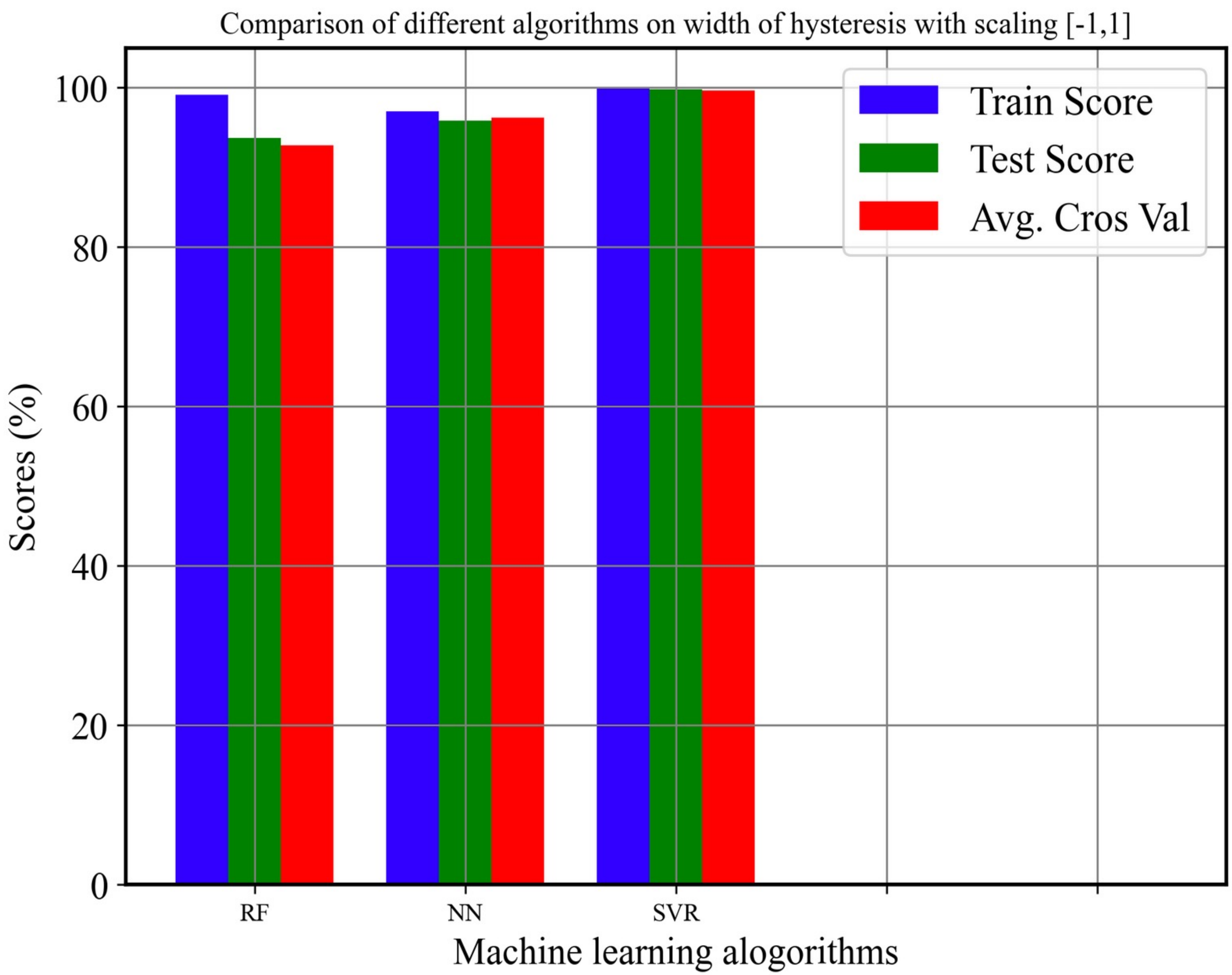


Intersection point

Creep length



Hysteresis width



Validation of surrogate model

	Pile up (µm)	x-pile up (µm)	Width (µm)	Depth (µm)	Displace-ment (µm)	X-intersect (µm)	Hys-width (µm)	Hys-lowest (µm)	Creep length (µm)	F-D slope (N/µm)
FEM	0.244	22.17	0.977	5.476	5.512	5.411	0.360	4.407	0.052	0.684
NN	0.230	22.22	0.953	5.568	5.576	5.572	0.353	4.686	0.062	0.676
NN Rel. dif. (%)	3.50%	0.23%	2.51%	1.68%	1.16%	2.97%	2.06%	6.32%	18.14%	1.28%
SVR	0.236	21.83	0.957	5.441	5.471	5.461	0.330	4.610	0.054	0.671
SVR Rel. dif. (%)	6.00%	1.52%	2.04%	0.64%	0.73%	0.91%	8.37%	4.60%	2.84%	1.96%
RFR	0.226	21.58	0.899	5.254	5.298	5.230	0.332	4.454	0.044	0.704
RFR Rel. dif. (%	7.53%	2.65%	8.04%	4.04%	3.87%	3.35%	7.82%	1.07%	14.26%	2.87%

Trained ML models are fed with unknown material parameters and compared with FEM simulation for validation

Relative difference:

$$\text{Rel. diff.} = \left|\frac{\text{FEM}-\text{predicted value}}{\text{FEM}}\right| * 100$$

Summary – Surrogate model

- Finite Element (FE) simulations can model mechanical problems with high accuracy
- For repeated tasks, as for example for inverse methods or optimization problems, the numerical cost of FE simulations poses a severe restriction
- Trained ML models can be used as numerically efficient surrogate models for such tasks – training effort only occurs once

Co-authors: H.M. Sajjad, Z. Hamzeh, P. Nooshmer, N. Vajragupta

unpublished work

Macroscopic Damage Modeling

input vector | **output vector**

p
$\dot{p}$
σ_{eq}
σ_{hydr}
σ_{tri}
D
$\vdots$

→ **Damage Model** →

$\dot{D}$

Parameterization with experiment

Macroscopic Damage Modeling

input vector

p
$\dot{p}$
σ_{eq}
σ_{hydr}
σ_{tri}
D
$\vdots$
microstructure

Trained Machine Learning Regressor

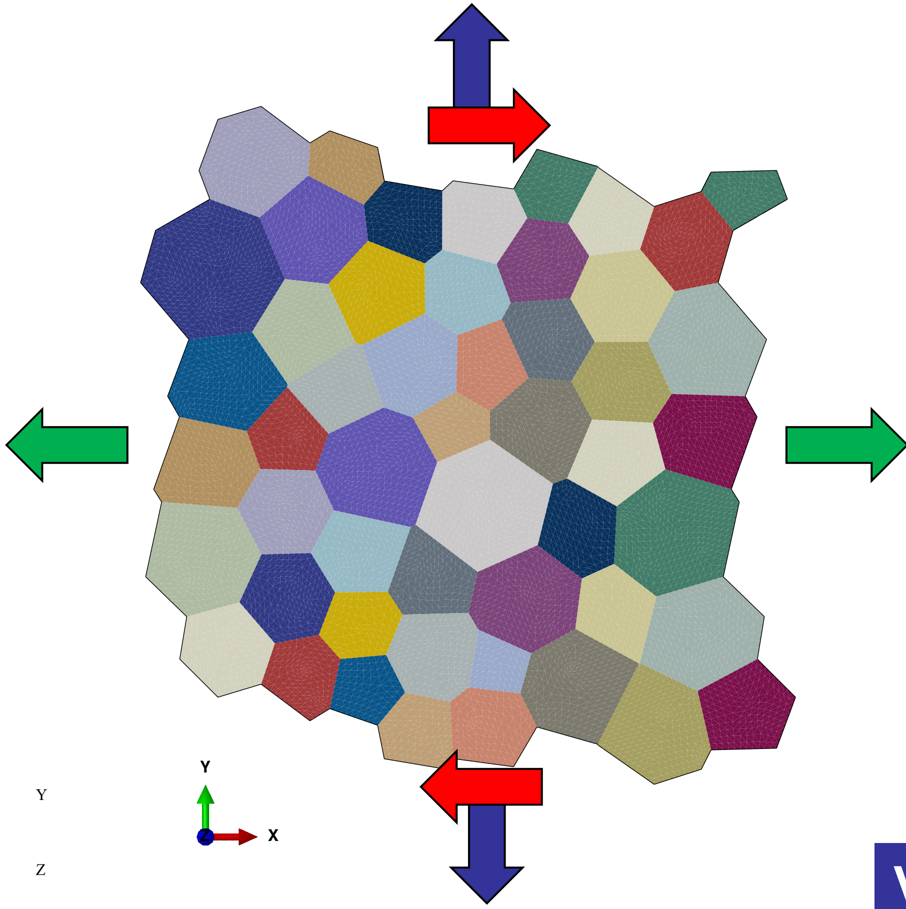
output vector

$\dot{D}$

Parameterization with ~~experiment~~

micromechanical simulation

Damage: Micromechanical model



- Number of grains: 51
- Random orientation of grains
- Grain size: 45-90 μm
- Number of elements: 9351
- Monotonic loading (20% strain)
- Periodic boundary conditions

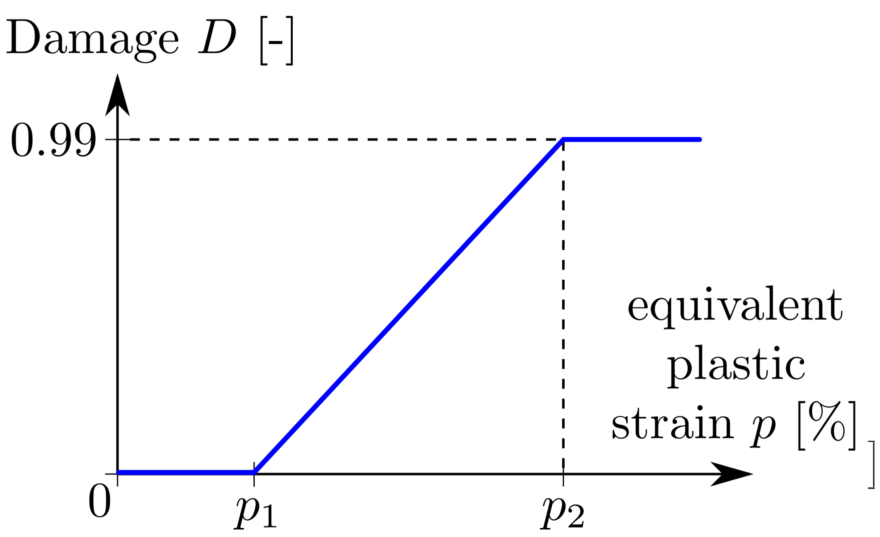
Virtual mechanical testing along various load paths (uniaxial, bi-axial, shearing, …)

Material model

Phenomenological crystal plasticity with shear rate evolution law and isotropic hardening:

$$\dot{\gamma} = \dot{\gamma}_0 \left|\frac{\tau}{\tau_c}\right|^m \operatorname{sign}(\tau) \qquad \dot{\tau}_c = \sum h_0 \left(1 - \frac{\tau_c}{\tau_s}\right)^n M \left|\dot{\gamma}\right|$$

Damage depending on the equivalent plastic strain:

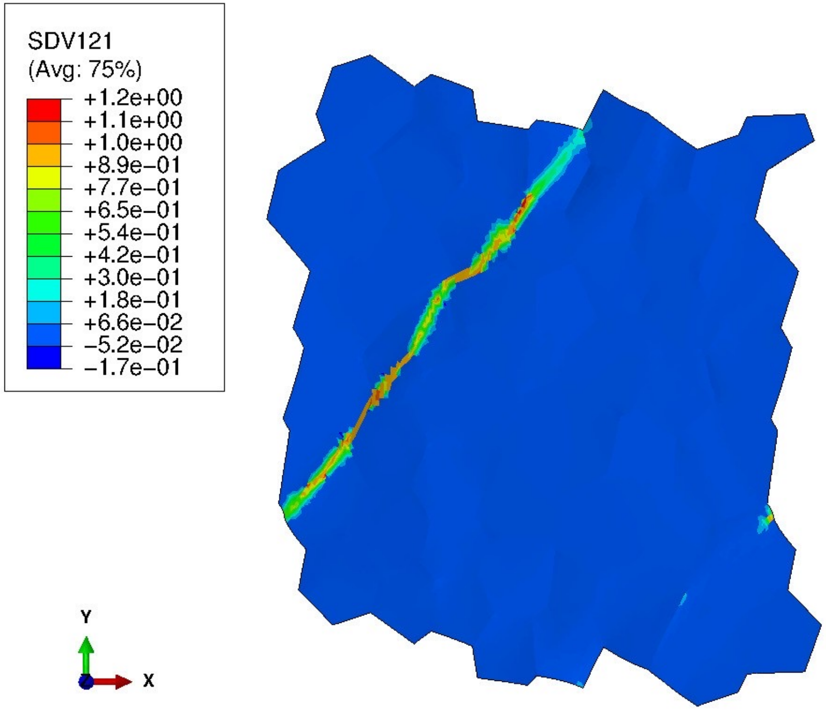


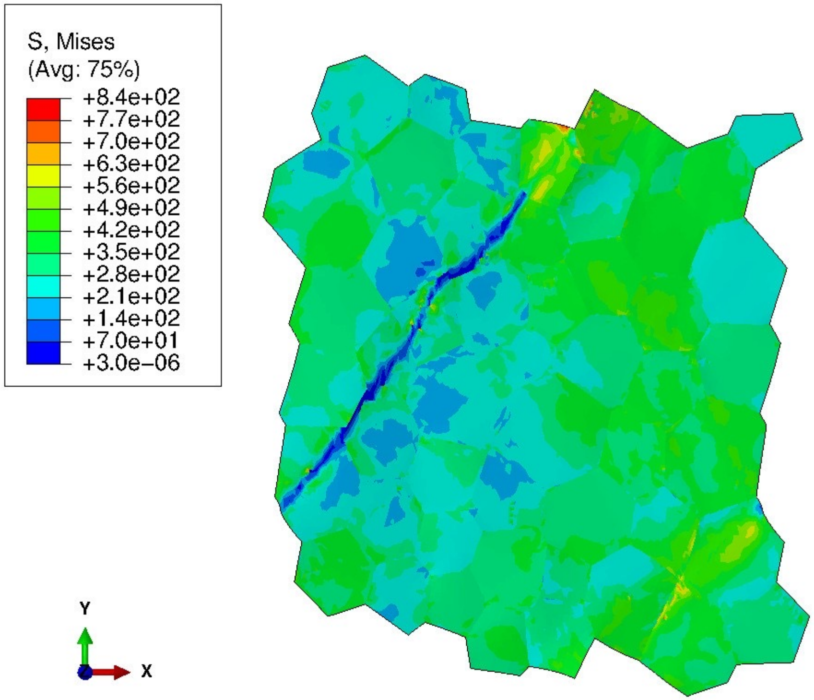
$$D = \frac{p - p_1}{p_2 - p_1}$$

with p_1 = 0.3
and p_2 = 0.5

Damage: Homogenization of Micromechanical Data

Damage and von Mises stress at time 1.66s (increment: 600)





Macroscopic (homogenized) damage

$$D^{\mathrm{RVE}} = \frac{\text{effective structural stiffness } C_D}{\text{initial stiffness } C_0}$$

Other quantities homogenized by volume averaging

Data extraction

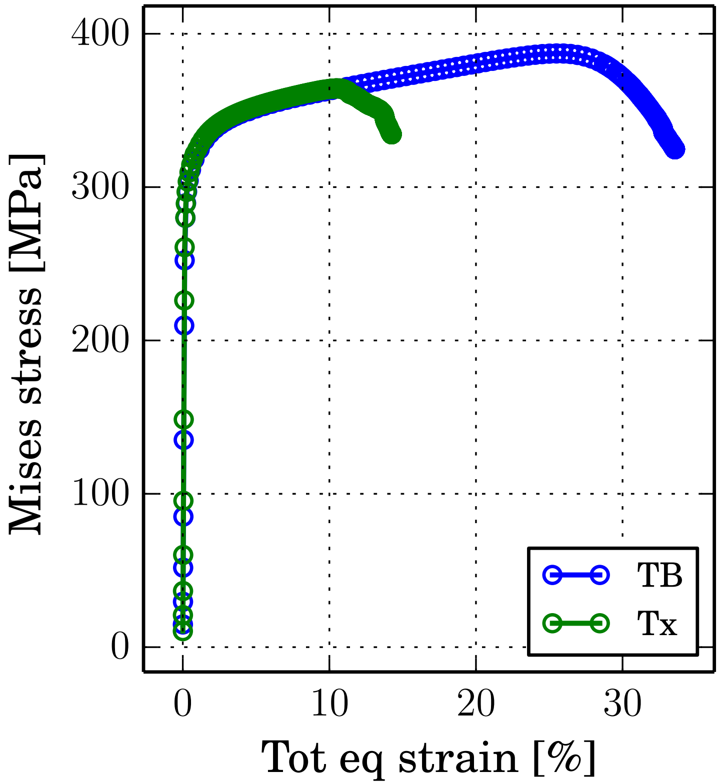
Extract local values of mechanical quantities (one value for each element):

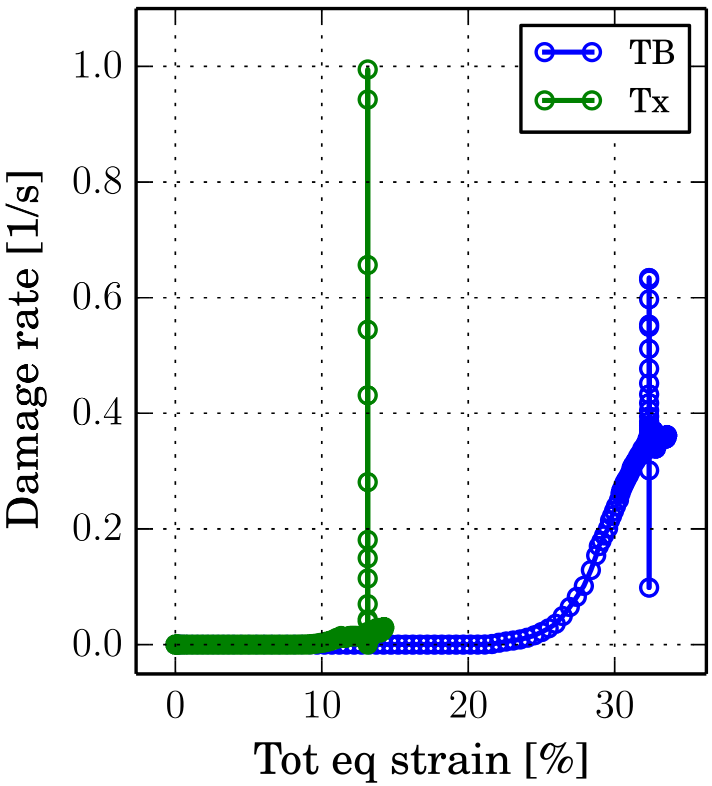
- equivalent plastic strain
- equivalent plastic strain rate
- equivalent total strain
- equivalent elastic strain
- equivalent stress
- hydrostatic stress
- element volume

Global data by averaging local values with element volume:

$$(\blacksquare)^{\mathrm{global}} = \frac{1}{V_{\mathrm{RVE}}} \sum_{\mathrm{elements}} (\blacksquare)^{\mathrm{local}} \, V_{\mathrm{element}}$$

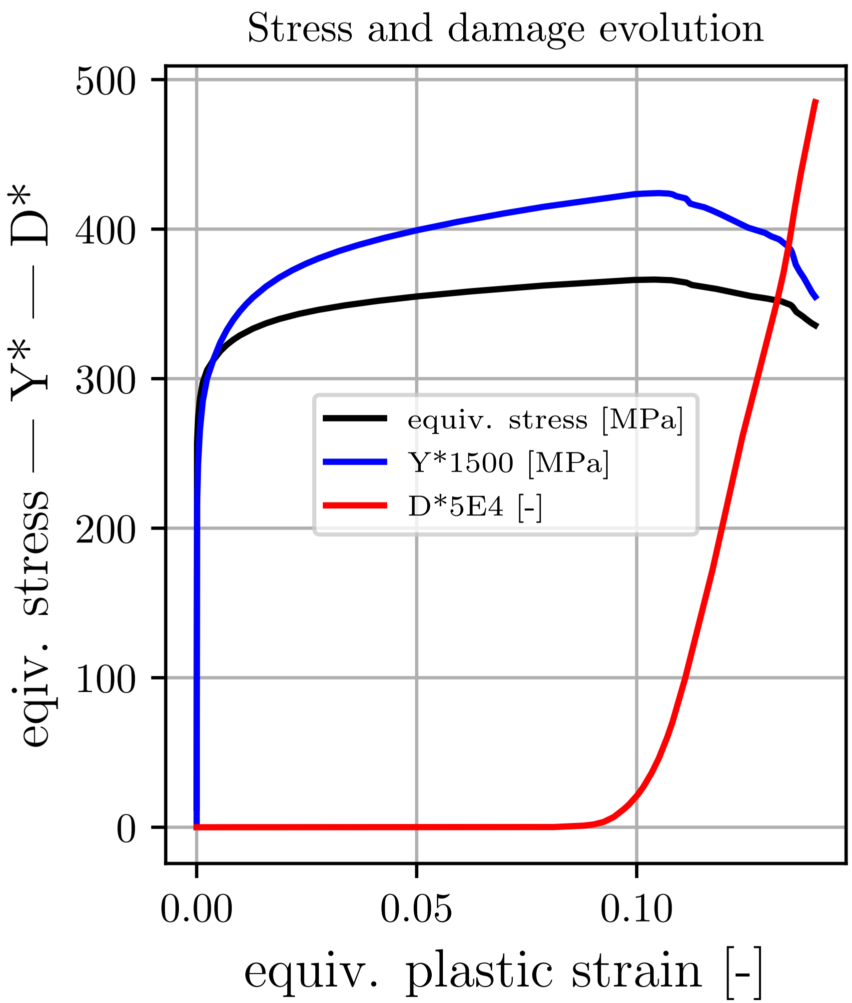
Virtual mechanical testing: Data generation





Generation of training & test data by virtual mechanical tests:
- 5 multiaxial load cases, 1545 data points
- filtering of peaks at UTS
- feature selection for ML algorithms according to established damage models

Comparison with analytical damage models



Chaboche model

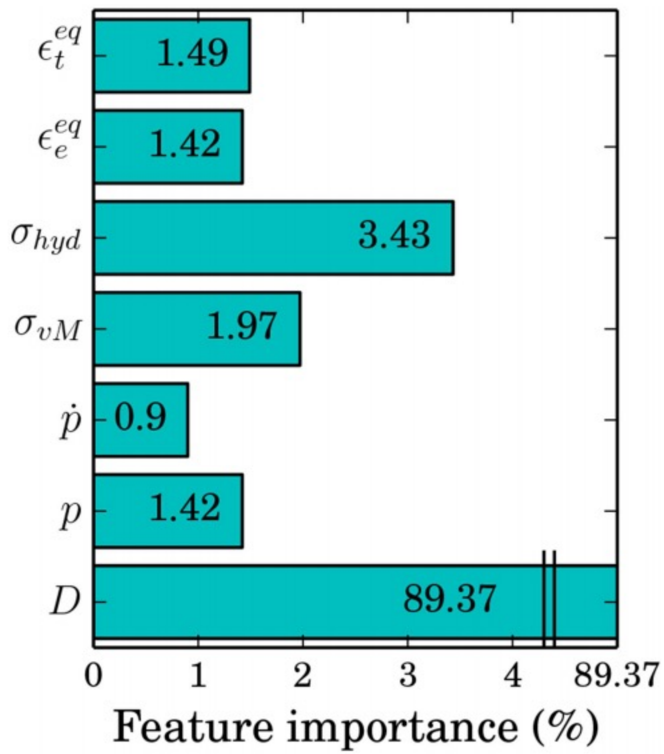
$$\dot{D} = \left(\frac{Y}{S}\right)^{s} \dot{p}$$

$$Y = \frac{{\sigma_{\mathrm{eq}}}^{2}}{2E(1-D)^{2}} \left[\frac{2}{3}(1+\upsilon) + 3(1-2\upsilon)\left(\frac{\sigma_{\mathrm{hyd}}}{\sigma_{\mathrm{eq}}}\right)^{2}\right]$$

Model parameters: *s=250, S=0.29* MPa

Reimann et al. Frontiers in Materials 2019

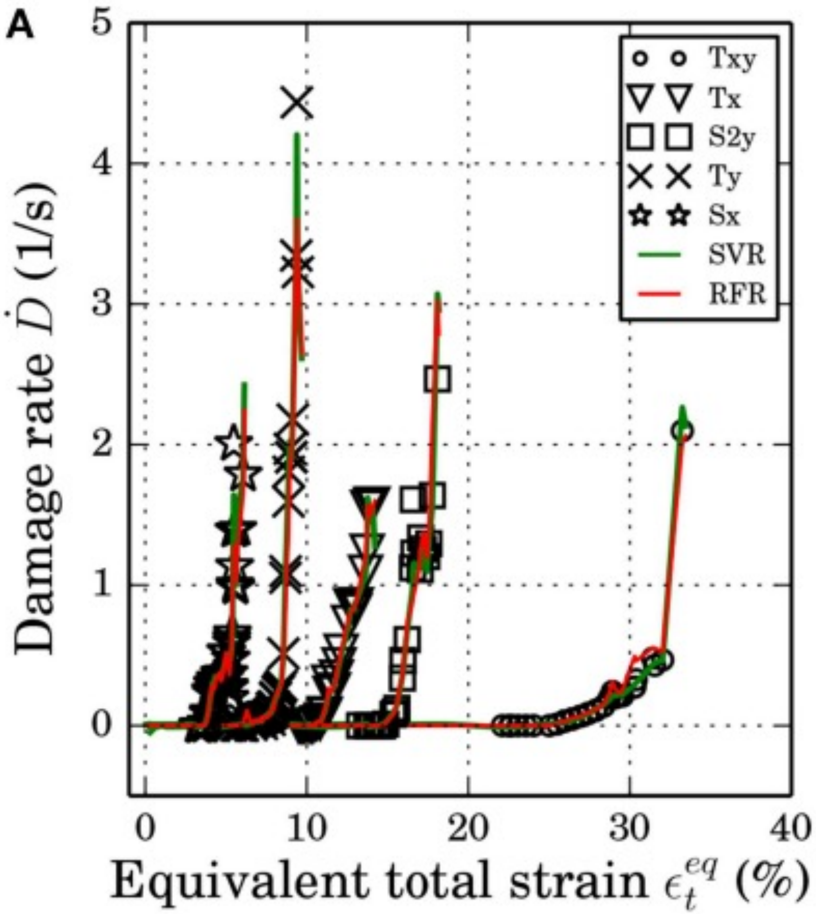
Feature selection

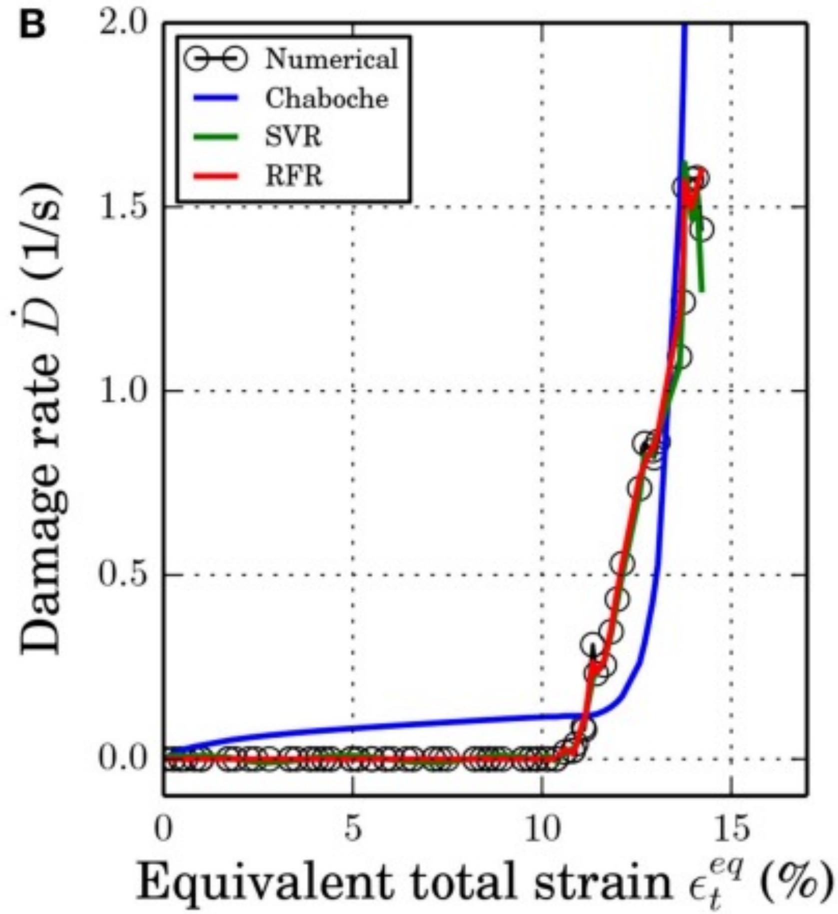


Influence of different mechanical quantities on damage evolution rate.

Selection of features for input vector of ML model.

Training result





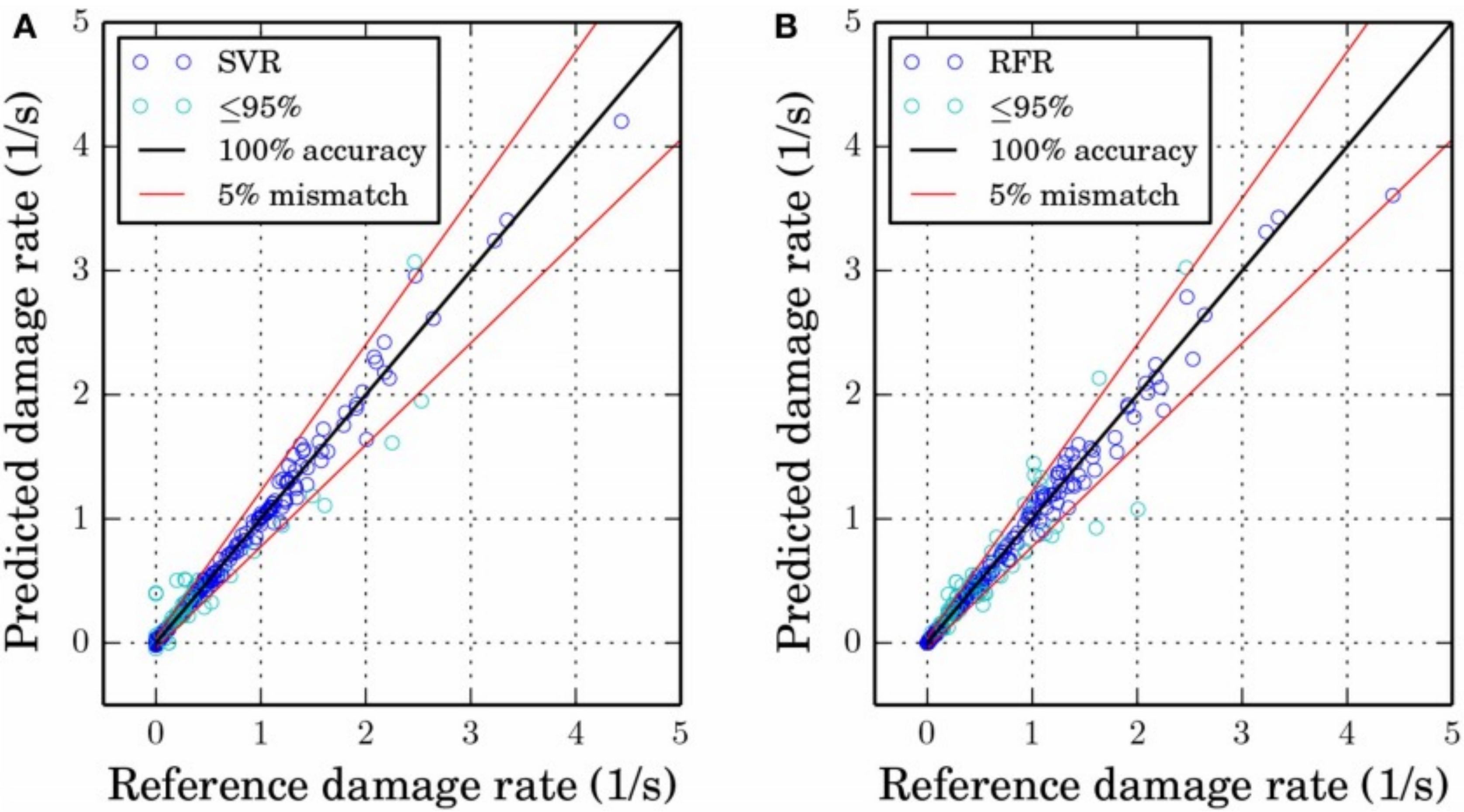
Filtered data for damage rate (removal of sharp peak)

Comparison with Chaboche damage model

Prediction of Damage Rate

Support Vector Regression
Test score: 98.25%

Random Forest Regression
Test score: 97.48%



Summary – Damage modeling

- ML models can be trained to serve as macroscopic damage models
- Features can be selected such that essential physics covered in established damage models is represented correctly
- ML models exhibit a higher versality than mathematical damage models and can be trained with microstructure sensitive data

Co-authors: D. Reimann, K. Nidadavolu, H. ul Hassan, N. Vajragupta, T. Glasmachers, P. Junker

published in Reimann et al. Frontiers in Materials 6 (2019) 181
doi: 10.3389/fmats.2019.00181

Solid mechanics

Load case
Boundary conditions
Mechanical equilibrium

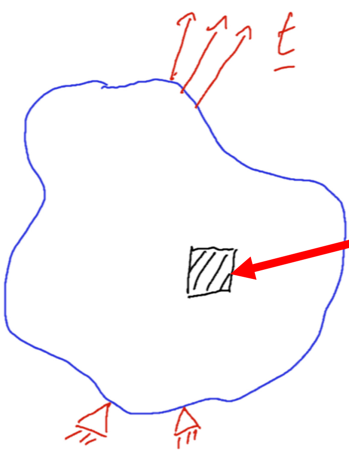
total strain rate $\dot{\boldsymbol{\varepsilon}}_{\text{tot}}$
temperature T
deformation history $\boldsymbol{\varepsilon}_{\text{pl}}$

Constitutive models
History dependent evolution equations for state variables

$\dot{\boldsymbol{\sigma}} = \dot{\boldsymbol{\sigma}}\left(\boldsymbol{\sigma}, \boldsymbol{\epsilon}_{\text{pl}}, \dot{\boldsymbol{\varepsilon}}_{\text{tot}}, D, T\right)$
$\dot{\boldsymbol{\varepsilon}}_{\text{pl}} = \dot{\boldsymbol{\varepsilon}}_{\text{pl}}\left(\boldsymbol{\sigma}, \boldsymbol{\varepsilon}_{\text{pl}}, \dot{\boldsymbol{\varepsilon}}_{\text{tot}}, D, T\right)$
$\dot{D} = \dot{D}\left(\boldsymbol{\sigma}, \boldsymbol{\varepsilon}_{\text{pl}}, \dot{\boldsymbol{\varepsilon}}_{\text{tot}}, D, T\right)$

Material response
Time dependent state variables

stress tensor $\boldsymbol{\sigma} = \boldsymbol{\sigma}(t)$
plastic strain $\boldsymbol{\varepsilon}_{\text{pl}} = \boldsymbol{\varepsilon}_{\text{pl}}(\text{t})$
damage $D = D(t)$



Material parameters
Material specific input to constitutive models

elastic tensor C_{ijkl}
yield strength σ_y
work hardening rate $\partial\boldsymbol{\sigma}/\partial\boldsymbol{\varepsilon}_{\text{pl}}$
damage onset $\varepsilon_{\text{pl}}^{\text{crit}}$

- Solutions involve rather intricate mathematical formalism
- Explicit consideration of microstructure is difficult

Solid mechanics

Load case
Boundary conditions
Mechanical equilibrium

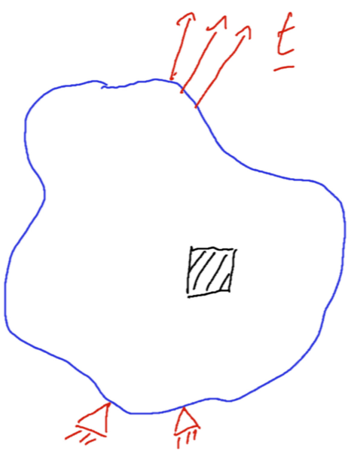
total strain rate $\dot{\boldsymbol{\varepsilon}}_{\text{tot}}$
temperature T
deformation history $\boldsymbol{\varepsilon}_{\text{pl}}$

Constitutive models
History dependent evolution equations for state variables

$\dot{\boldsymbol{\sigma}} = \dot{\boldsymbol{\sigma}}\left(\boldsymbol{\sigma}, \boldsymbol{\epsilon}_{\text{pl}}, \dot{\boldsymbol{\varepsilon}}_{\text{tot}}, D, T\right)$
$\dot{\boldsymbol{\varepsilon}}_{\text{pl}} = \dot{\boldsymbol{\varepsilon}}_{\text{pl}}\left(\boldsymbol{\sigma}, \boldsymbol{\varepsilon}_{\text{pl}}, \dot{\boldsymbol{\varepsilon}}_{\text{tot}}, D, T\right)$
$\dot{D} = \dot{D}\left(\boldsymbol{\sigma}, \boldsymbol{\varepsilon}_{\text{pl}}, \dot{\boldsymbol{\varepsilon}}_{\text{tot}}, D, T\right)$

Material response
Time dependent state variables

stress tensor $\boldsymbol{\sigma} = \boldsymbol{\sigma}(t)$
plastic strain $\boldsymbol{\varepsilon}_{\text{pl}} = \boldsymbol{\varepsilon}_{\text{pl}}(\text{t})$
damage $D = D(t)$



Material parameters
Material specific input to constitutive models

elastic tensor C_{ijkl}
yield strength σ_y
work hardening rate $\partial\boldsymbol{\sigma}/\partial\boldsymbol{\varepsilon}_{\text{pl}}$
damage onset $\varepsilon_{\text{pl}}^{\text{crit}}$

Data-based material representation

- **mechanical behavior**
- **microstructure**
- **texture**

Solid mechanics

Load case
Boundary conditions
Mechanical equilibrium

total strain rate $\dot{\boldsymbol{\varepsilon}}_{\mathrm{tot}}$
temperature T
deformation history $\boldsymbol{\varepsilon}_{\mathrm{pl}}$

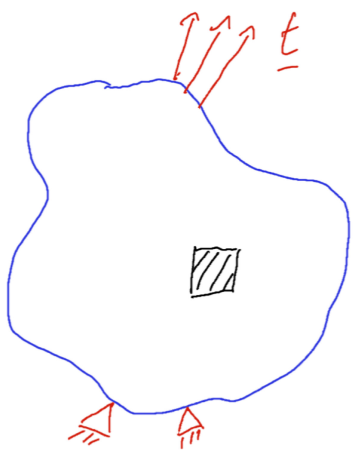
Data-based material representation

- mechanical behavior
- microstructure
- texture

Compatibility with standard finite element solvers.

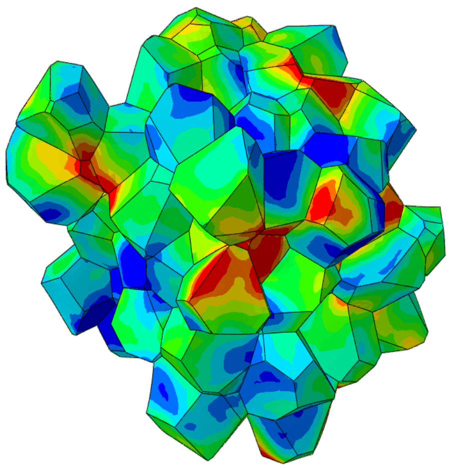
Material response
Time dependent state variables

stress tensor $\boldsymbol{\sigma} = \boldsymbol{\sigma}(t)$
plastic strain $\boldsymbol{\varepsilon}_{\mathrm{pl}} = \boldsymbol{\varepsilon}_{\mathrm{pl}}(\mathrm{t})$
damage $D = D(t)$

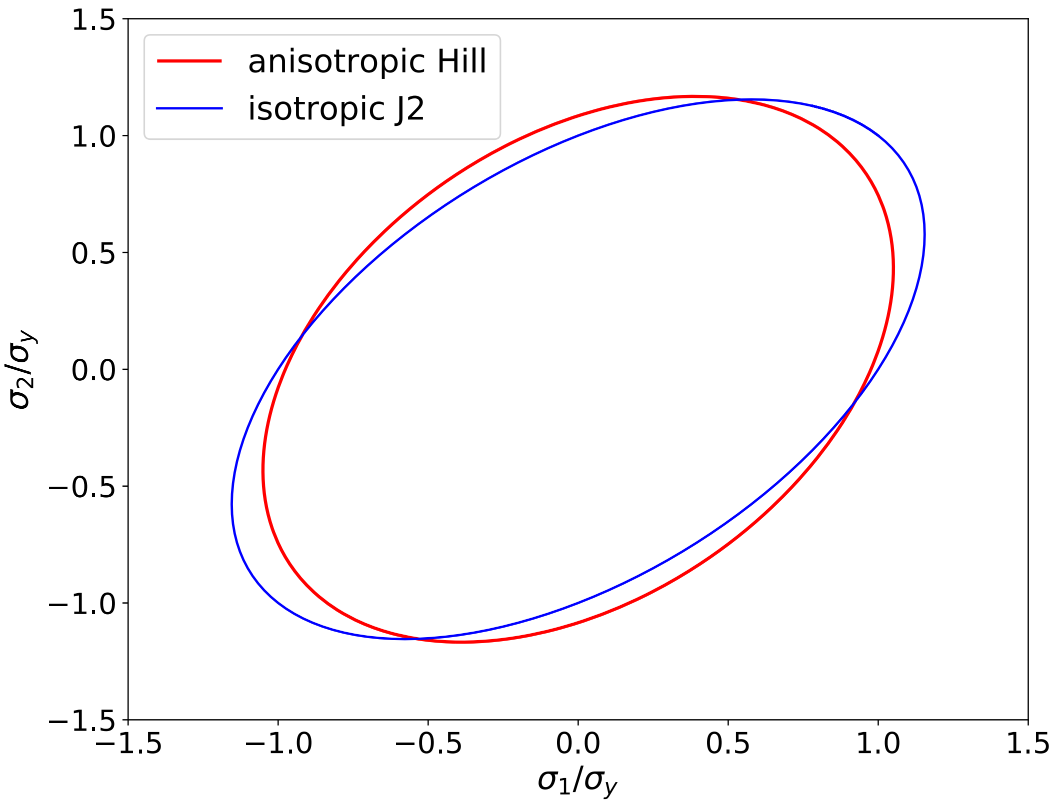


Data-generation

validated
micromechanical models



Continuum plasticity



Yield loci of isotropic J2 and anisotropic Hill material definitions in cross-section of principle stress space

Yield function

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

Yield locus $f(\boldsymbol{\sigma}) = 0$

- elasticity inside $f(\boldsymbol{\sigma}) < 0$
- plasticity on yield locus
- material does not support stresses outside yield locus

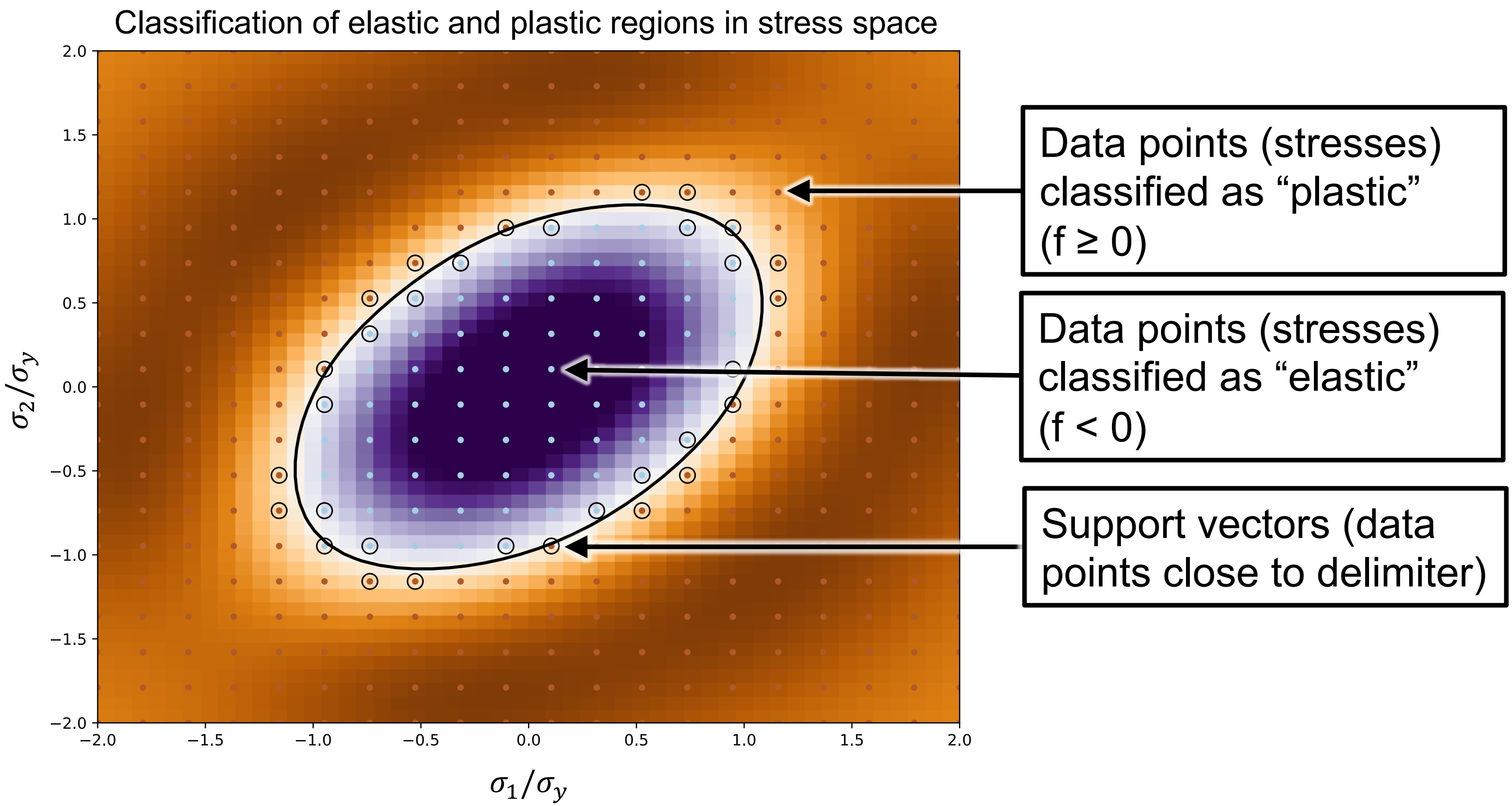
Plastic strain rate (flow rule)

$$\dot{\boldsymbol{\varepsilon}}_{\mathrm{pl}} = \dot{\lambda}\boldsymbol{n}$$

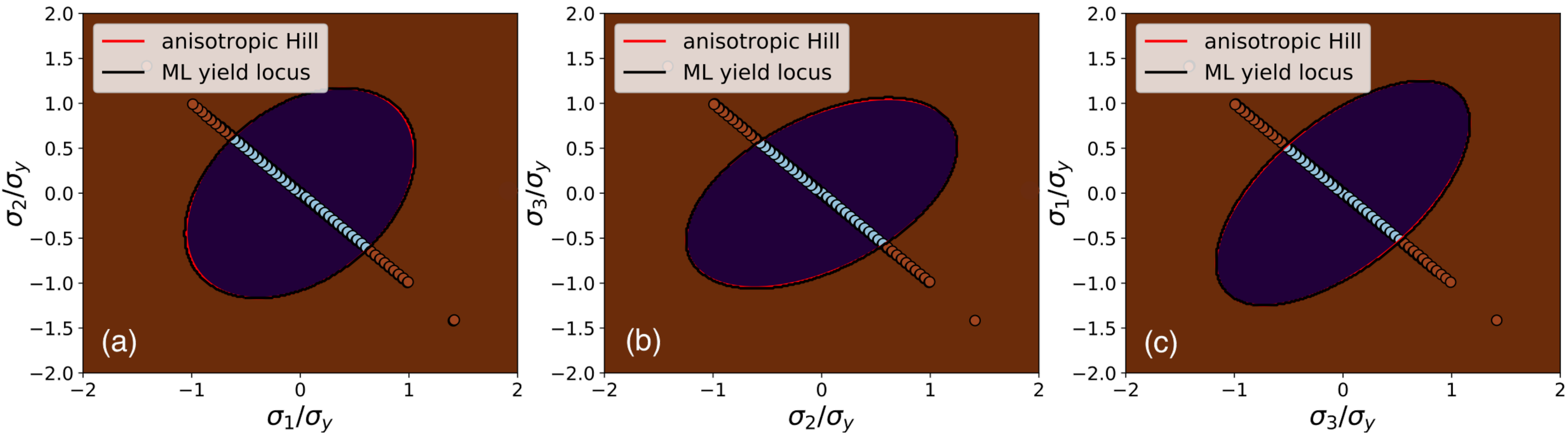
- $\dot{\lambda}$: plastic strain multiplier obtained from return mapping algorithm
- $\boldsymbol{n}$: normal to yield locus

Data-based model for plastic yielding

Trained Support Vector Classification



Machine Learning Flow Rule

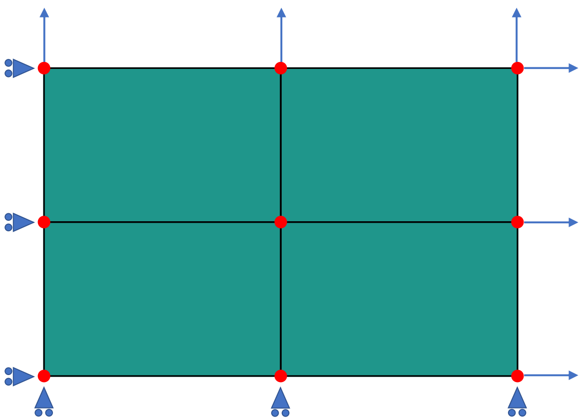


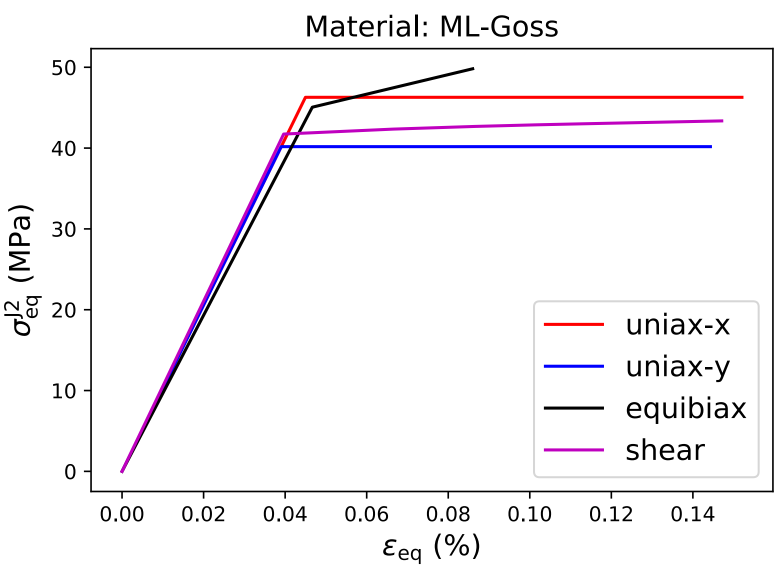
Support Vector Classification

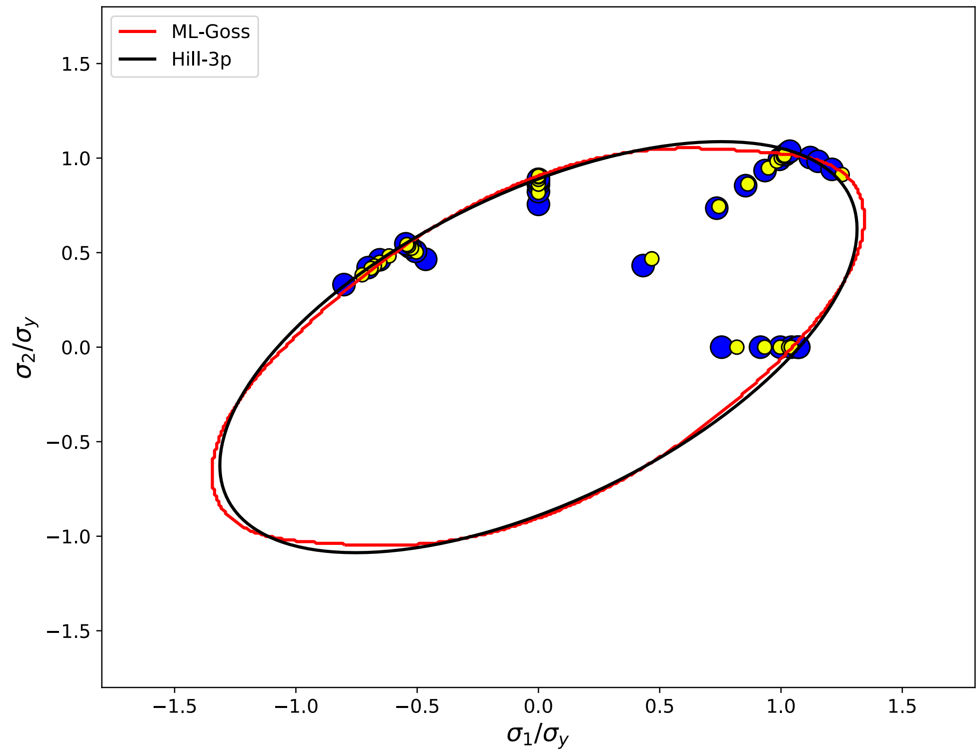
- determines elastic/plastic regions in principle stress space
- delimiter line is yield locus
- formulation of data-oriented flow rule
- normal $\boldsymbol{n}$ can be calculated analytically from SVC formulation
- only training data from deviatoric load cases required (reduction to 2d stress space)

A. Hartmaier Materials 2020

Application of ML Flow Rule



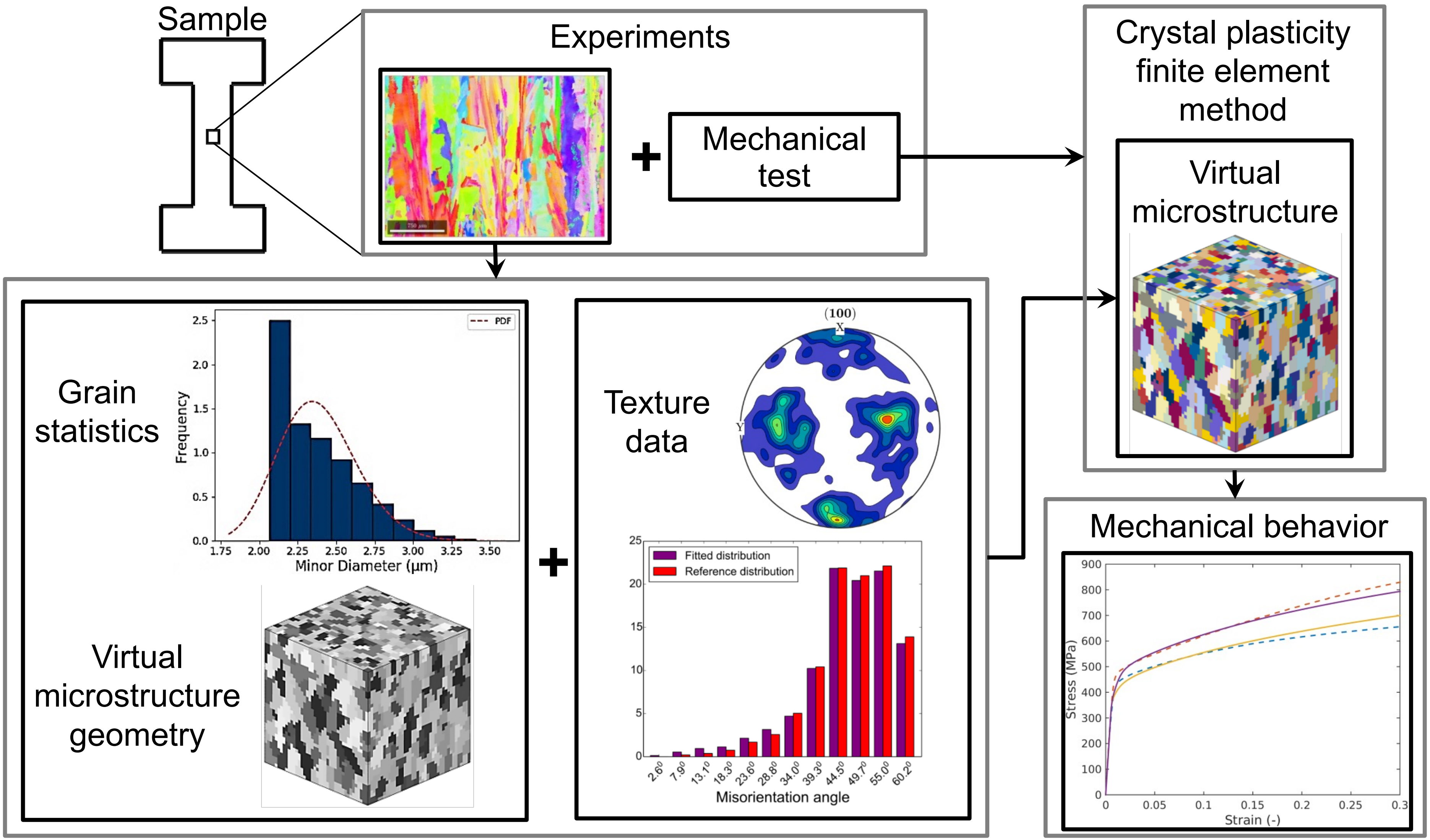




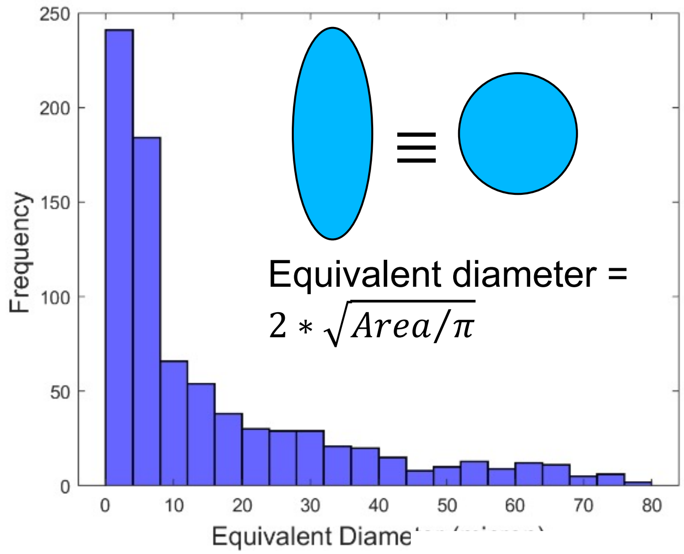
Application of ML flow rule in simple FE model

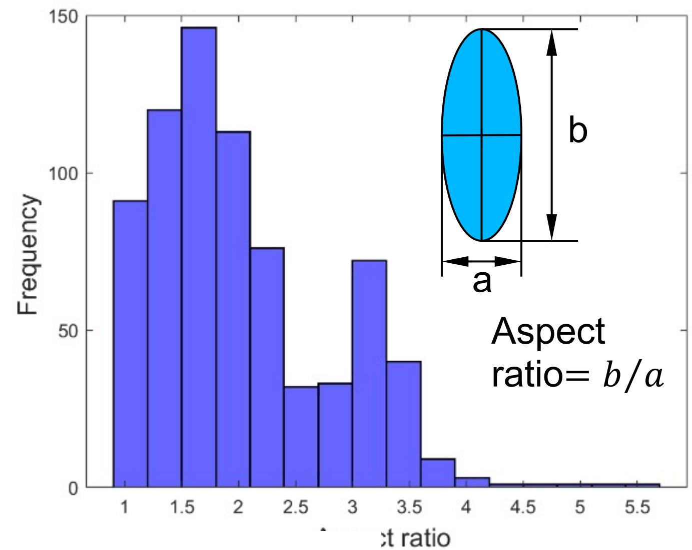
- Comparison to Hill-3p flow rule
- Stress evolution in principle stress space
- Stress strain curves

Data Generation by Micromechanical Model

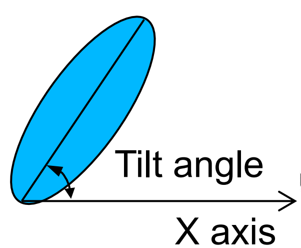


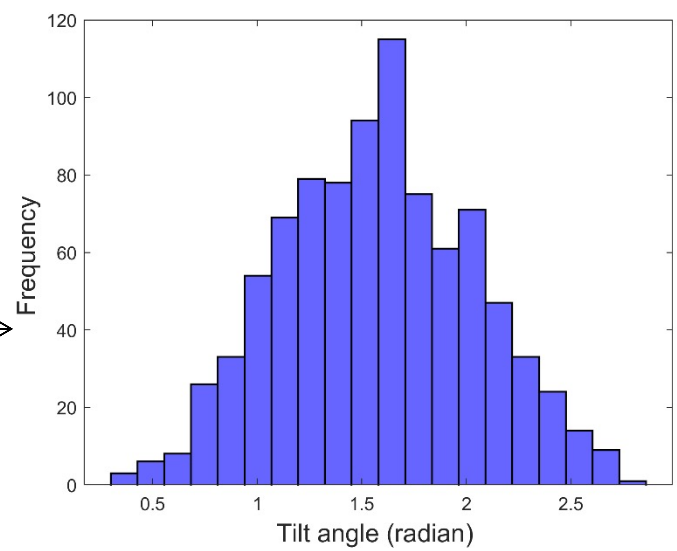
Micromechanical Model: Grain size & shape

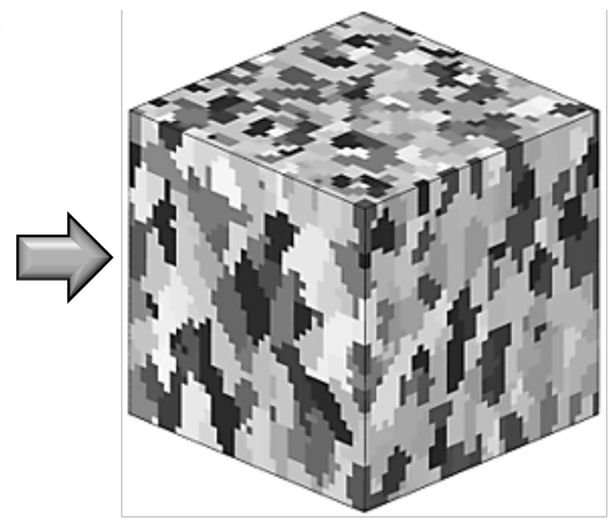




Prasad et al., The Journal of open Source Software, 2019



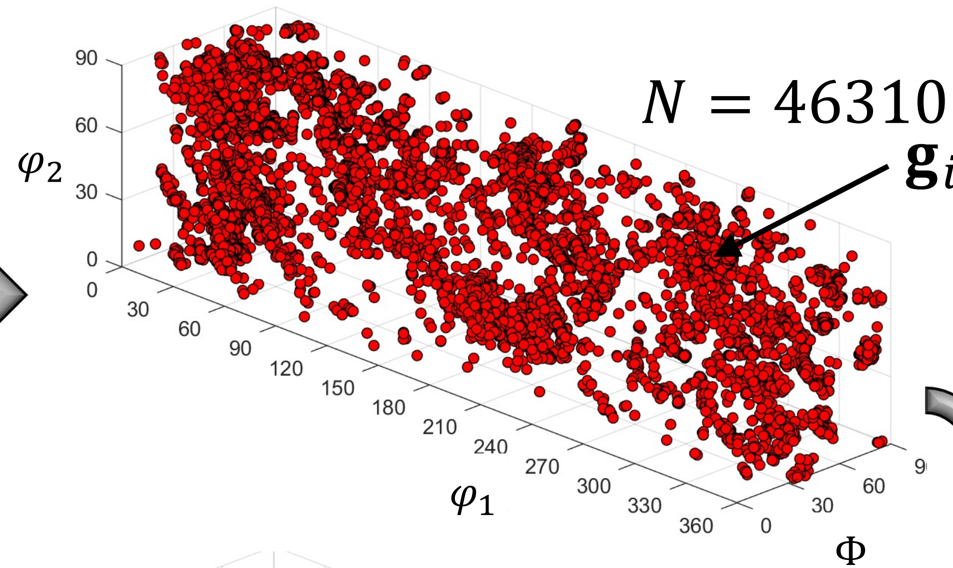


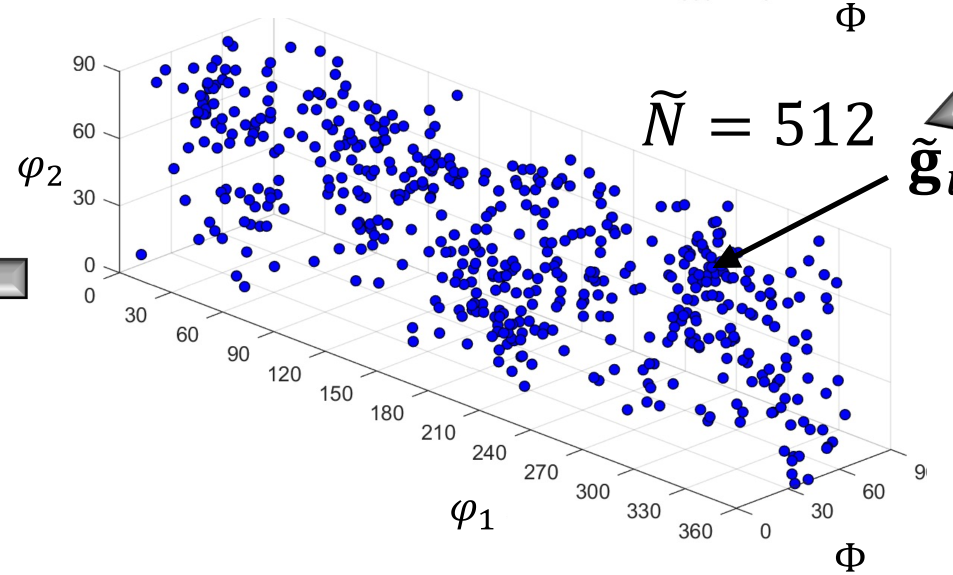


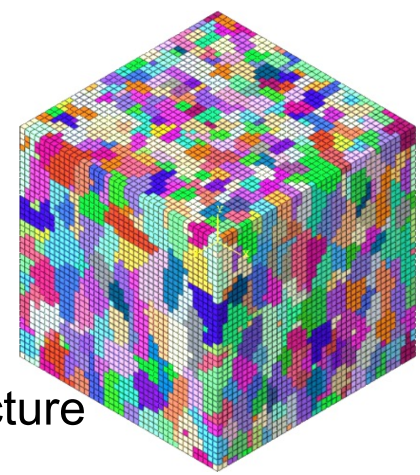
Kanapy: https://github.com/mrgprasad/kanapy

Micromechanical Model: Texture

EBSD

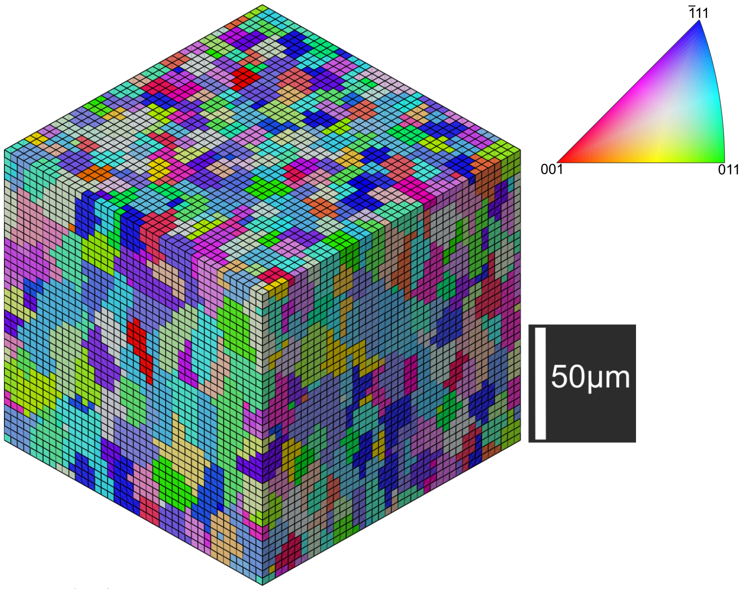


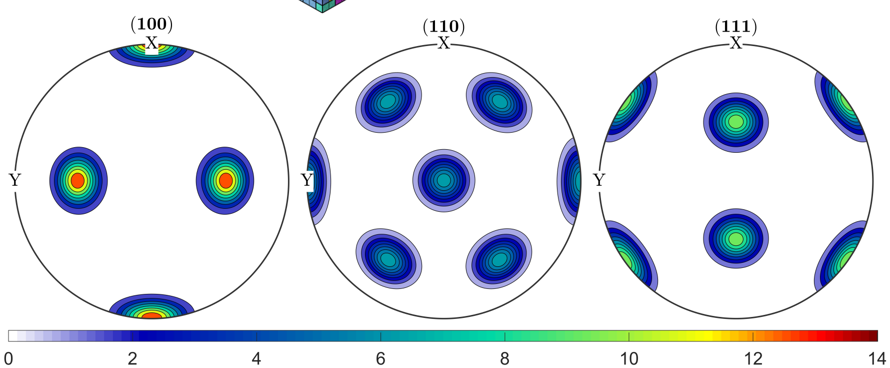




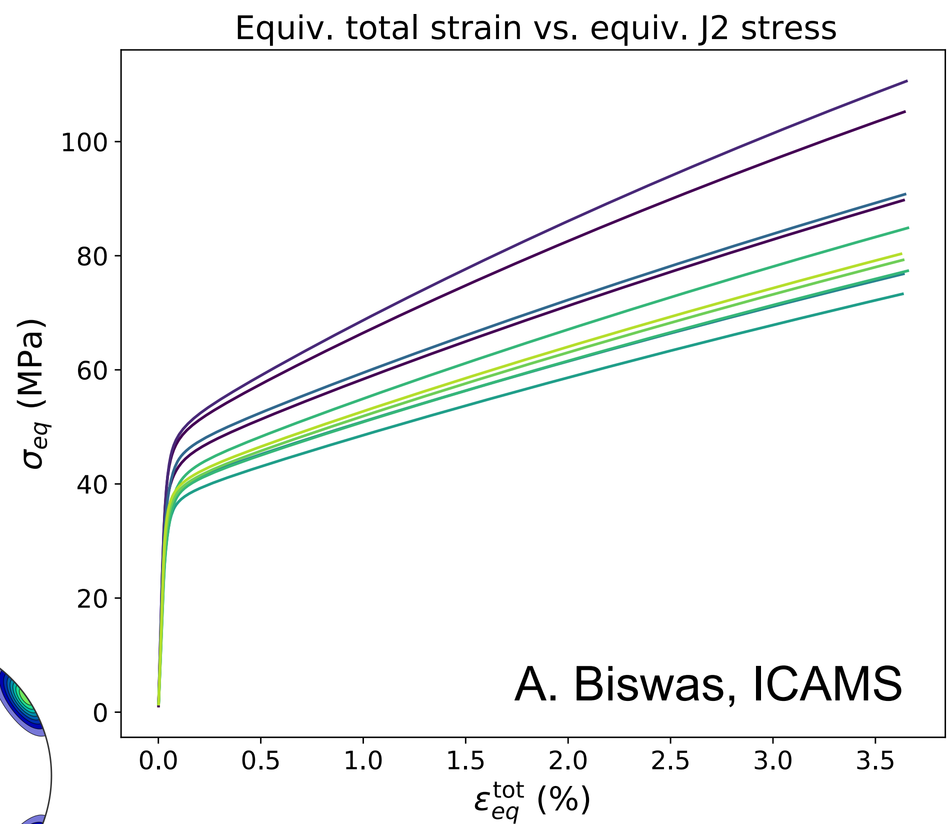
Virtual microstructure

Micromechanical Model: Property prediction





Micromechanical model with ~2,200 grains and Goss texture

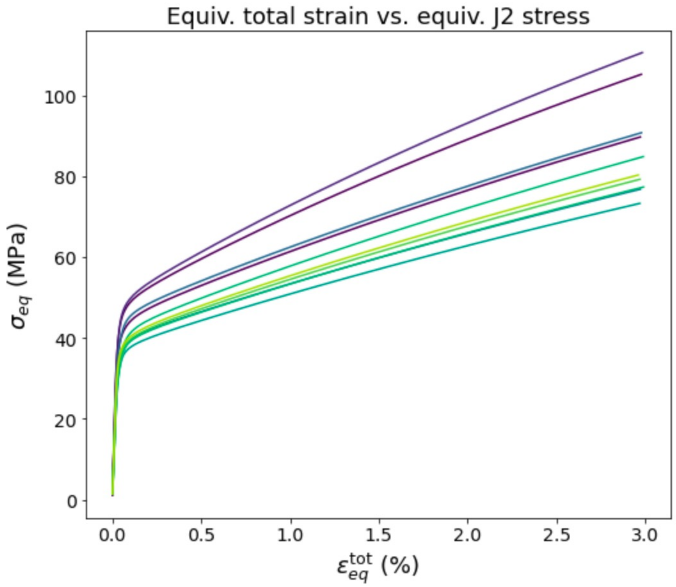


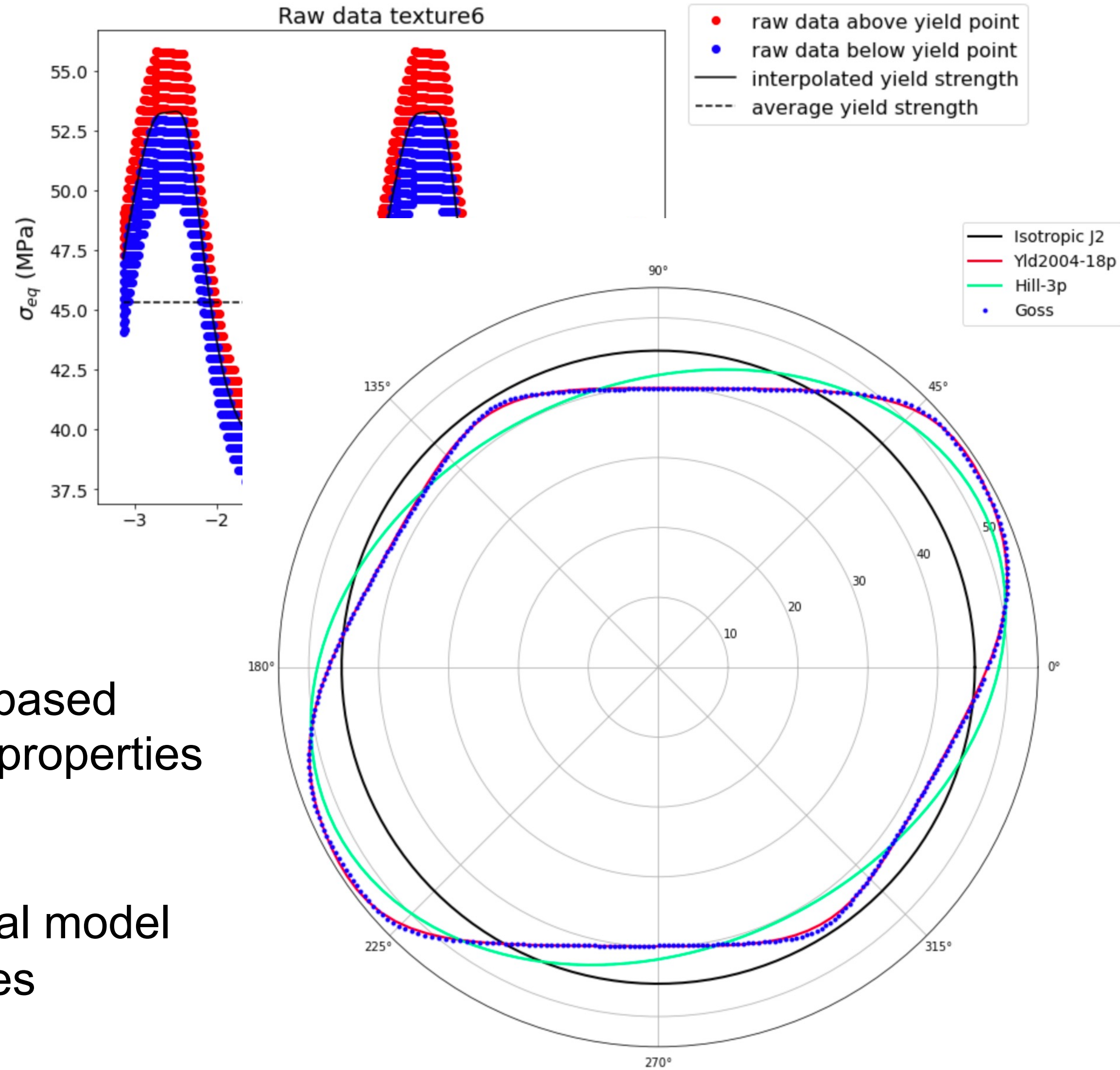
Resulting stress-strain curves

- Crystal plasticity model
- Different deviatoric load cases

Anisotropic yield strength → data

Micromechanical Model: data for material description



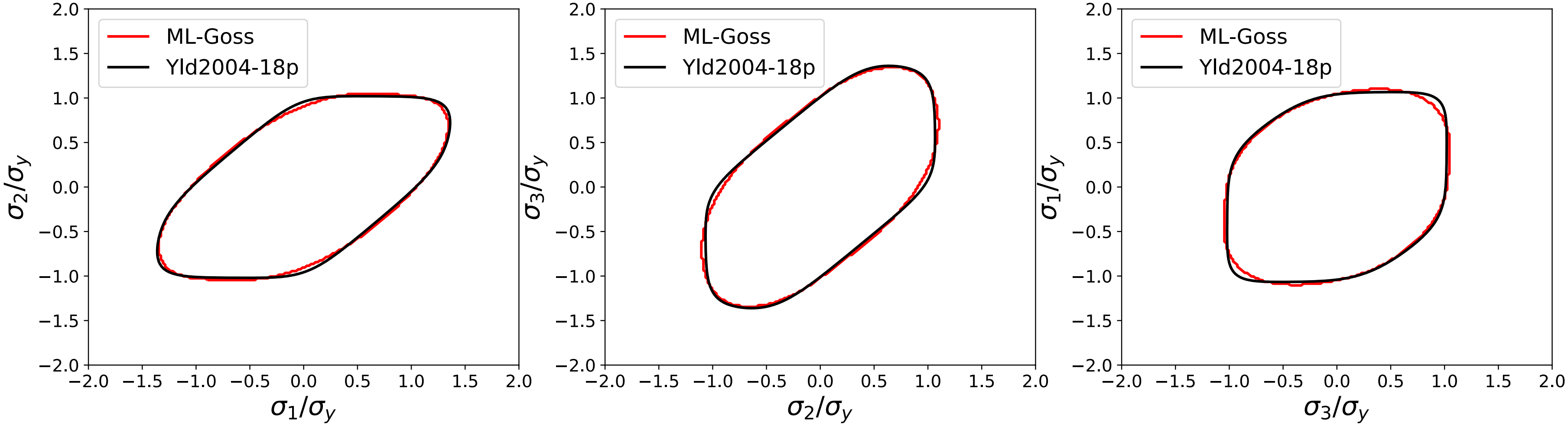


Microstructure sensitive data-based description of plastic material properties

Experimental data

- validation of micromechanical model
- hybrid mechanical data bases

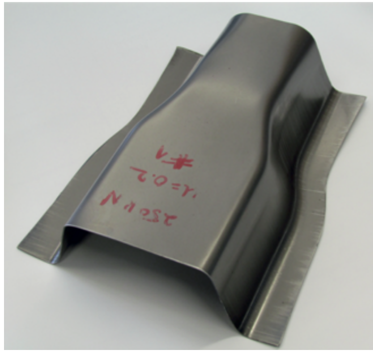
Trained Machine Learning Flow Rule

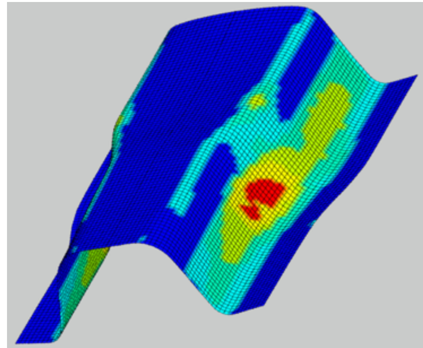


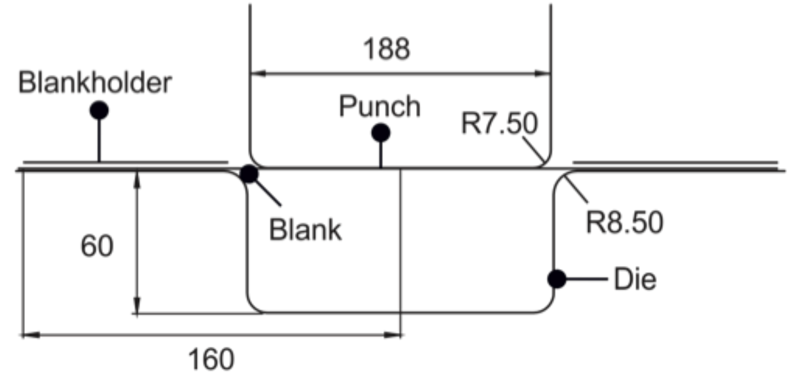
Yield loci in different cross-sections of principle stress space

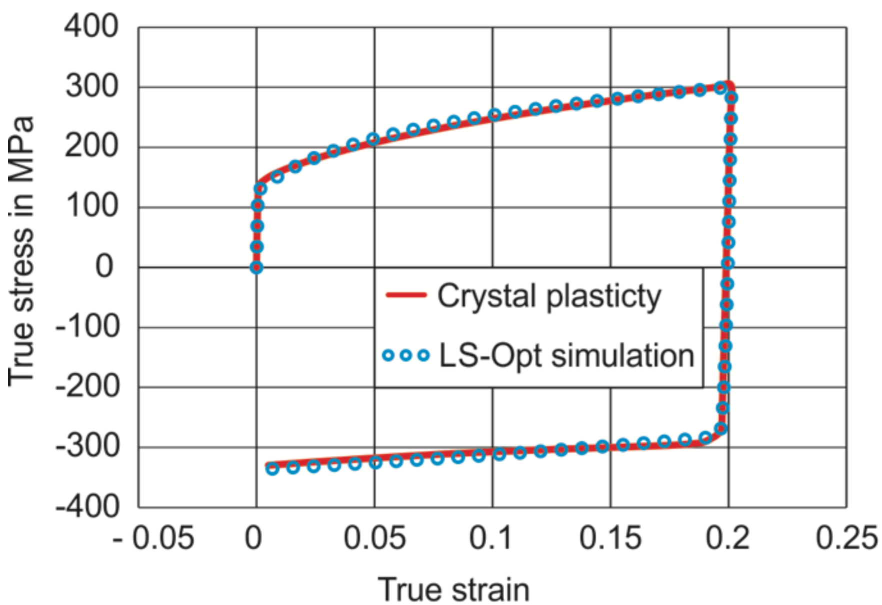
- Mechanical data obtained from micromechanical model
- Training of Machine Learning flow rule
- Barlat Yld2004-18p yield function fitted to same data

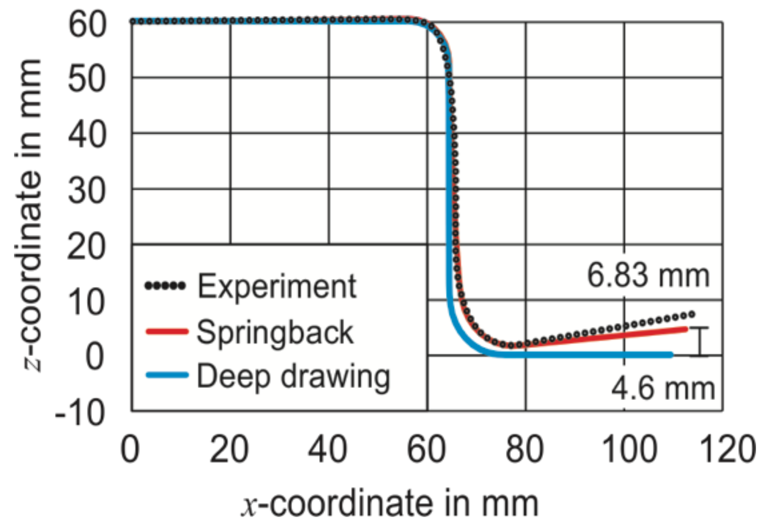
Outlook: Application to component behavior











Fitting of Yoshida-Uemori model to RVE simulations

Vajragupta, Hassan & Hartmaier, Procedia Engineering, 2017

Summary

- **Data-oriented material descriptions based on ML models** can replace classical constitutive rules and their parameters in finite element modeling
  ➞ Advantage: **Consideration of microstructure** is possible
- Micromechanical modeling (synthetic RVEs, crystal plasticity, damage) is a powerful tool to generate **data for microstructure-property relationships**
- Fully parameterized and validated micromechanical models can complement experimental data to **hybrid mechanical data**