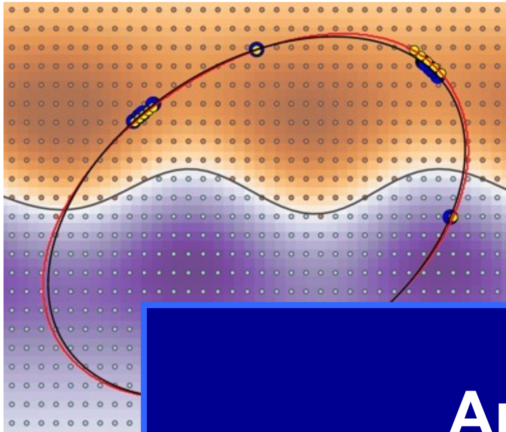


2021/06/25

Applications of Machine Learning to Mechanical Systems

A. Hartmaier

Interdisciplinary Centre for Advanced Materials Simulation (ICAMS)

Ruhr-Universität Bochum

alexander.hartmaier@rub.de

INTERDISCIPLINARY CENTRE FOR ADVANCED MATERIALS SIMULATION

RUHR UNIVERSITÄT BOCHUM

Outline

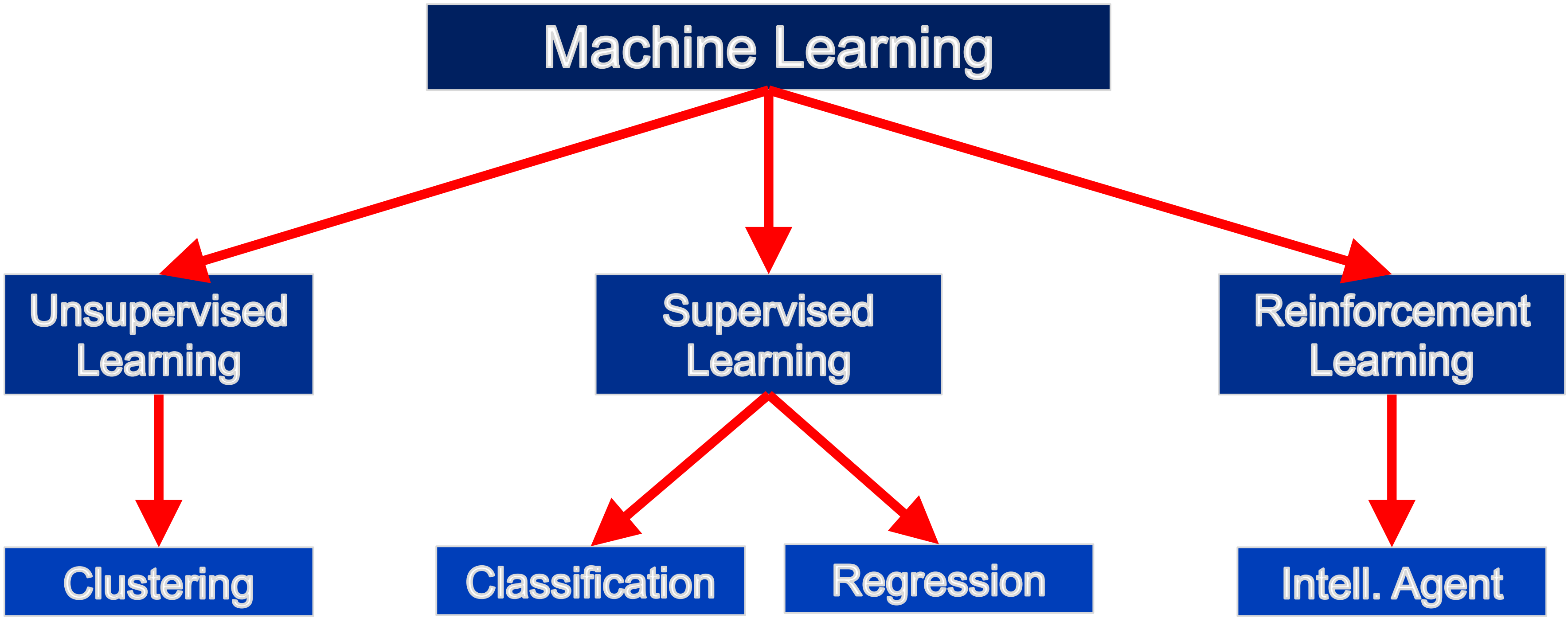
I. Lecture
  1. Machine Learning (ML) methods
  2. Application as surrogate model
  3. Regression models for damage homogenization
  4. Classification problem in continuum plasticity

II. Jupyter notebooks
  1. ML Tutorial Regression
  2. ML Tutorial Classification
  3. Surrogate model for indentation

  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 (ML)



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

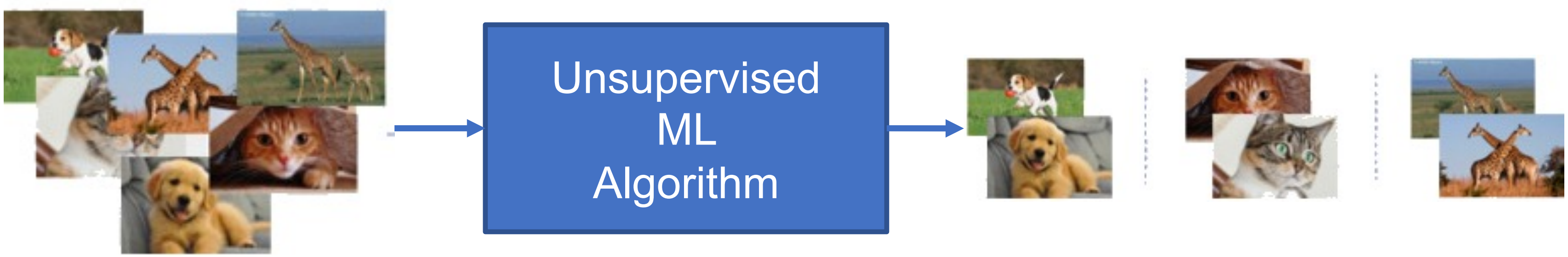
Reinforcement Learning: Intelligent Agents

Task: Create a computer code that can play "Go"

Source: Wikipedia
Wikimedia Commons, CC BY-SA 3.0

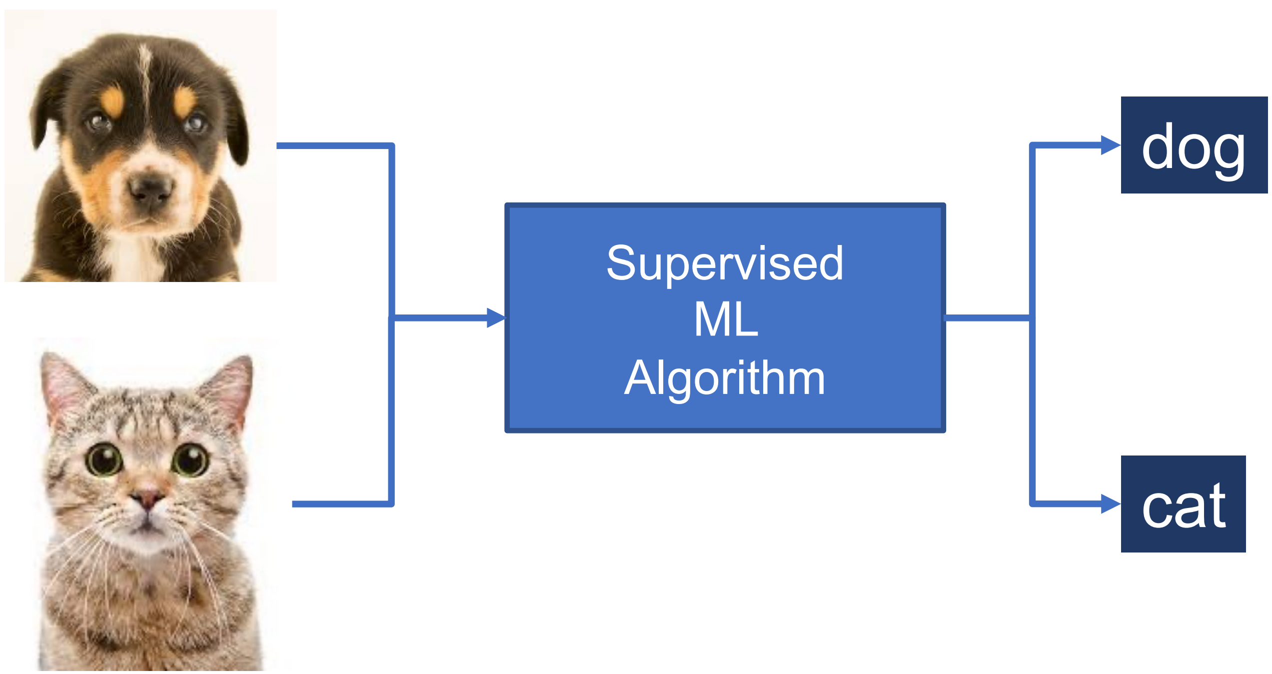
Unsupervised Learning: Clustering

Task: Sort pictures of same animals into groups (clustering)



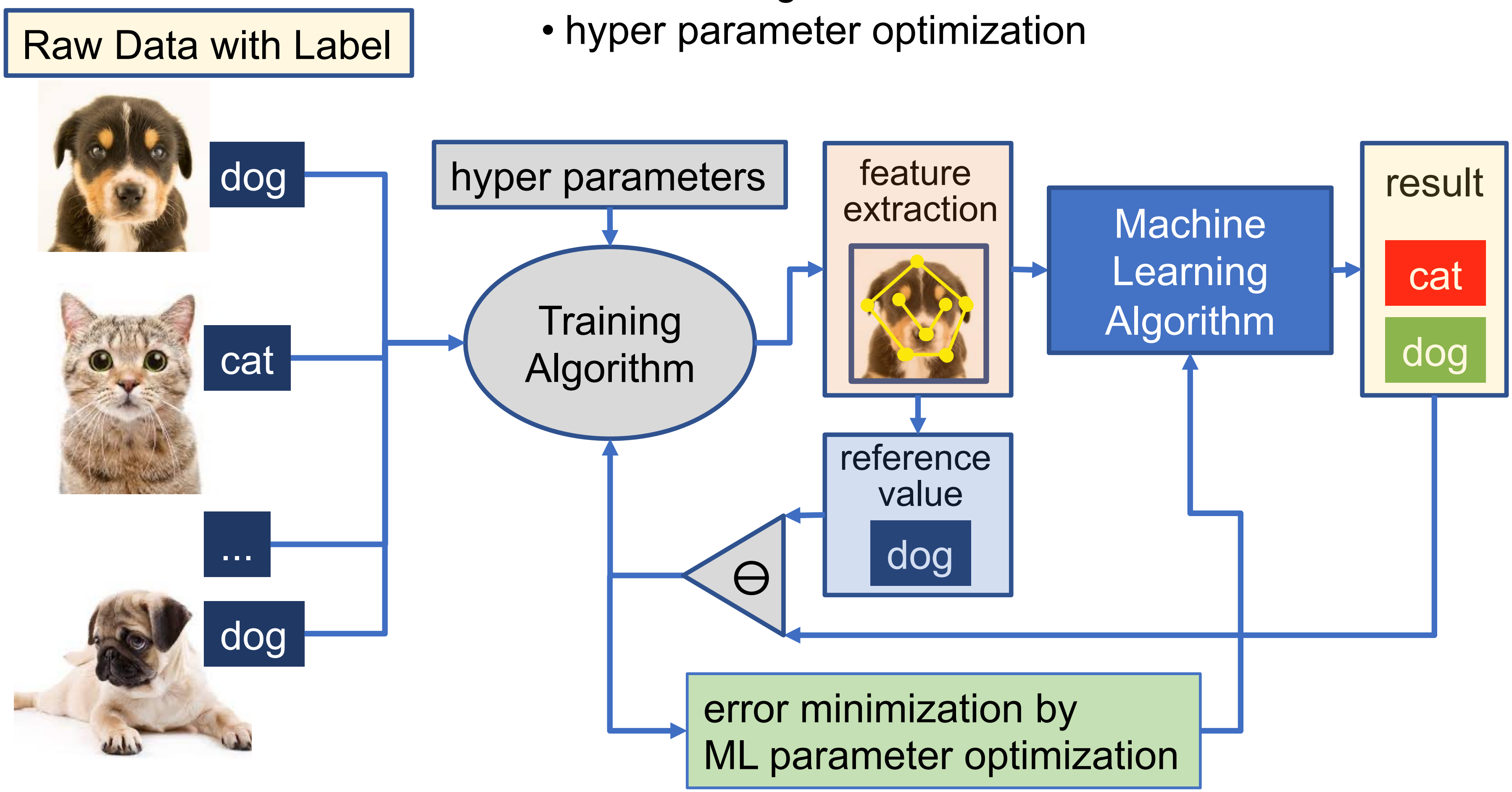
Supervised Learning: Classification

Task: Identify pictures of cats and dogs (classification)



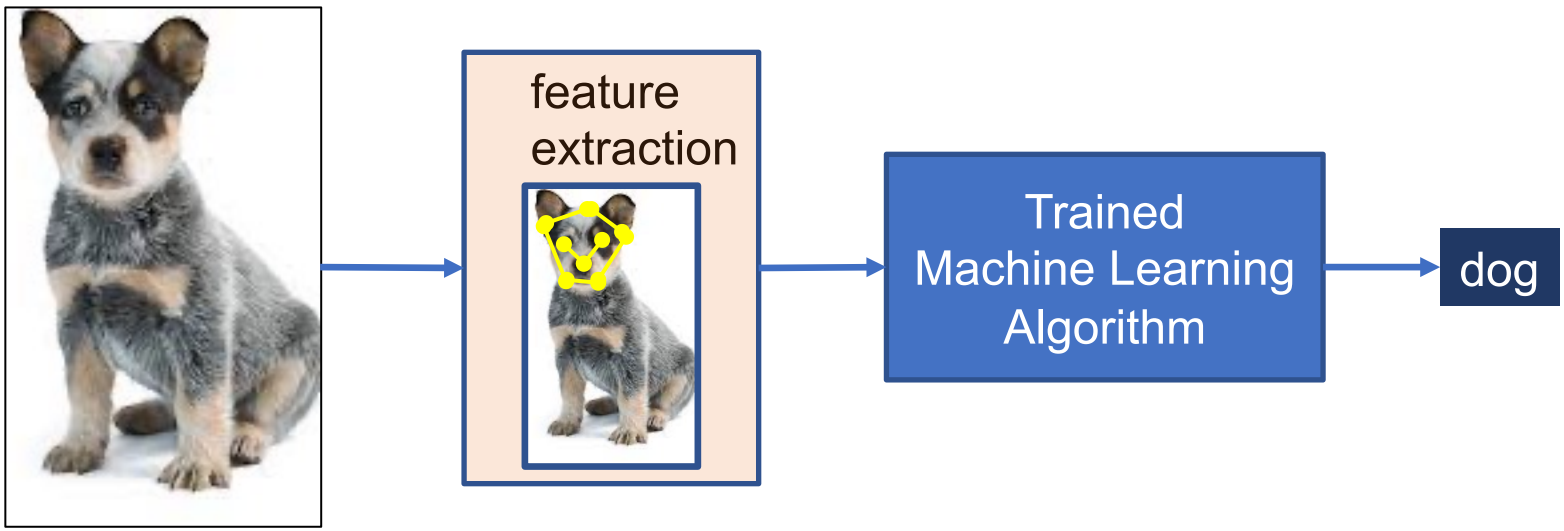
Supervised Learning: Training of ML Model

- choice of ML algorithm
- feature design
- hyper parameter optimization



Supervised Learning: Validation

Validation with unseen data



Supervised Learning: Feature Extraction

feature extraction

Automated feature extraction from raw data:
- convolutions of raw data (CNN)
- autoencoder
- Fast Fourier Transform
- N-point-statistics/auto correlation
- Principle Component Analysis
- Singular Value Decomposition

Feature extraction based on domain knowledge:
- extraction of physical quantities
- correlation analysis
- experience with similar tasks

Supervised Learning: Regression

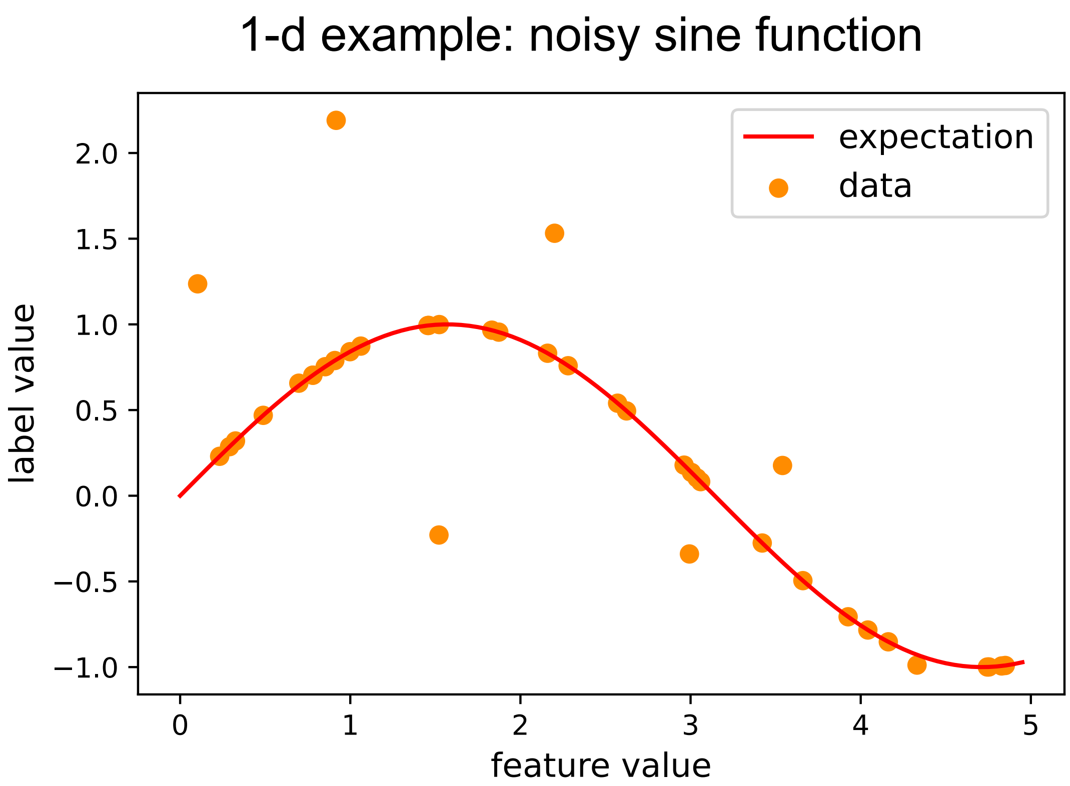
input vector
“features”

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

output vector
“label” / result

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

Supervised Learning: Regression

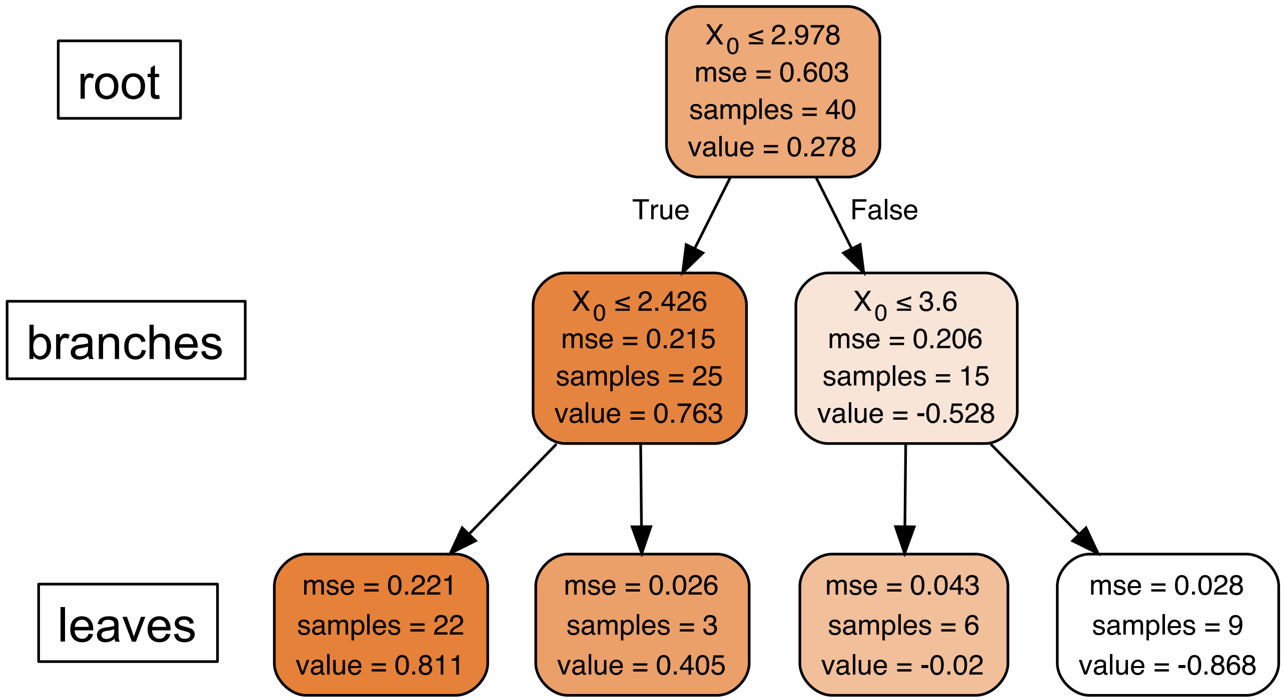


$$y = f(x) = \sin x$$

$$0 \le x \le 5$$

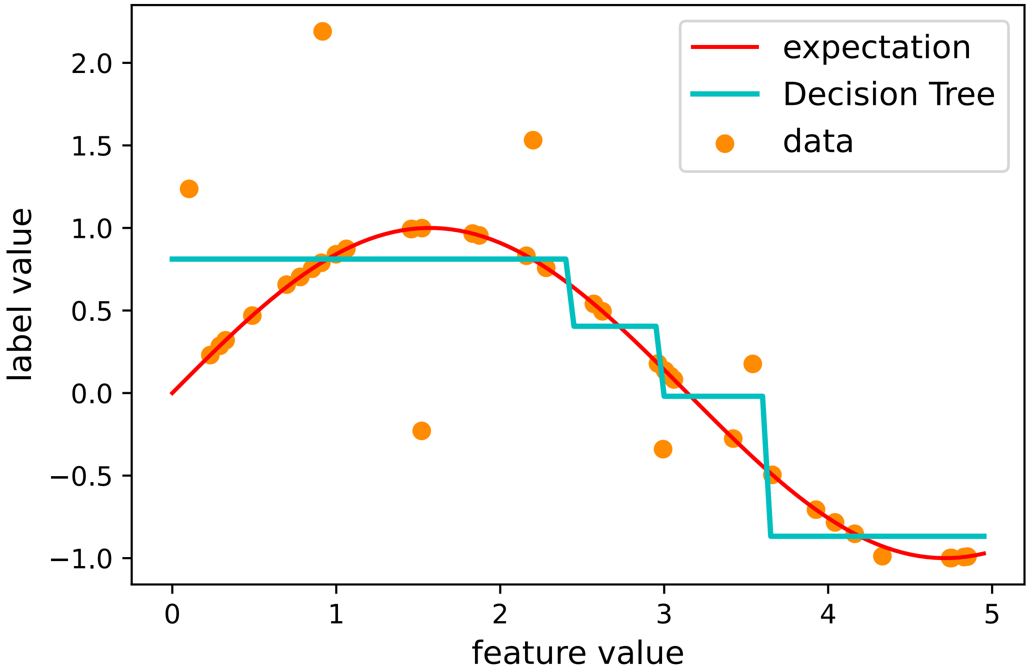
Decision Tree Regression

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



Supervised Learning: Decision Tree Regression

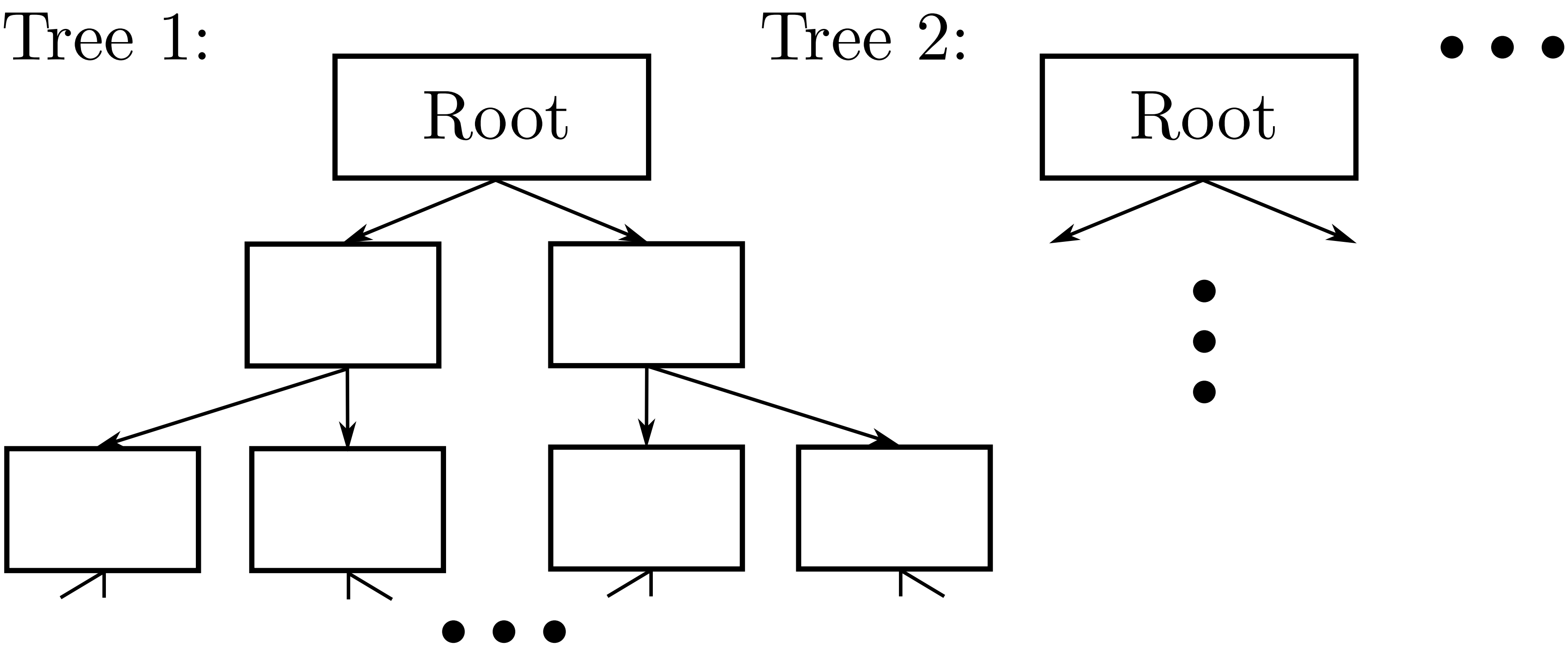
1-d example: noisy sine function



Decision Tree
hyper parameters:
depth = 2

$$y = f(x) = \begin{cases} 0.811 & \text{if} \quad x \leq 2.426 \\ 0.405 & \text{if} \quad 2.426 < x \leq 2.978 \\ -0.02 & \text{if} \quad 2.978 < x \leq 3.6 \\ -0.868 & \text{if} \quad x > 3.6 \end{cases}$$

Random Forest Regression

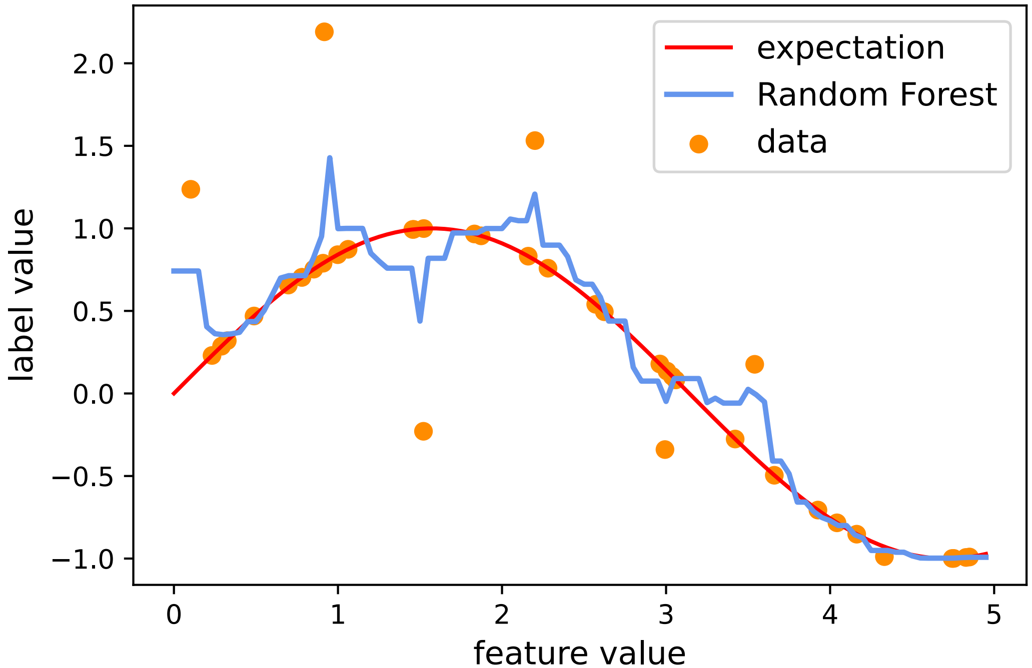


Goal: Create model that predicts output value for given input data by learning simple decision rules

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

Supervised Learning: Random Forest Regression

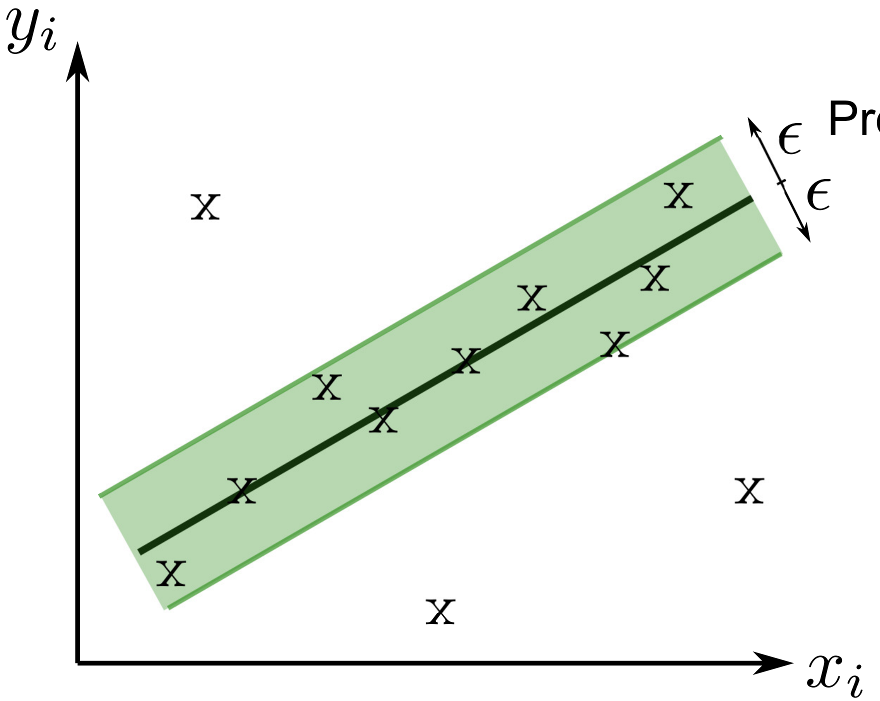
1-d example: noisy sine function



Random Forest
hyper parameters:
max. depth = 5
N_{tree} = 20

$$y = f(x) = w \sum_{i=1}^{N} f_{DT}^{(i)}(x)\,, \qquad w = \frac{1}{N_c}$$

Support Vector Machine (Regression/Classification)



Precision ϵ = 1%

SVM function:

$$f(x) = \sum_{k=1}^{n} y_k a_k K\left(x_k^{(SV)}, x\right) + \rho$$

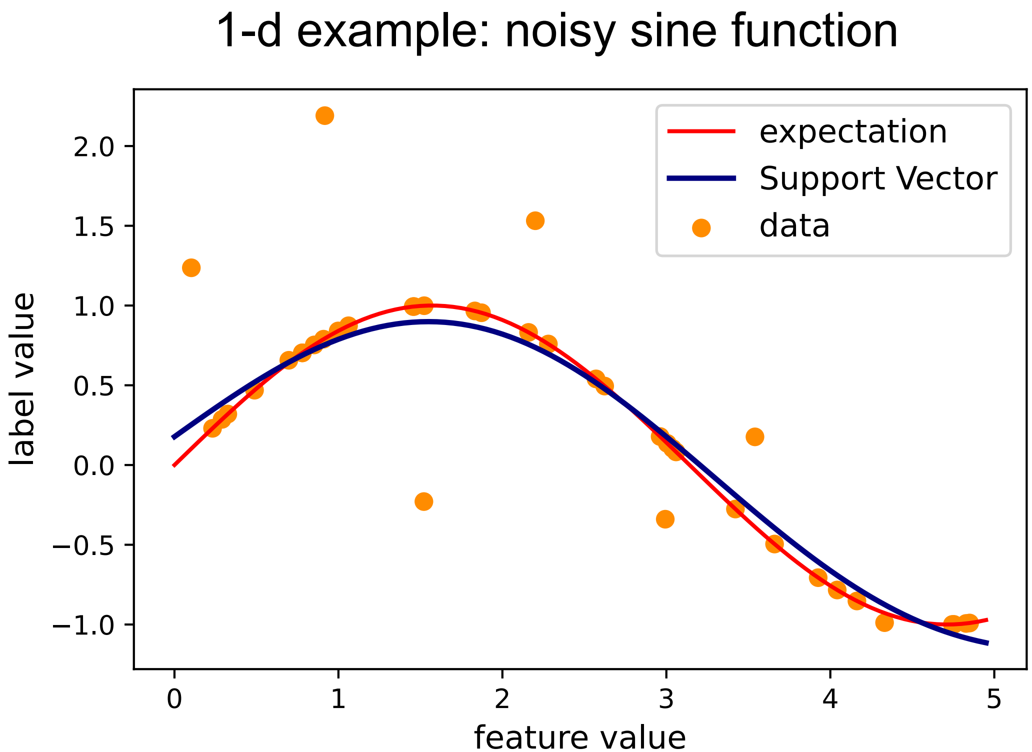
RBF kernel:

$$K\left(x_k^{(SV)}, x\right) = \exp\left(-\gamma \left\| x - x_k^{(SV)} \right\|^2\right)$$

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)

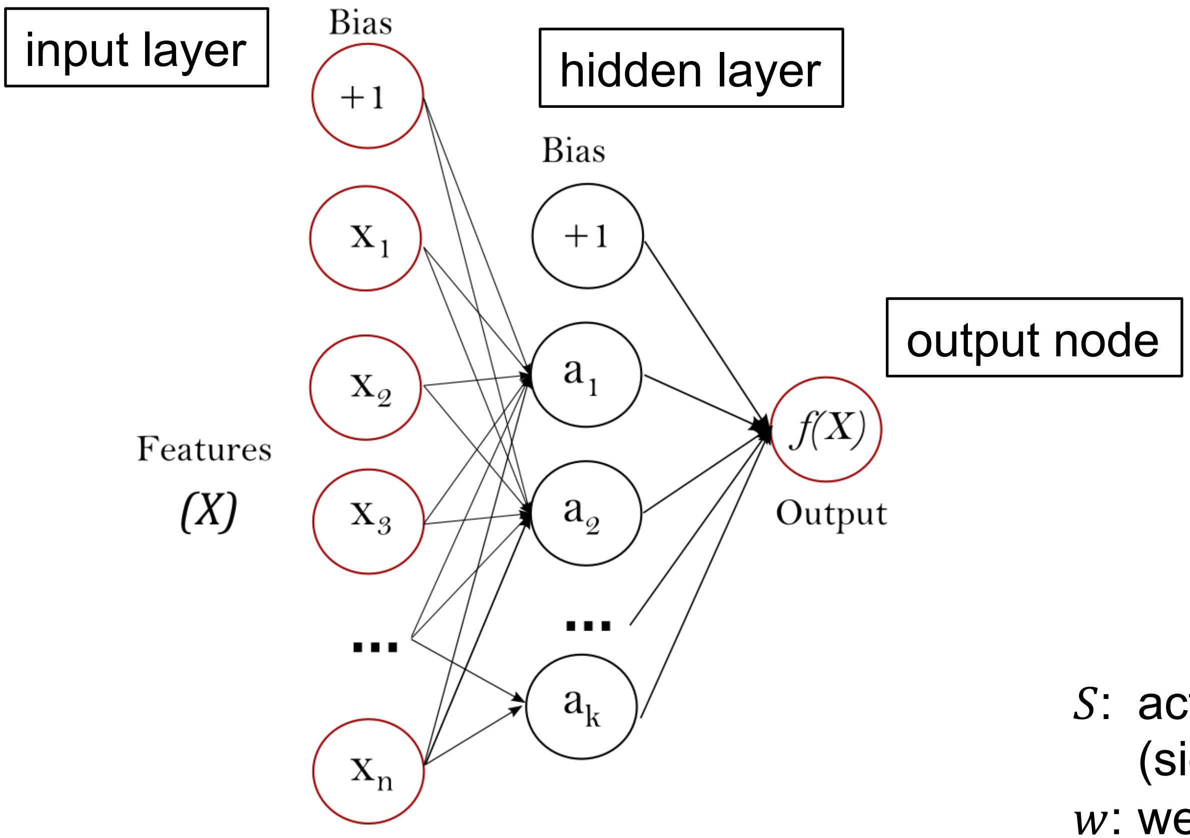
Supervised Learning: Support Vector Regression



Support Vector Mach.
hyper parameters:
C = 10
gamma = 0.1
kernel: radial basis fct.

$$f(x) = \sum_{k=1}^{n} y_k a_k K\left(x_k^{(SV)}, x\right) + \rho$$

Neural Networks (Regression/Classification)



$$f_{\text{neuron}}(x) = S(w \cdot x - U)$$

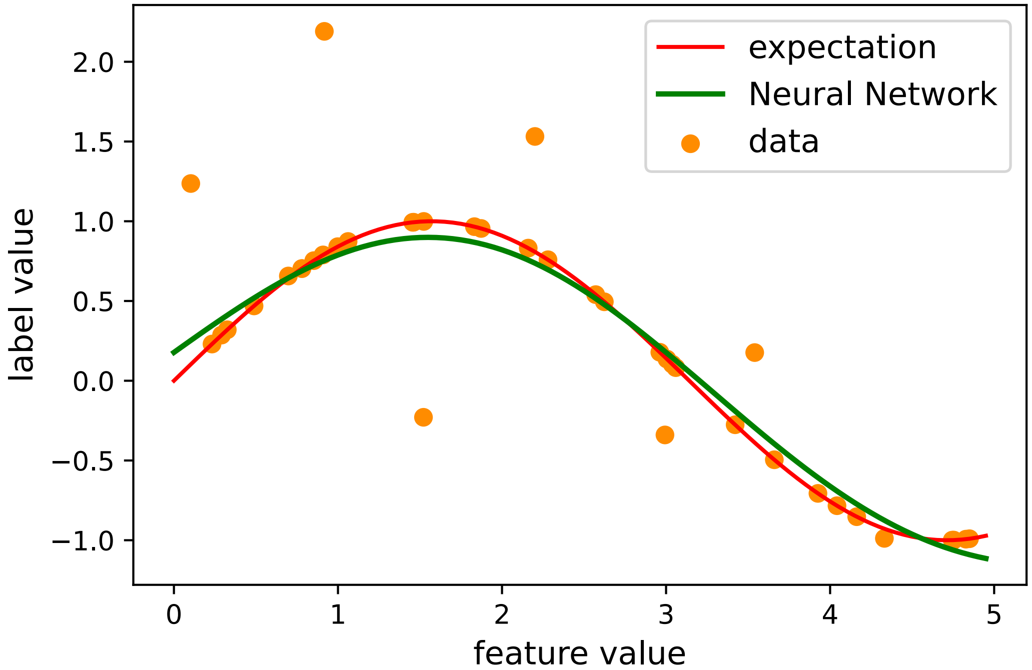
S: activation function (sigmoid, hyperb. tan, ...)
w: weight for input signals
U: bias

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

Source: scikit-learn

Supervised Learning: Neural Network

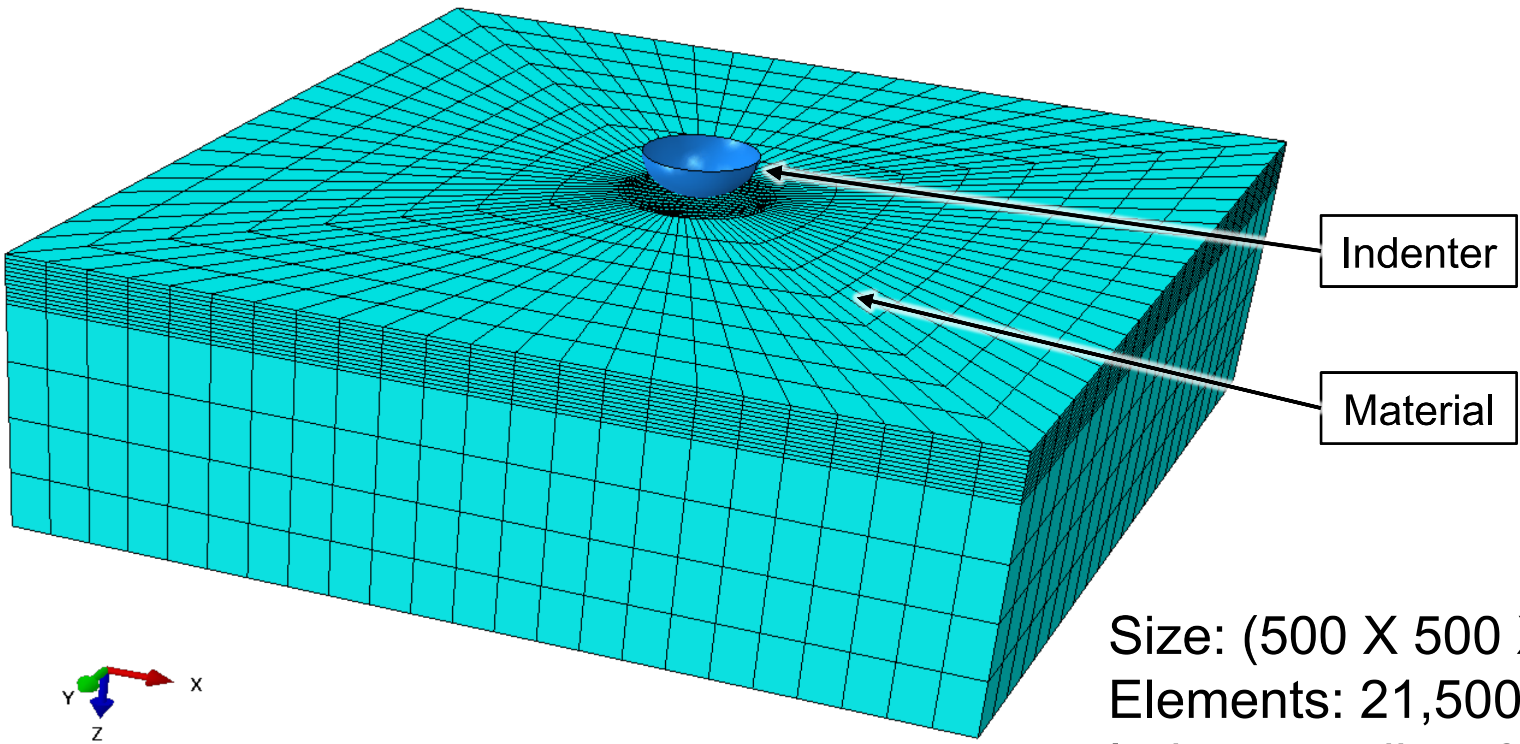
1-d example: noisy sine function



Neural Network
hyper parameters:
hidden layers: 1
neurons: 5

$$f(x) = \sum_{i=1}^{N_{\text{neuron}}} w^{(i)} f_{\text{neuron}}^{(i)}$$

Finite Element Model of indentation

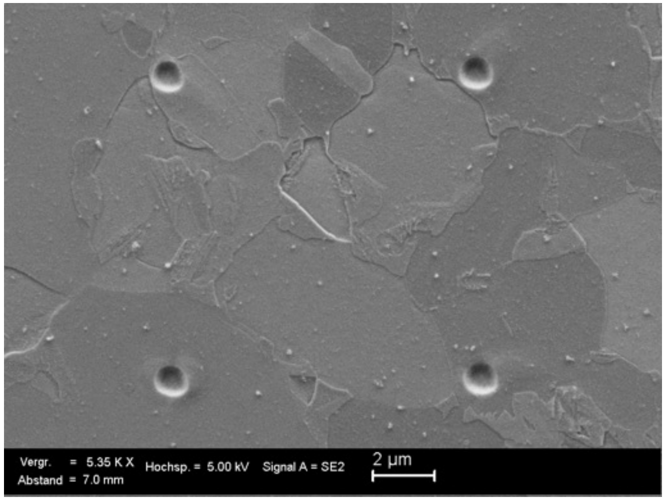


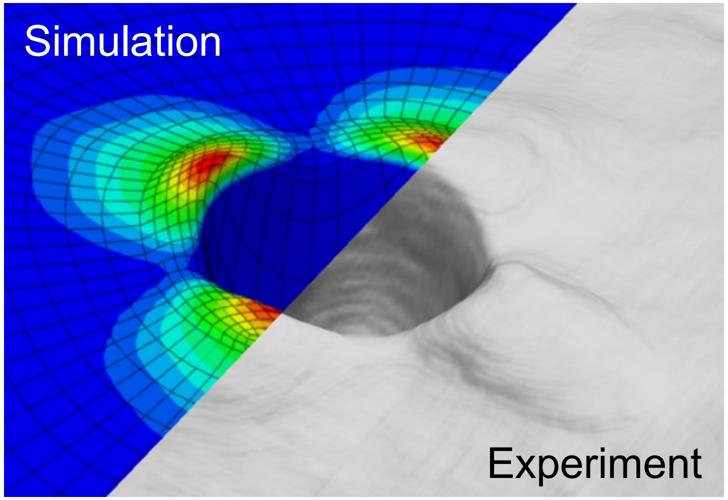
Size: (500 X 500 X 200) μm
Elements: 21,500
Indenter radius: 30 μm

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

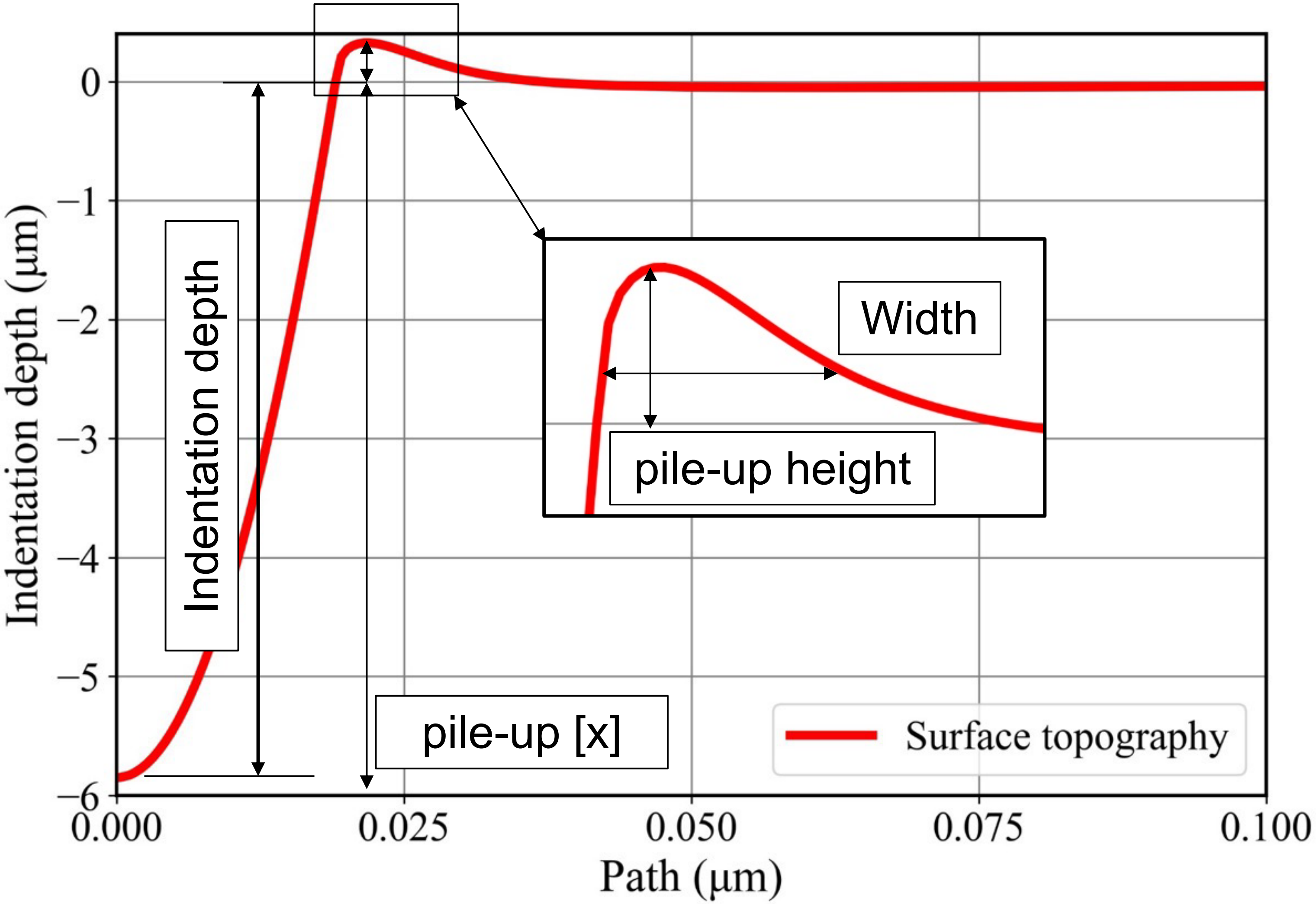
Finite Element Model of indentation

- Spherical nanoindentation into individual ferrite grains (ARMCO iron)
- Indenter tip radius: 800nm
- Max. load: 2.5 mN

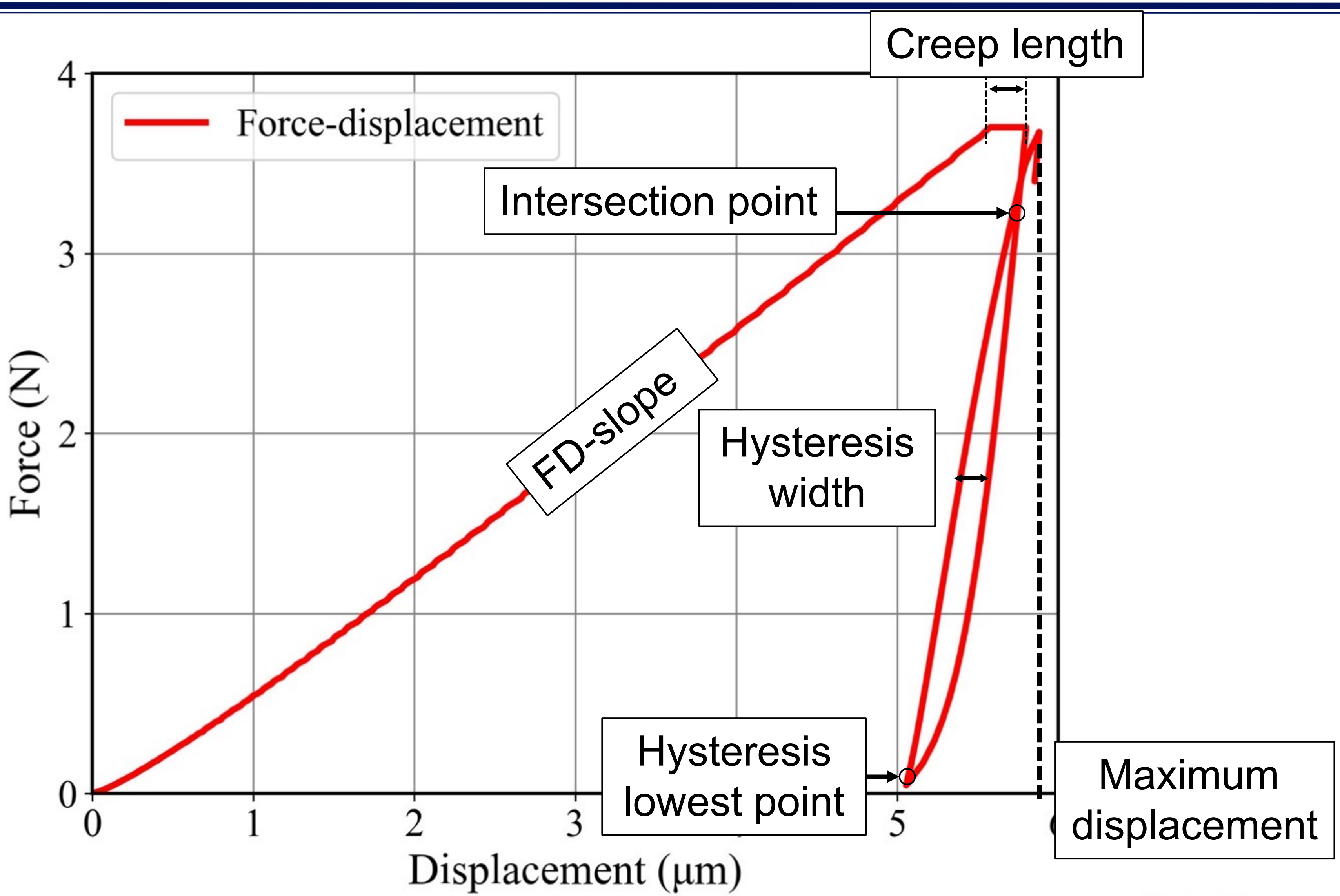




Definition of labels: Residual imprints



Definition of labels: Force-displacement curve



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

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

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

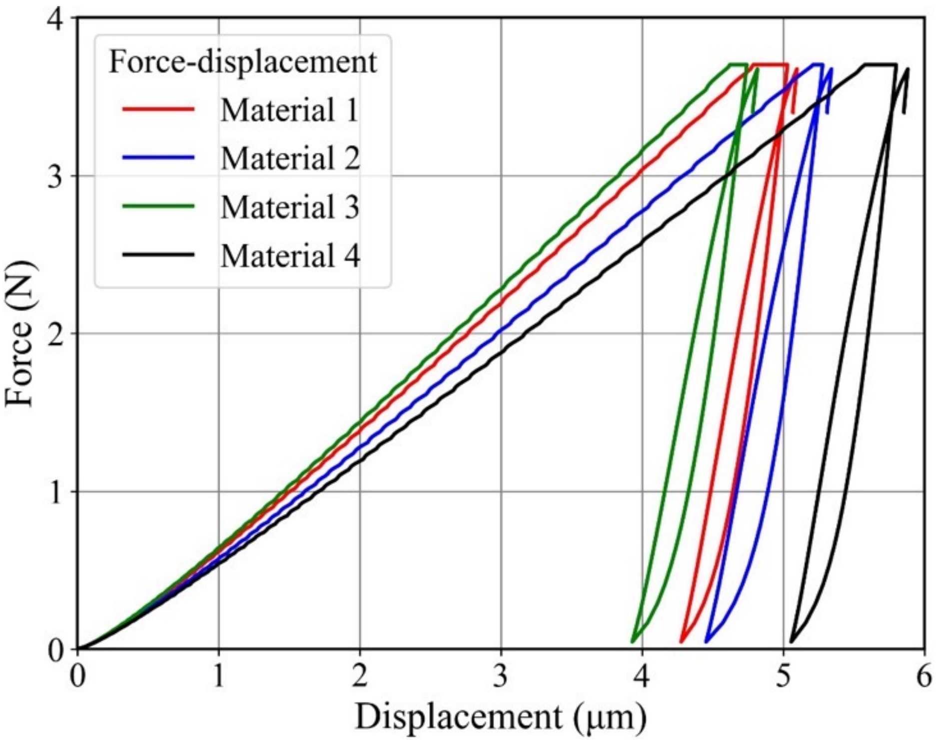
Material Parameter Ranges

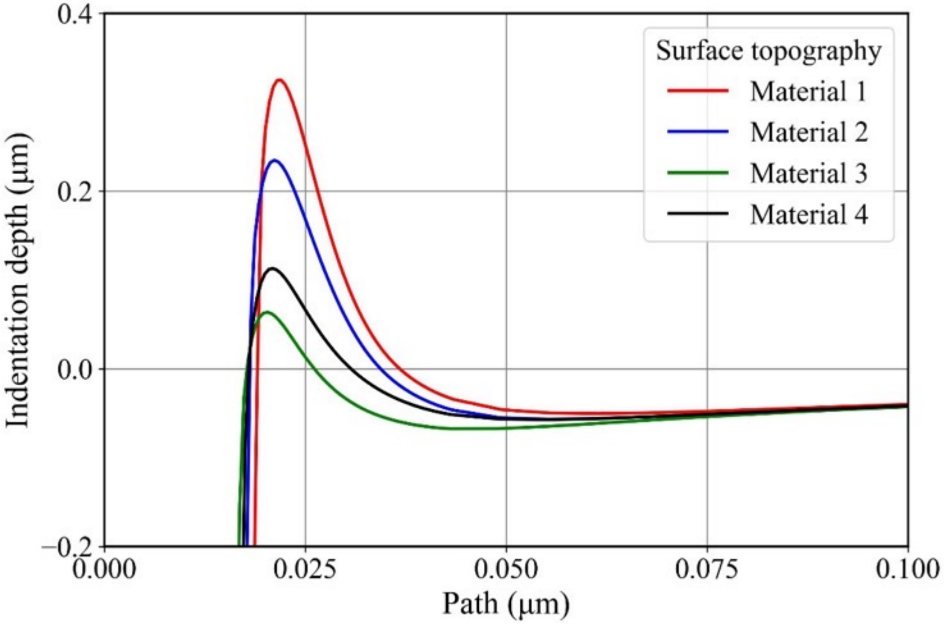
Parameter	Min	Max
A_0, 1/s	1E-07	1E-05
n, -	1.75	3.0
m, -	-0.95	-0.5
C_1, MPa	125000	225000
g_1, -	350	550
C_2, MPa	3000	5500
Q, MPa	-350	-1750
b, -	0.5	25

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

Database generation

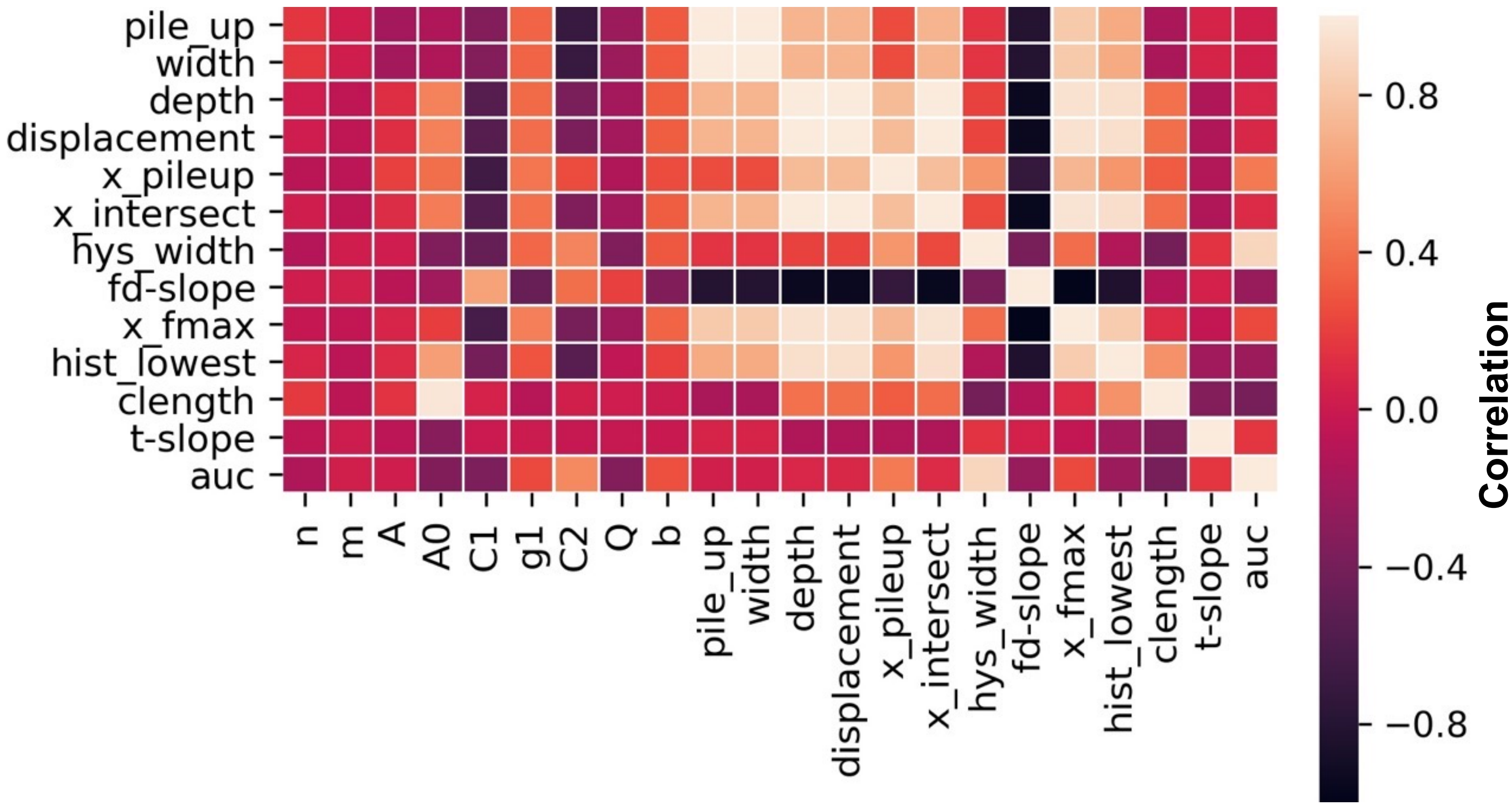
Simulation output





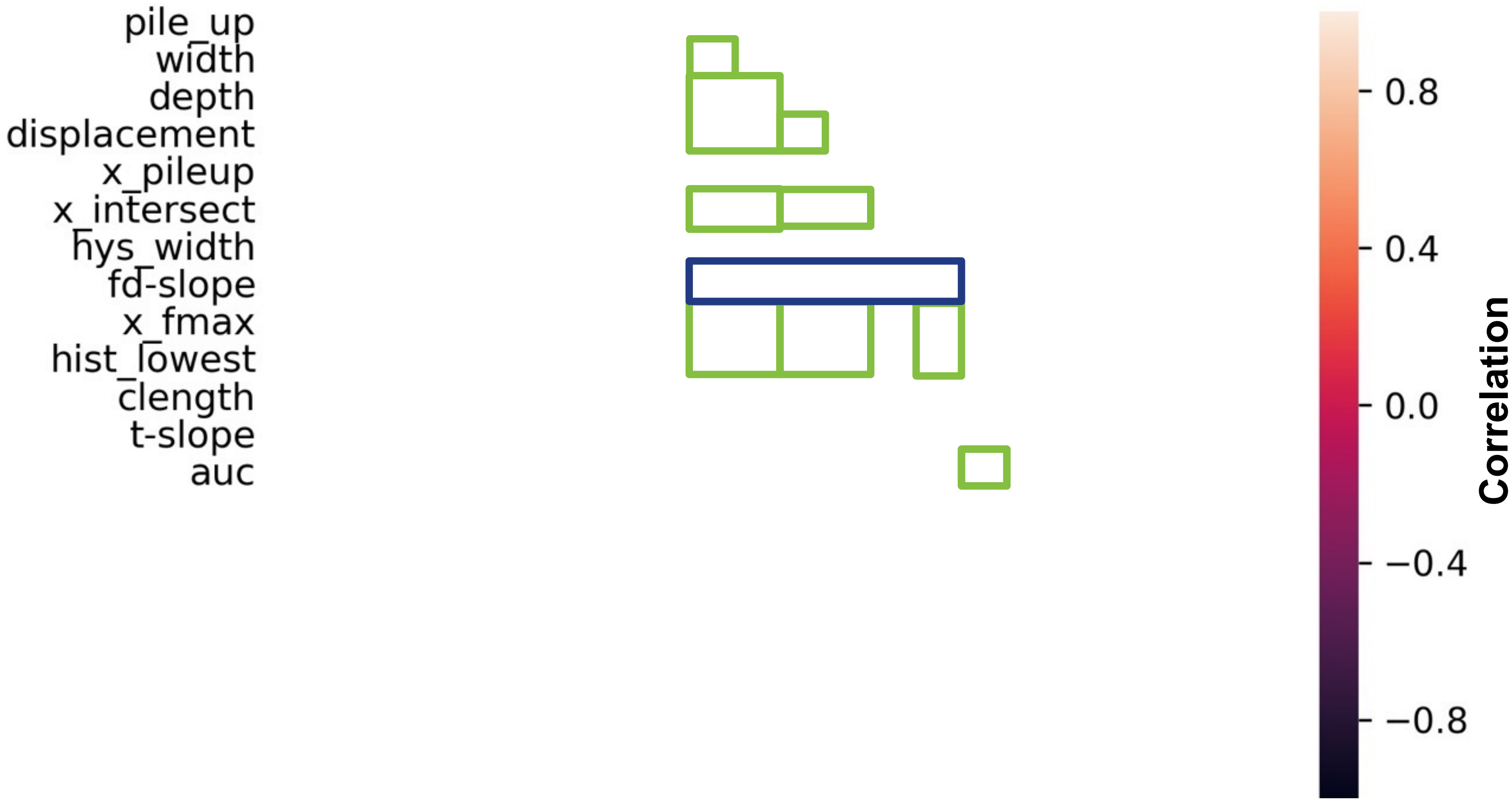
Feature selection

Heat map of extracted features and labels

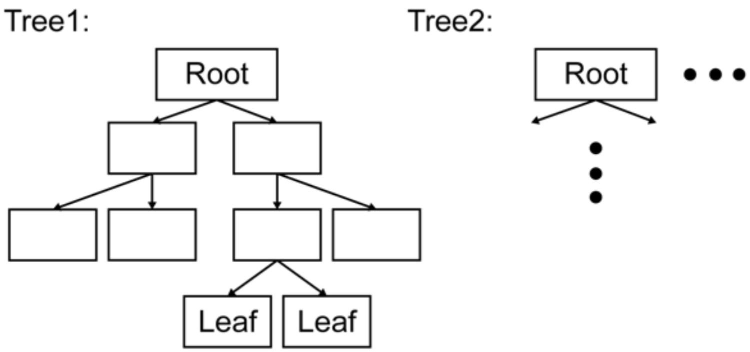


Feature selection

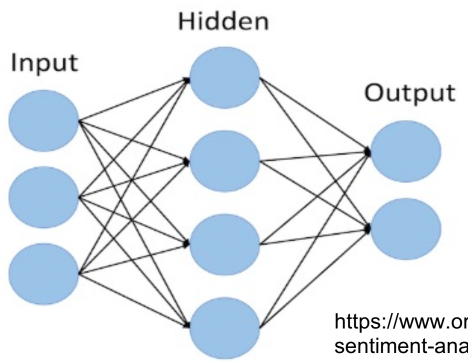
Heat map of extracted features and labels



Training and testing of ML algorithms

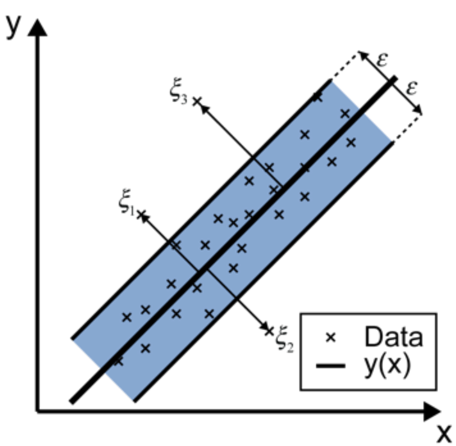


Random forest regression



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

Artificial neural networks

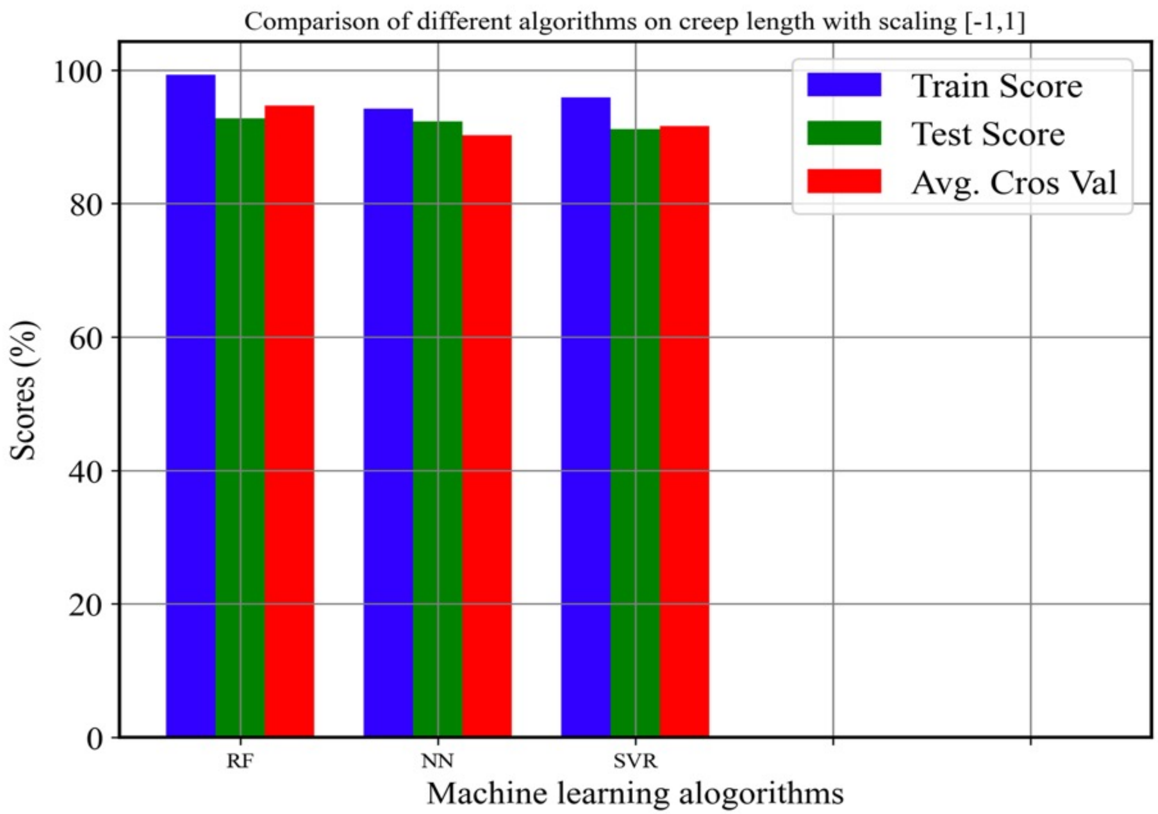


Support vector regression

- 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

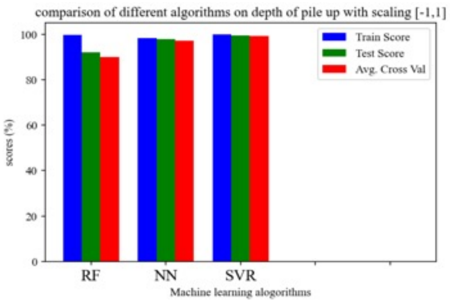
ML Regression for Characteristic Quantities

Function relating material parameters to characteristic quantities of indentation process:

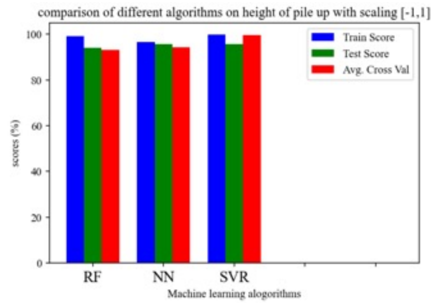


- creep length
- F-D slope
- indentation depth
- pile-up height
- …

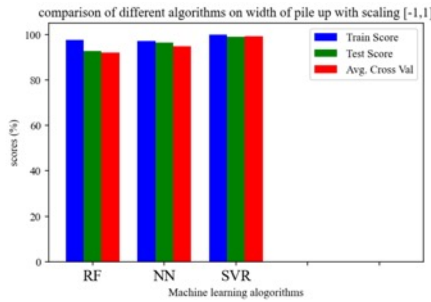
Training and testing results



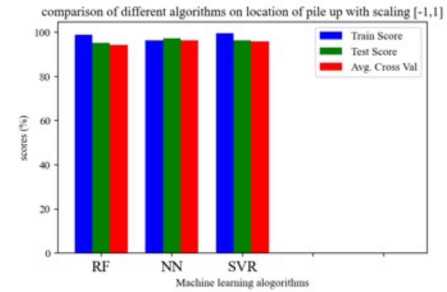
Indentation depth



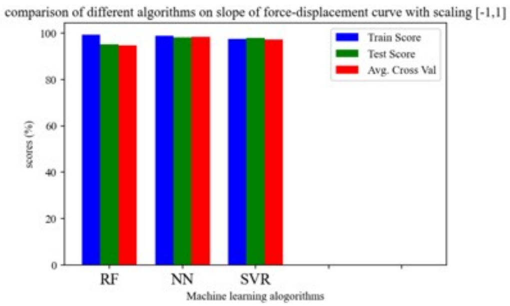
Pile-up height



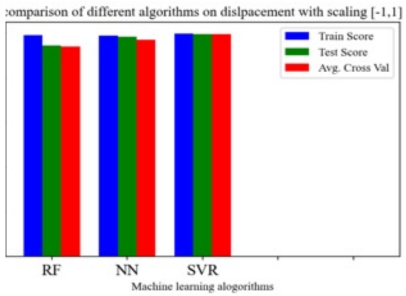
Pile-up width



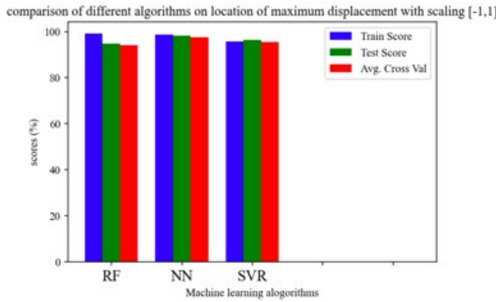
Pile-up location



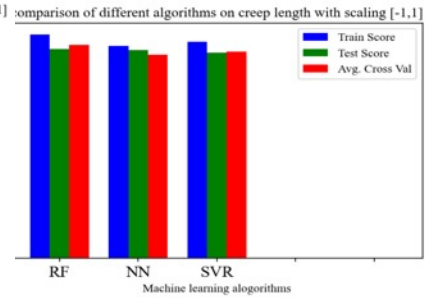
F-D slope



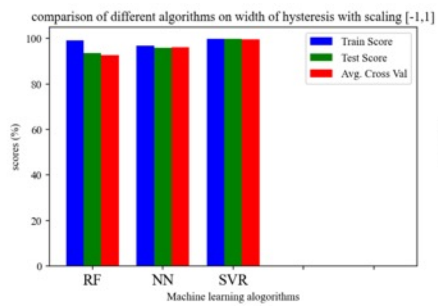
Displacement



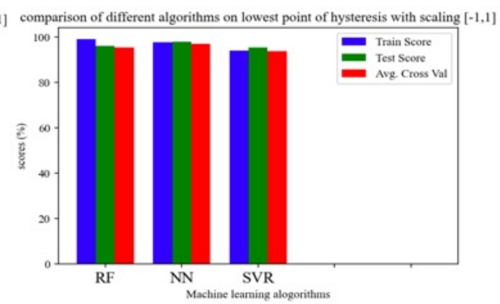
Max. displacement



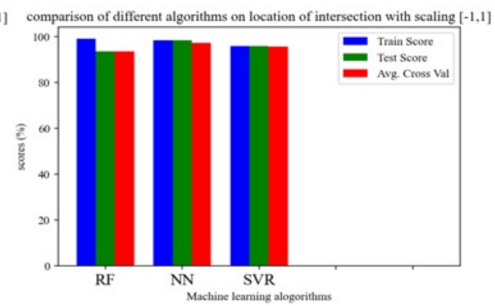
Creep length



Hysteresis width



Hyst. lowest point



Intersection point

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

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

Damage Model

output vector

$\dot{D}$

Parameterization with experiment

Macroscopic Damage Modeling

input vector

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

microstructure

Trained Machine Learning Regressor

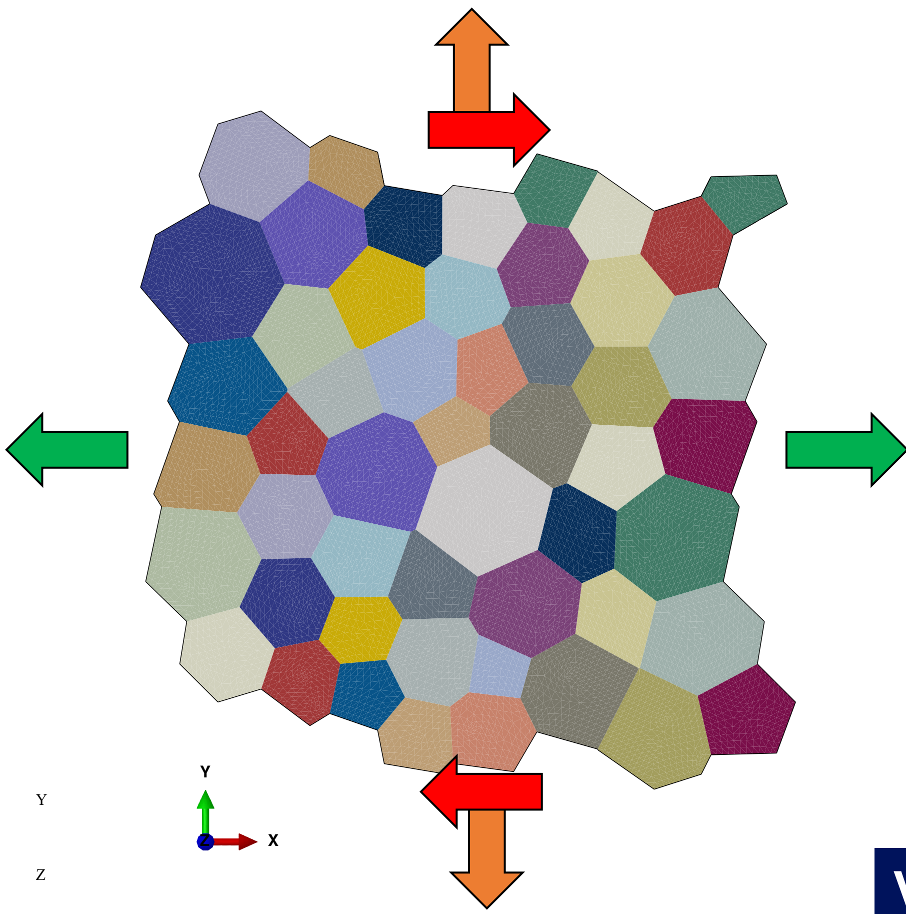
output vector

$\dot{D}$

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

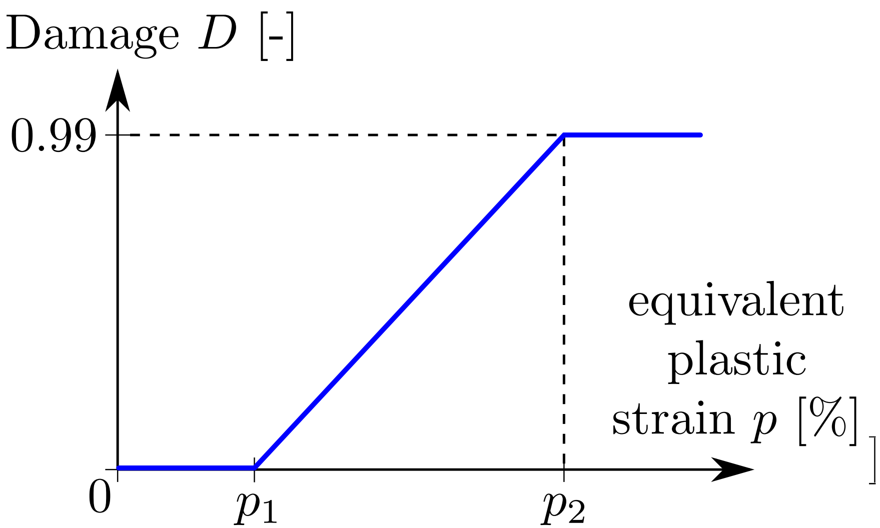
Virtual mechanical testing along various load paths (uniaxial, bi-axial, shearing, …)

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:

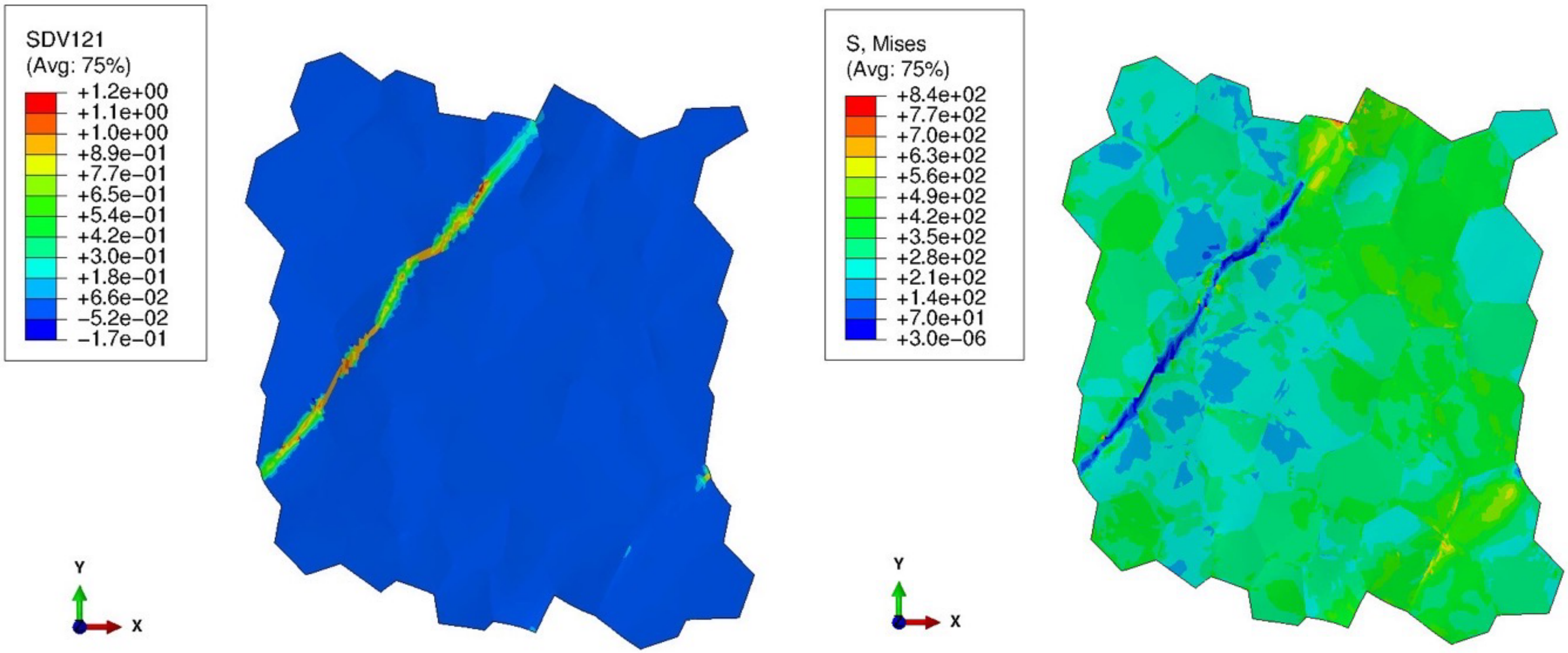


$$D = \frac{p - p_1}{p_2 - p_1}$$

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)

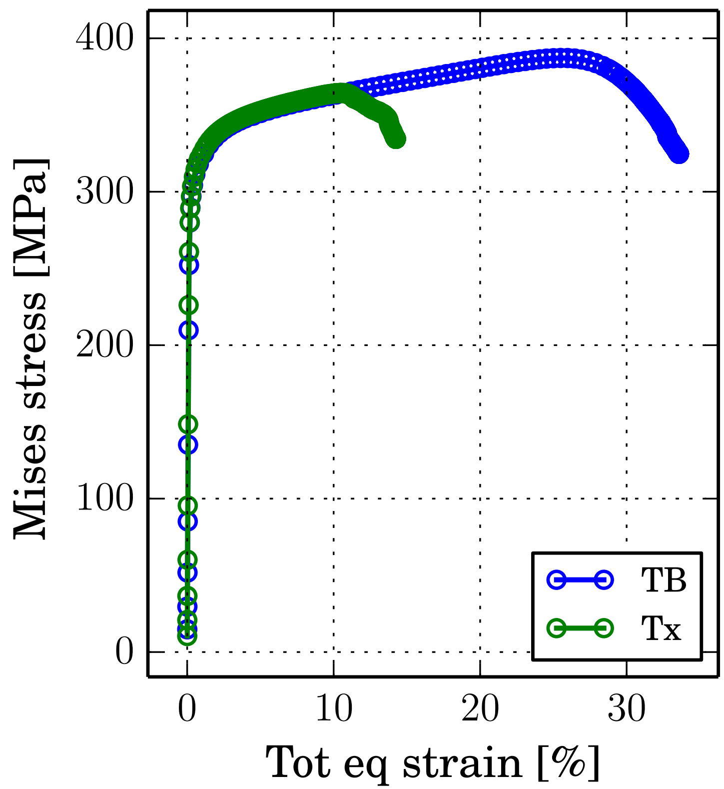


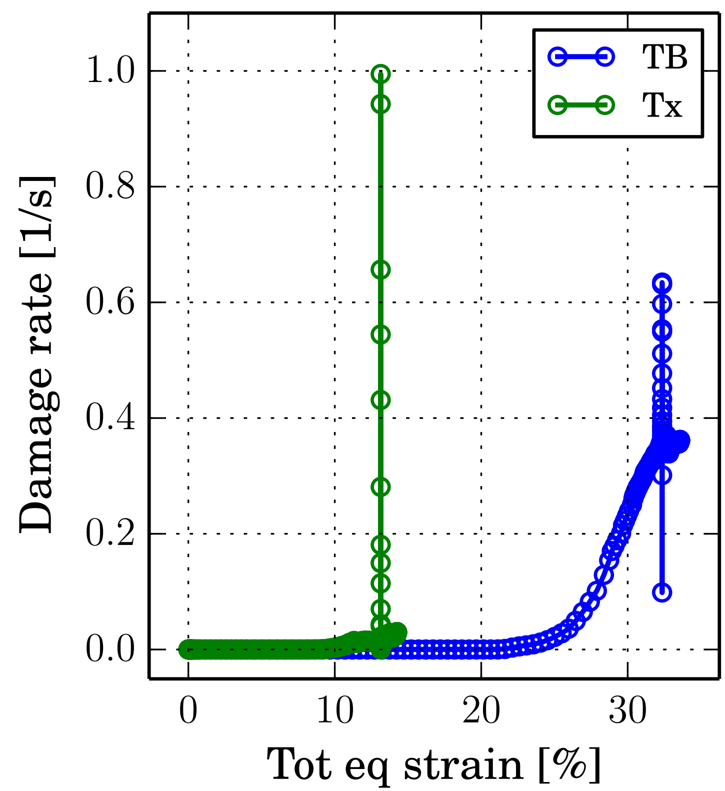
Macroscopic (homogenized) damage

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

Other quantities homogenized by volume averaging

Virtual mechanical testing: Data generation

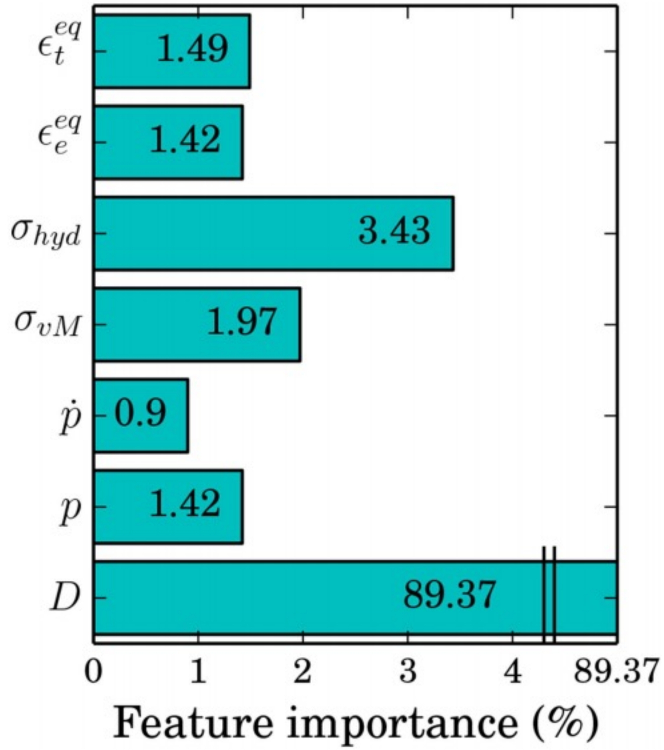




Generation of training & test data by virtual mechanical tests:

- 5 multiaxial load cases, 1545 data points
- filtering of peaks at UTS
- feature selection for ML algorithms according to established damage models

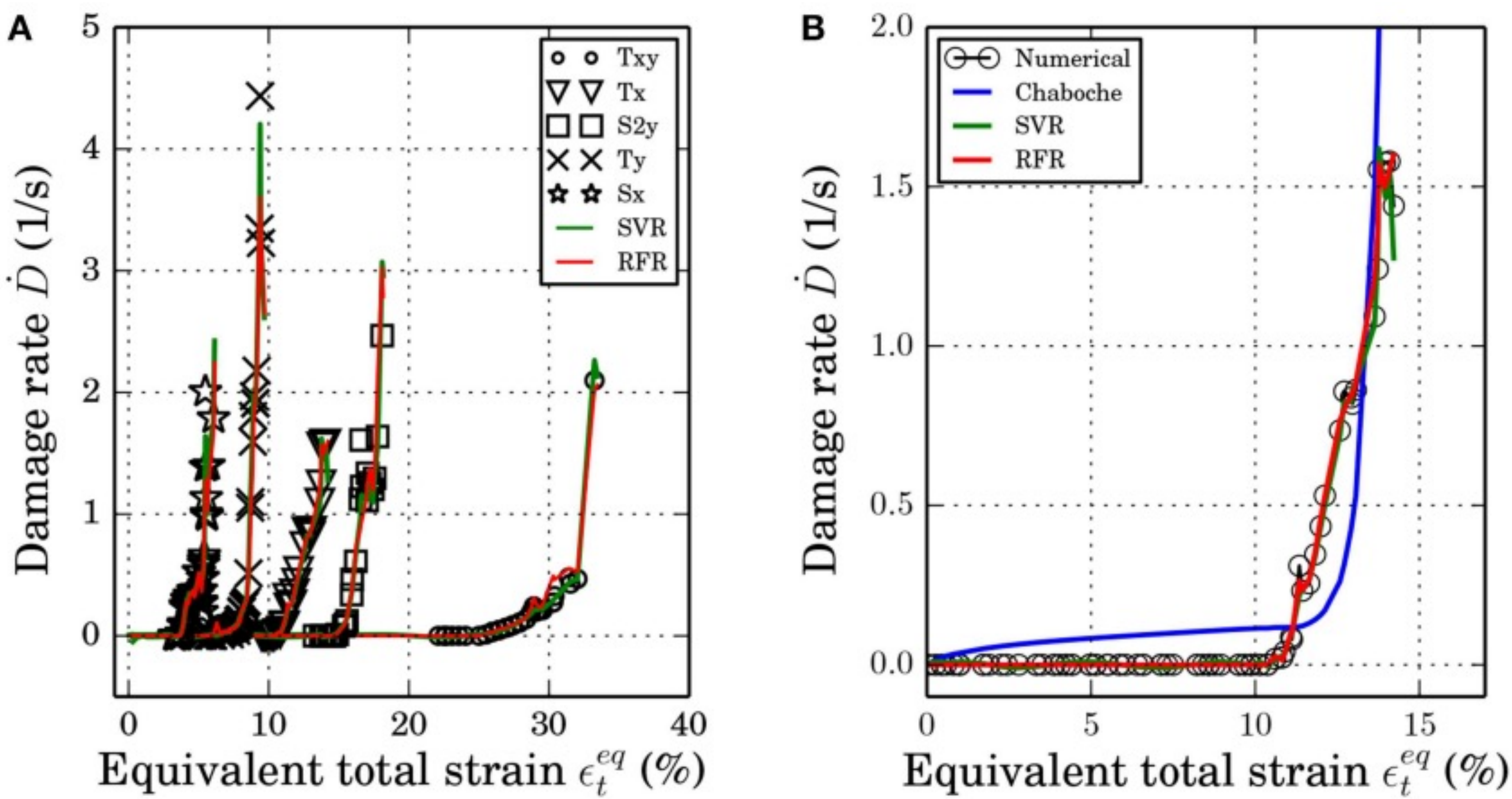
Feature selection



Influence of different mechanical quantities on damage evolution rate.

Selection of features for input vector of ML model.

Training result



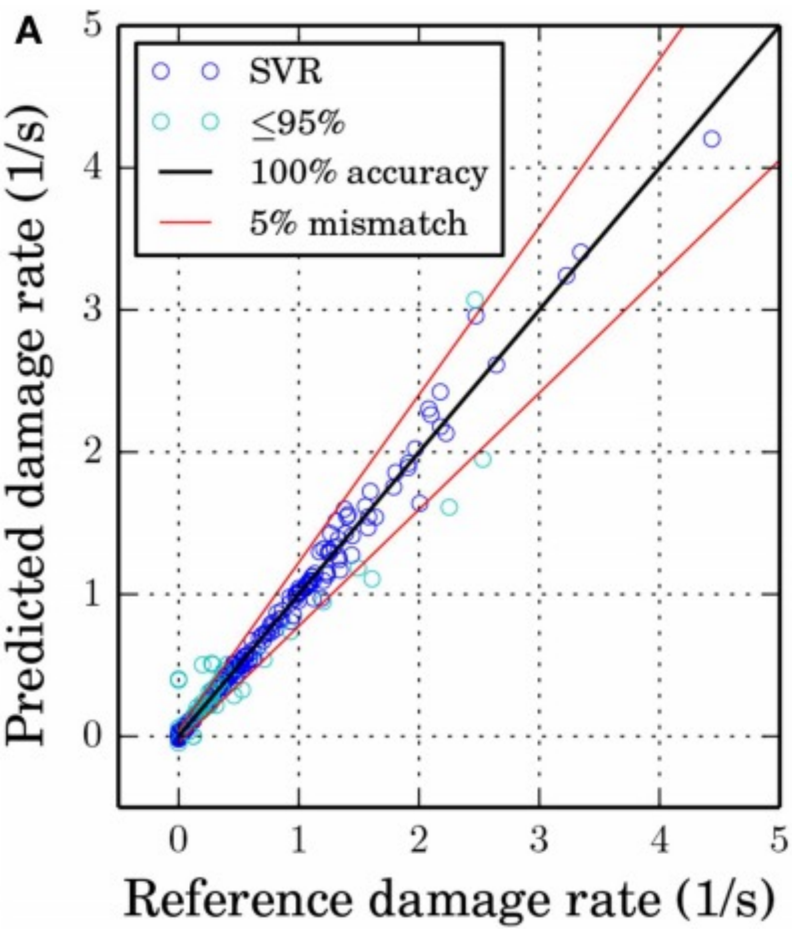
Filtered data for damage rate (removal of sharp peak)

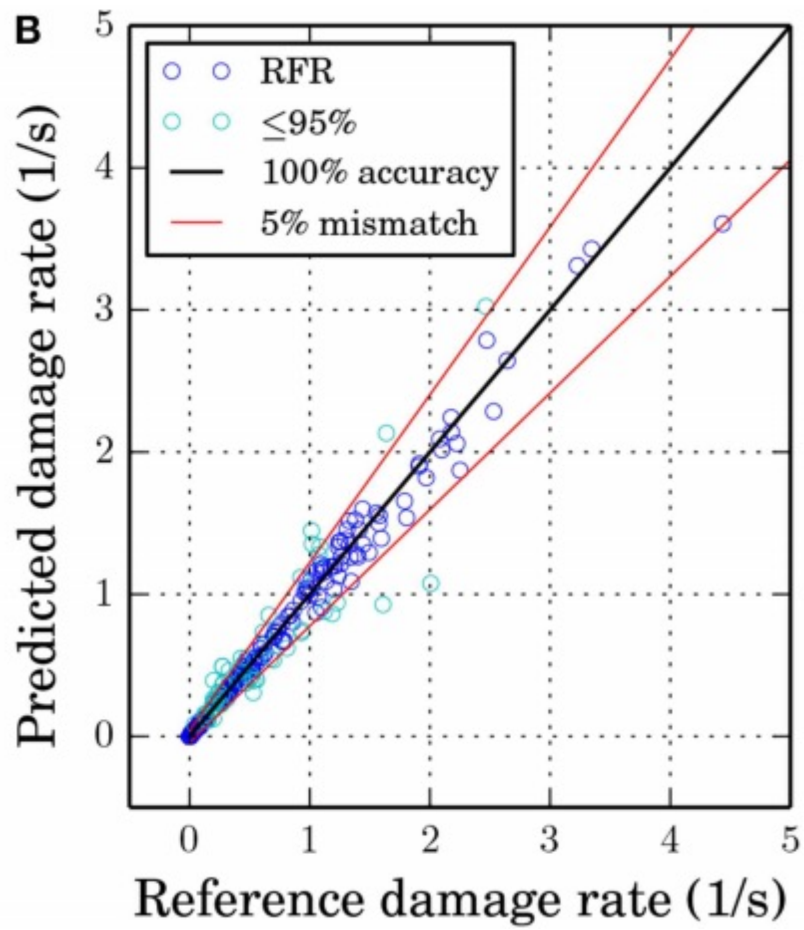
Comparison with Chaboche damage model

Prediction of Damage Rate

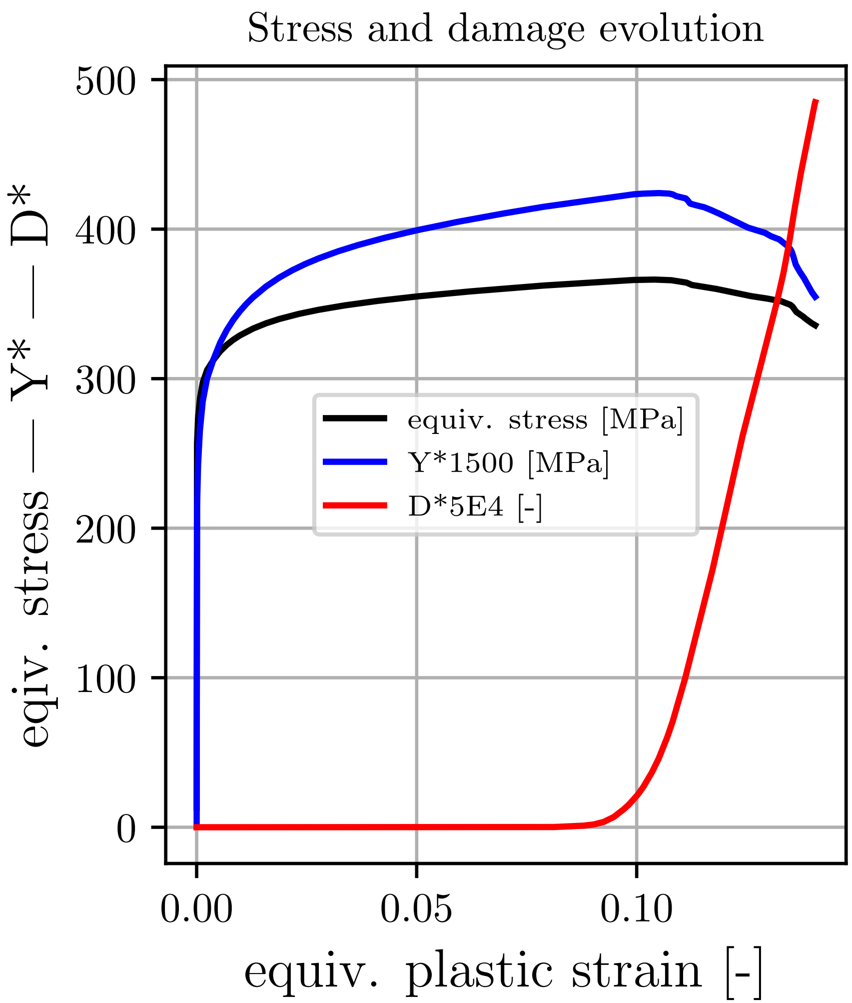
Support Vector Regression
Test score: 98.25%

Random Forest Regression
Test score: 97.48%





Comparison with analytical damage models



Chaboche model

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

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

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

Reimann et al. Frontiers in Materials 2019

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

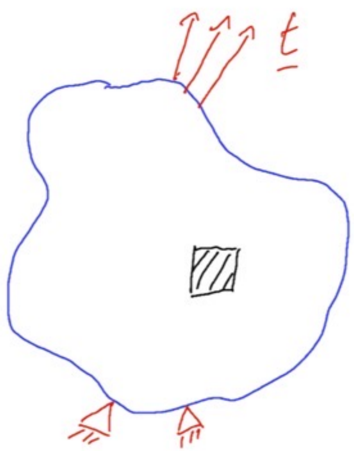
Co-authors: D. Reimann, K. Nidadavolu, H. ul Hassan, N. Vajragupta, T. Glasmachers, P. Junker

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

Constitutive modeling (State-of-the-Art)

Component
Mechanical / thermal / environmental loading conditions

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



Constitutive models
History dependent evolution equations for state variables

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

Material parameters
Material specific input to constitutive models

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

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 $D = D(t)$

Problem
- Solutions involve rather intricate mathematical formalism
- Explicit consideration of microstructure is difficult
- Material parameters vary with microstructure

Constitutive Modeling: Digital Twin of Material

Component
Mechanical / thermal / environmental loading conditions

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

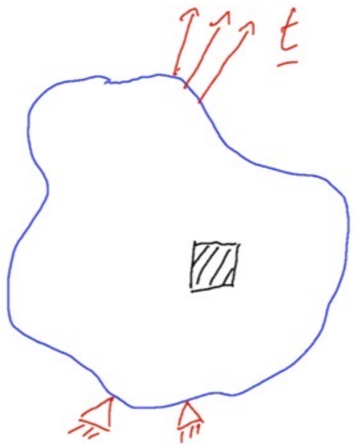
Digital Twin

Data-based material representation

- mechanical & functional properties
- transport properties
- explicit consideration of microstructure

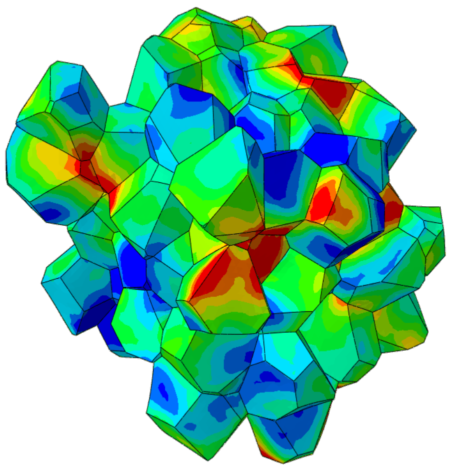
Material response
Time dependent state variables

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

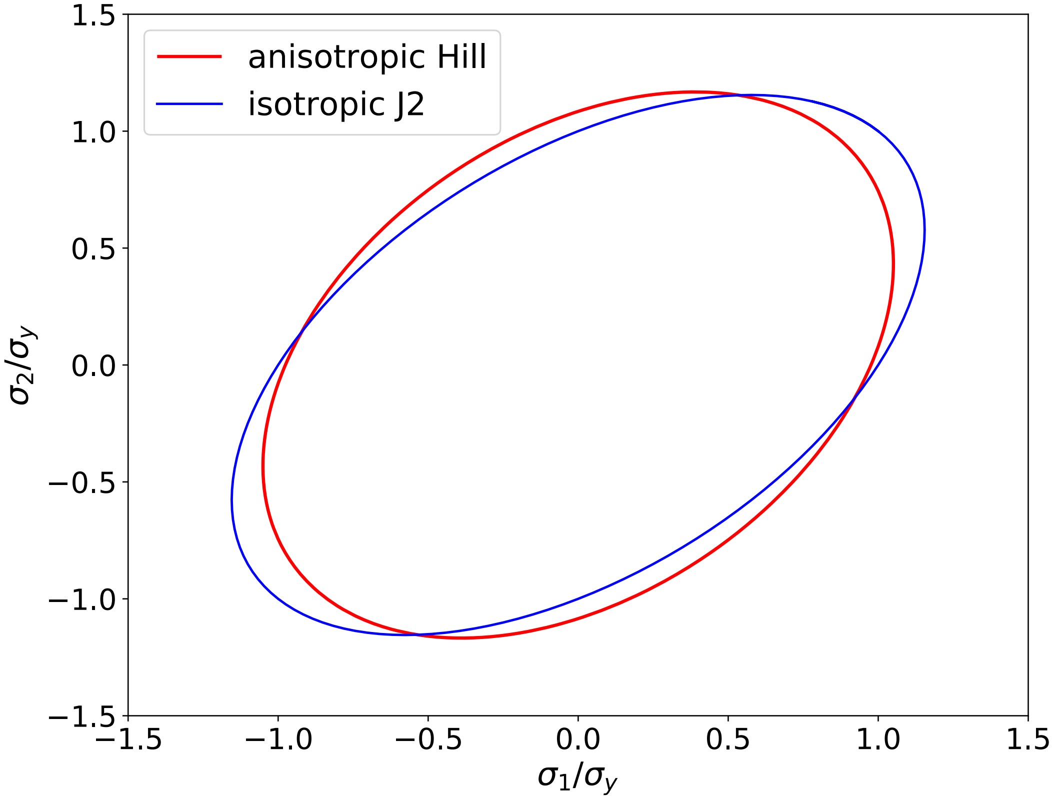


Hybrid materials data

- physics-based models
- experiment



Continuum plasticity



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

Yield function

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

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

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

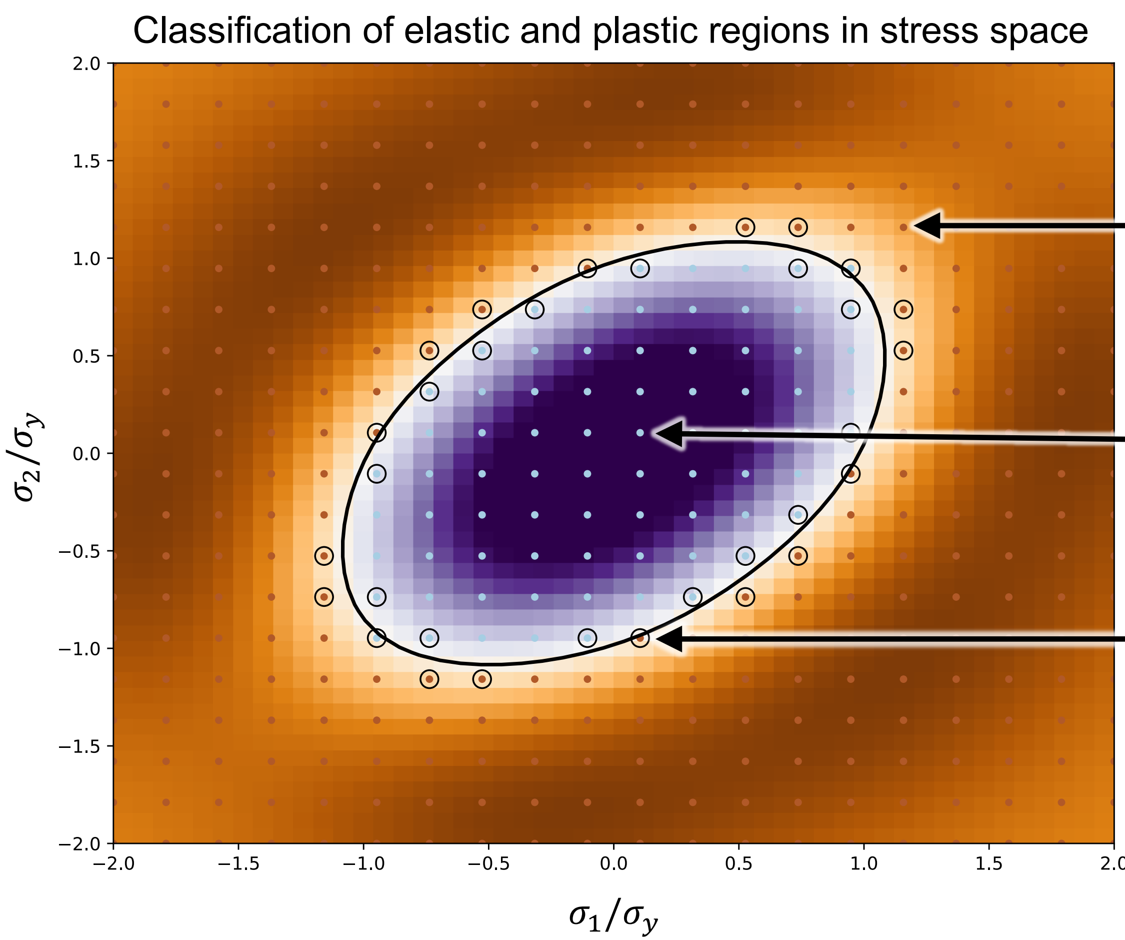
Plastic strain rate (flow rule)

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

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

Data-based model for plastic yielding

Trained Support Vector Classification



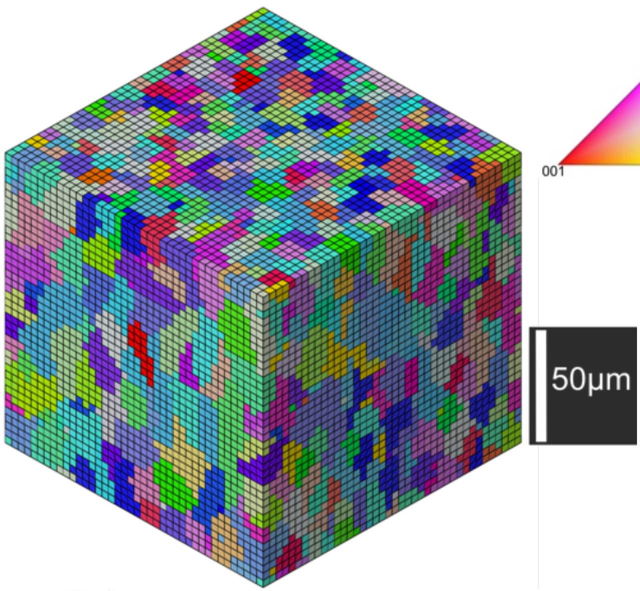
Data points (stresses) classified as “plastic” (f ≥ 0)

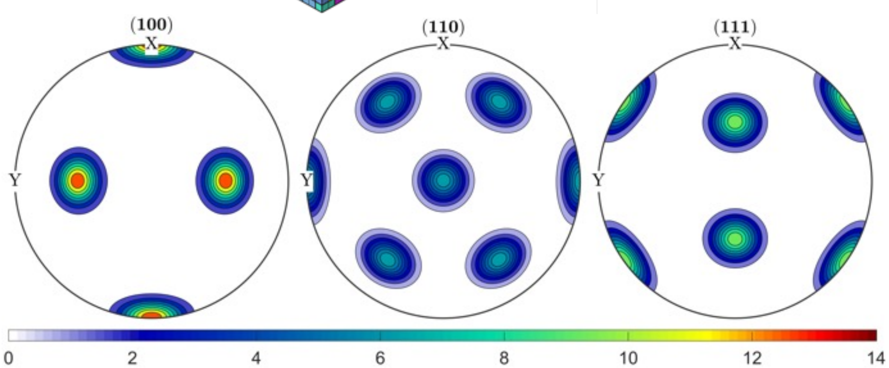
Data points (stresses) classified as “elastic” (f < 0)

Support vectors (data points close to delimiter)

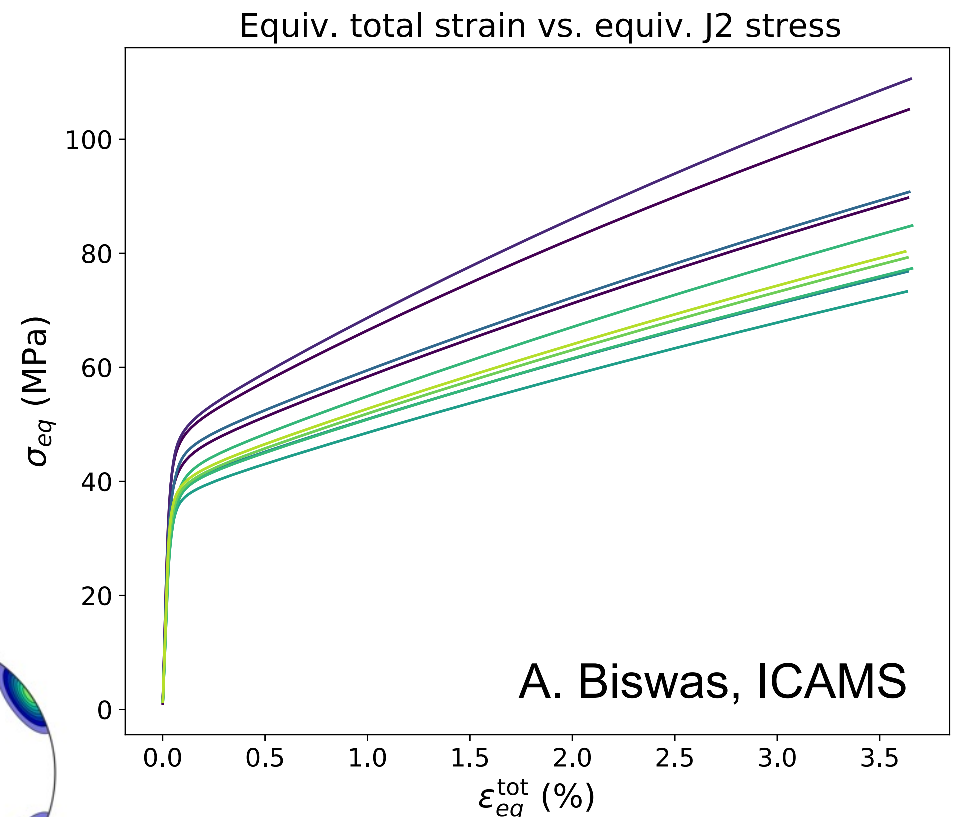
A. Hartmaier Materials 2020

Training Data Generation by Micromechanical Model





Micromechanical model with ~2,200 grains and Goss texture

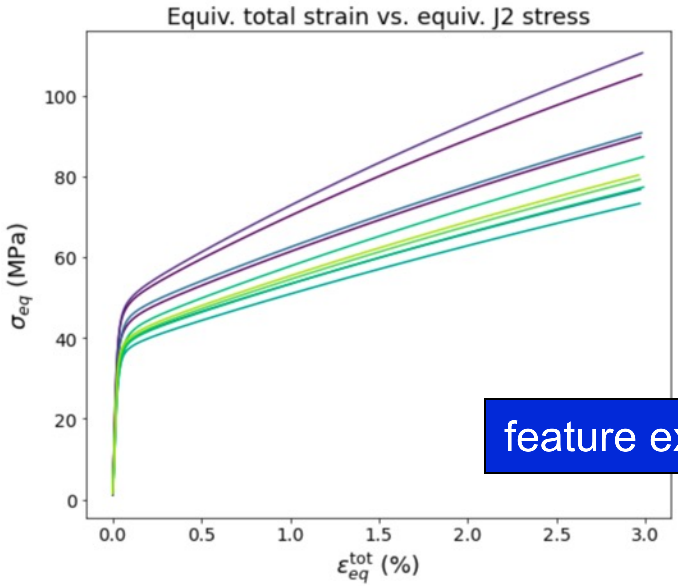


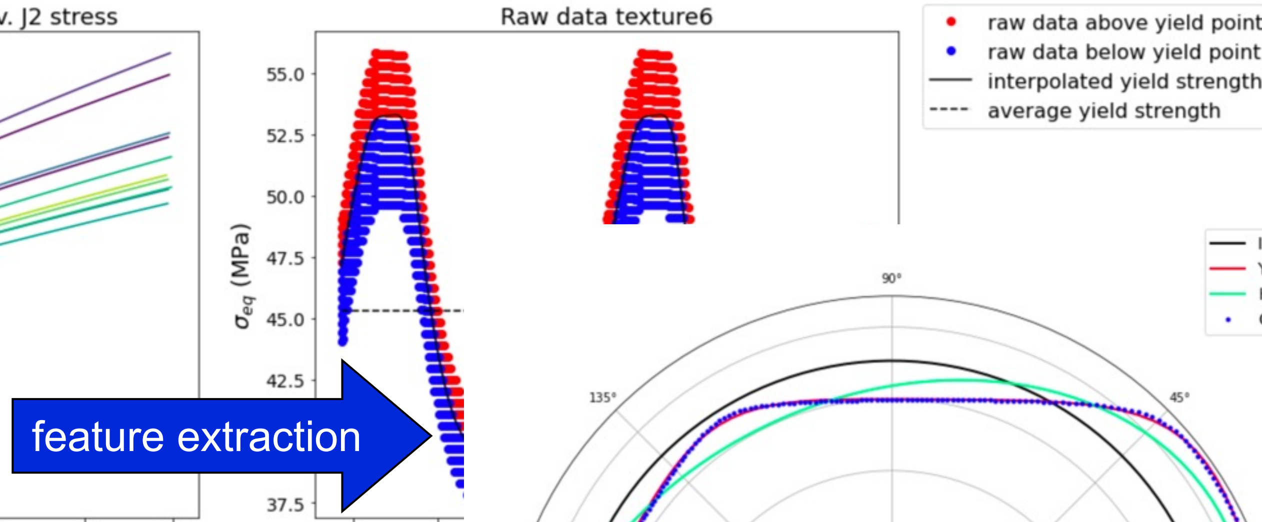
Resulting stress-strain curves

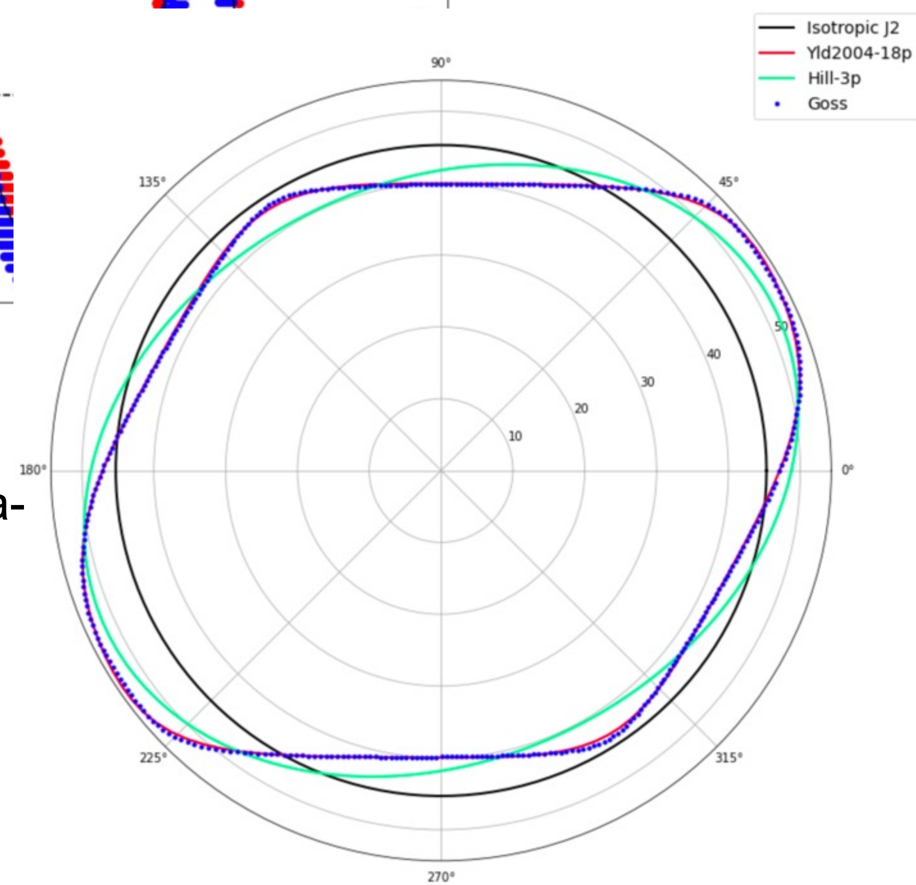
- Crystal plasticity model
- Different deviatoric load cases

Anisotropic yield strength → data

Training of Classifier for Elastic / Plastic Regions





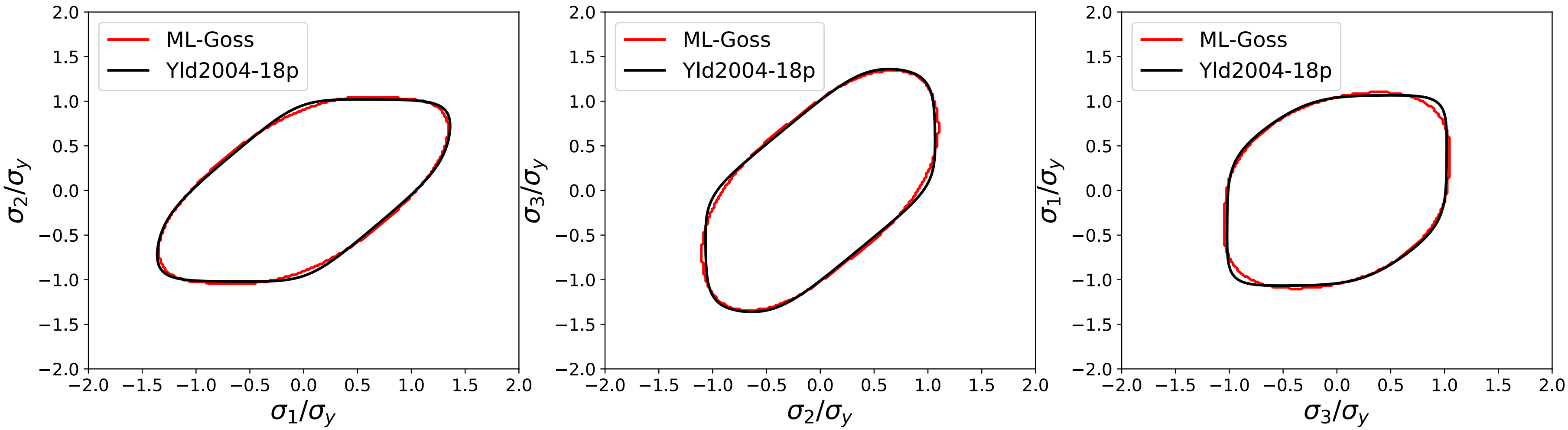


Digital Twin: Microstructure sensitive data-based description of plastic material properties

Experimental data

- validation of micromechanical model
- hybrid mechanical data bases

Trained Machine Learning Flow Rule

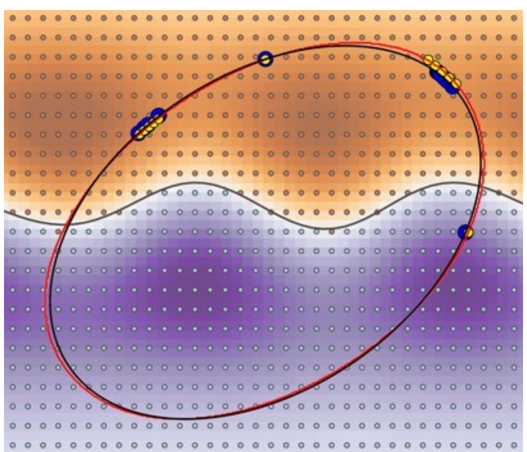


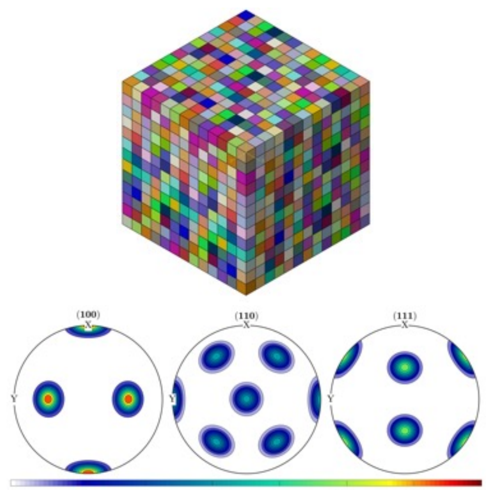
Yield loci in different cross-sections of principle stress space

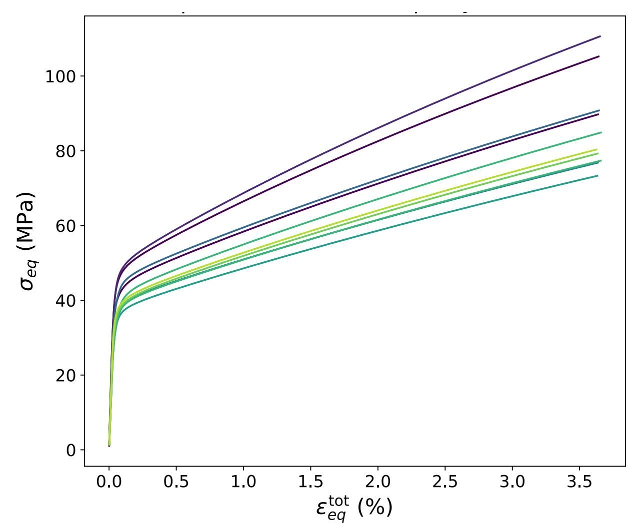
- Mechanical data obtained from micromechanical model
- Training of Machine Learning flow rule
- Barlat Yld2004-18p yield function fitted to same data

Summary – Continuum Plasticity

- **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**







Machine Learning for Mechanical Systems

Supervised Machine Learning (ML) methods:

- Support vector Machines
  (robust and moderate data requirements, only 2 hyper parameters)
- Neuronal Networks
  (prone to overfitting, large volumes of training data required, many hyper parameters)
- Gaussian Process
  (not covered here)
- Random Forrest
  (simple and robust reference method, prone to overfitting, only 2 hyper parameters)

Feature Selection

Hyper parameter tuning

Independent Validation