ALERT Geomaterials Doctoral School 2023

Overview of Machine Learning

Ioannis Stefanou

Ecole Centrale de Nantes, GeM (Institut de Recherche en Génie Civil et Mécanique), France

ioannis.stefanou@ec-nantes.fr

CENTRALE NANTES
SHAKE THE FUTURE.

GeM
Institut de Recherche en Génie Civil et Mécanique

erc
European Research Council
Established by the European Commission

CoQuake.com

CONNECTALENT.ORG

Loupl

Learning (definition)

LoupI is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at accomplishing tasks in T, as measured by P, improves with experience E.

Training LoopI to sit down

LoupI is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at accomplishing tasks in T, as measured by P, improves with experience E.

- Class of tasks, T:

  - Sitting down after vocal command in French
  - Sitting down after vocal command in Greek
  - Sitting down after gesture

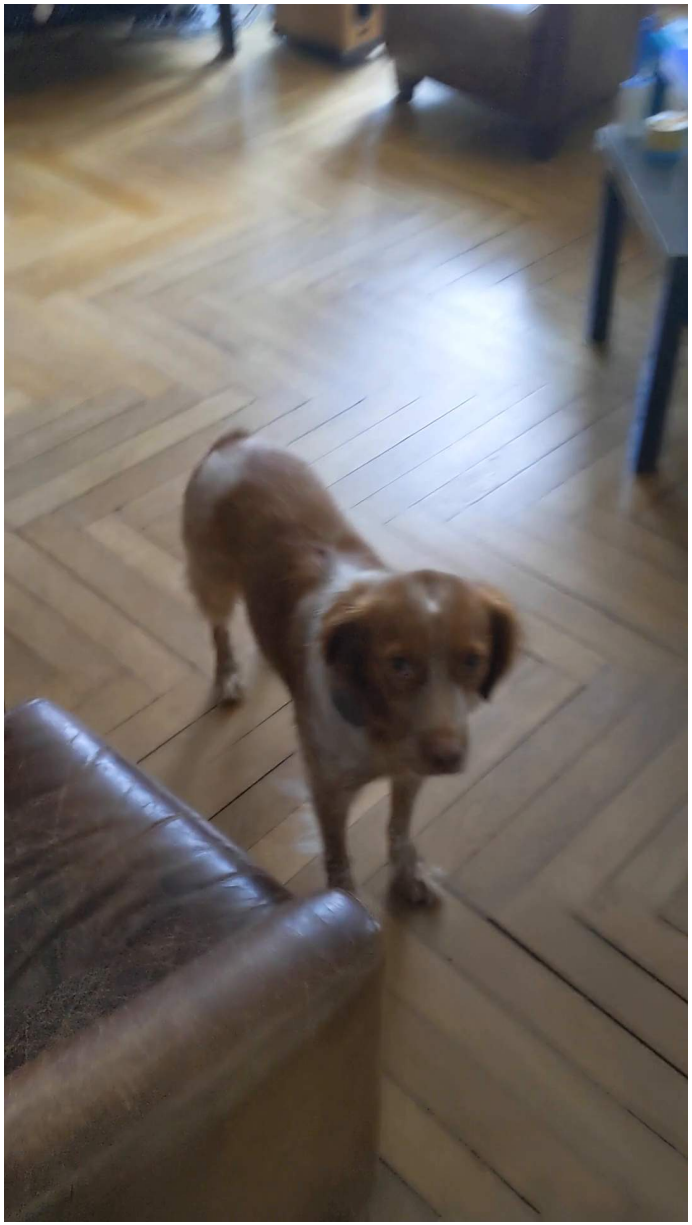
- Experience, E:

  - Hearing the vocal command
  - Taking an action (e.g. sitting down)
  - Obtaining a dog treat (if he sat down)

- Performance measure:

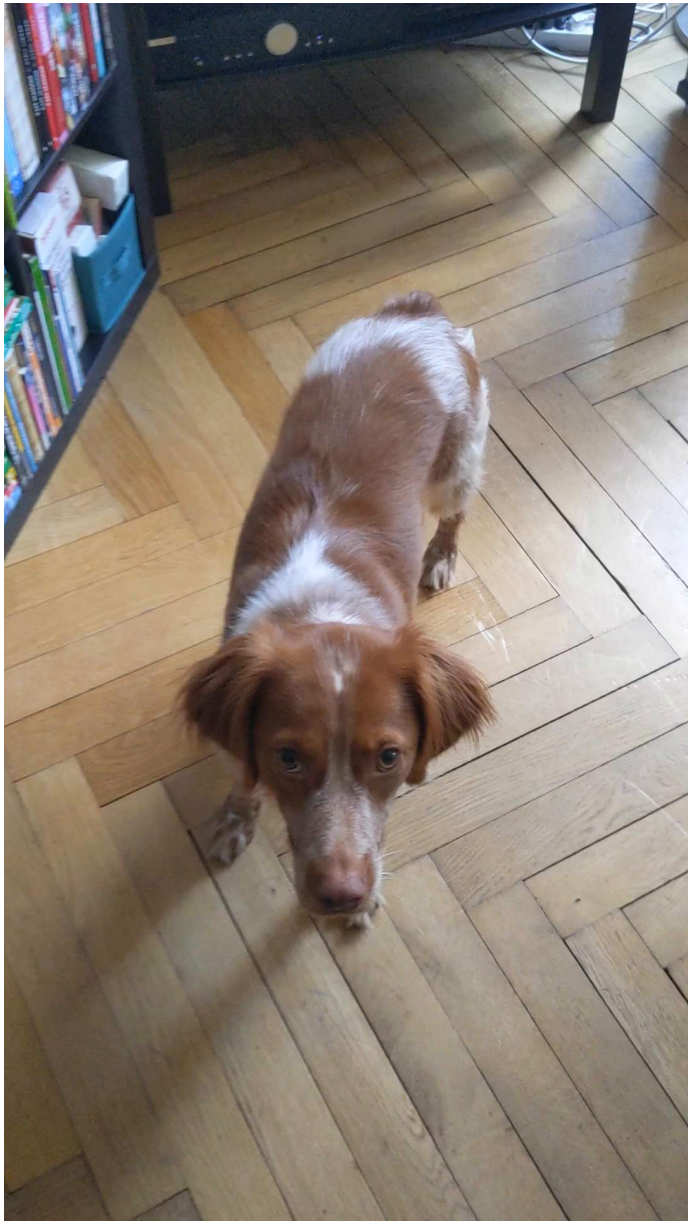
  - Is LoupI sitting down?

Inference (1 of …)



Inference (2 of …)

...but not always!



90% success rate

Machine Learning (definition)

~~LoupI~~ A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at [accomplishing] tasks in T, as measured by P, improves with experience E.

(Tom Mitchel, 1997)

A simple example of ML (regression)

A C.P. is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at accomplishing tasks in T, as measured by P, improves with experience E.

- Class of tasks, T:

  - Given x predict y
    such that to belong to C

- Experience, E:

  - Pairs (x,y) close to C

- Performance measure:

  - Distance of prediction(s) from C

A simple example of ML (regression)

A C.P. is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at accomplishing tasks in T, as measured by P, improves with experience E.

- Class of tasks, T:

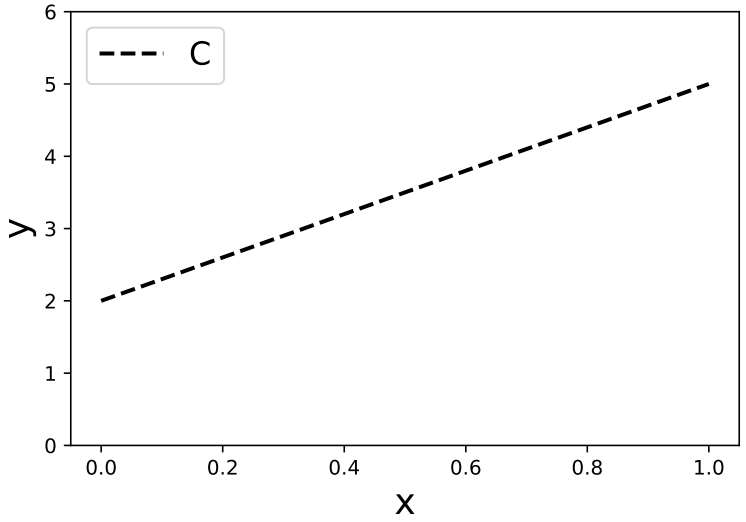
  - Given x predict y
    such that to belong to C

- Experience, E:

  - Pairs (x,y) close to C

- Performance measure:

  - Distance of prediction(s) from C



A simple example of ML (regression)

A C.P. is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at accomplishing tasks in T, as measured by P, improves with experience E.

- Class of tasks, T:

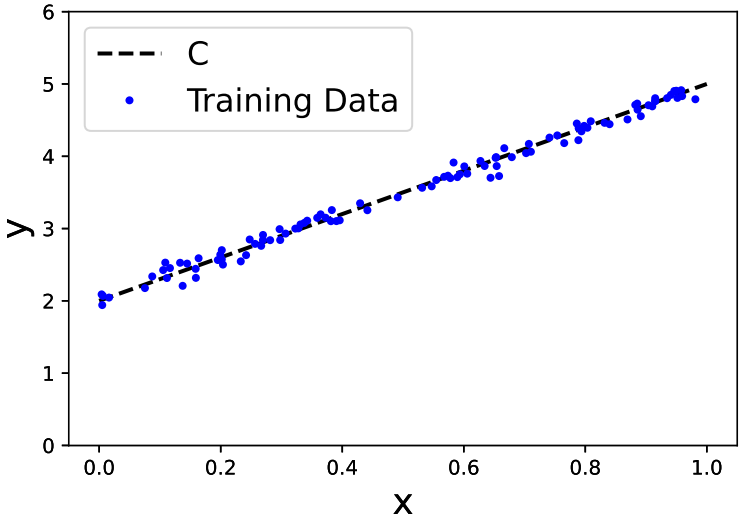
  - Given x predict y such that to belong to C

- Experience, E:

  - Pairs (x,y) close to C

- Performance measure:

  - Distance of prediction(s) from C

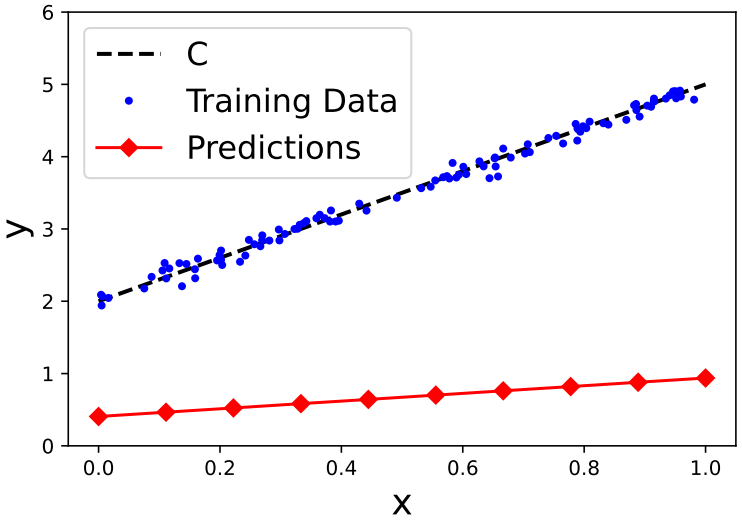


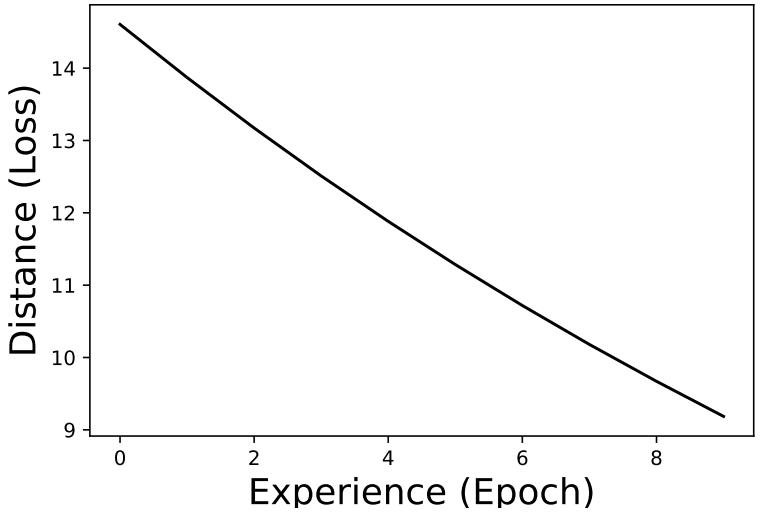
Using a simple Neural Network (pyTorch)

$$y_{pred} = f_{ANN}(x, w)$$

Using a simple Neural Network (pyTorch)

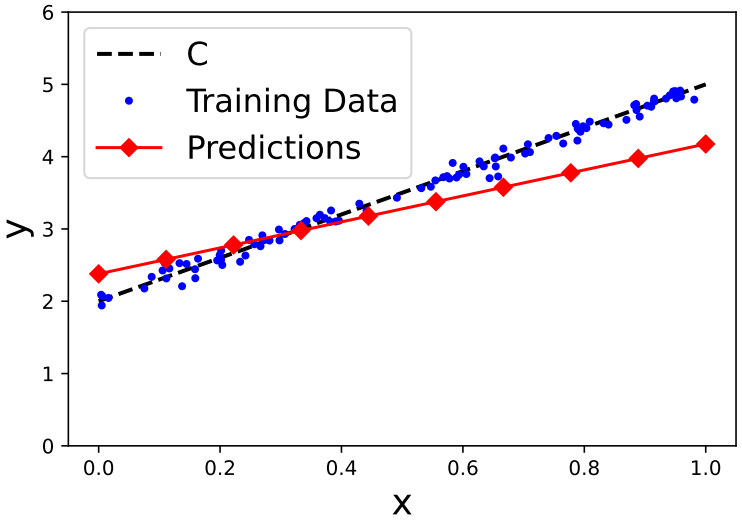
$$y_{pred} = f_{ANN}(x, w)$$

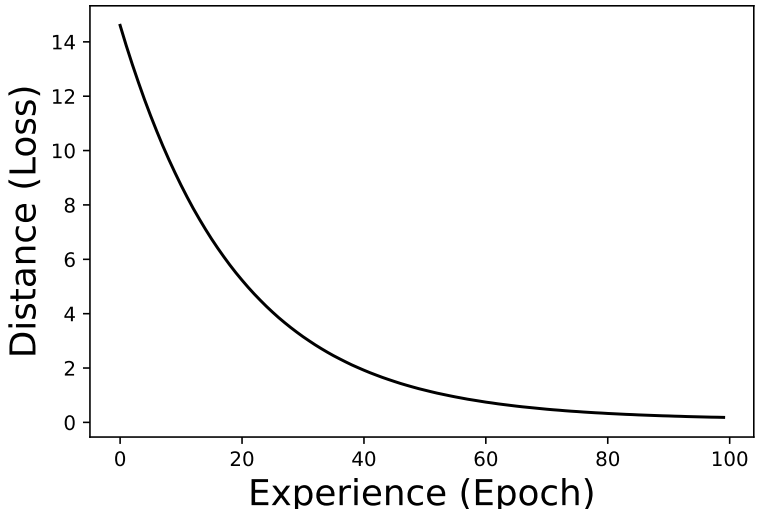




Using a simple Neural Network (pyTorch)

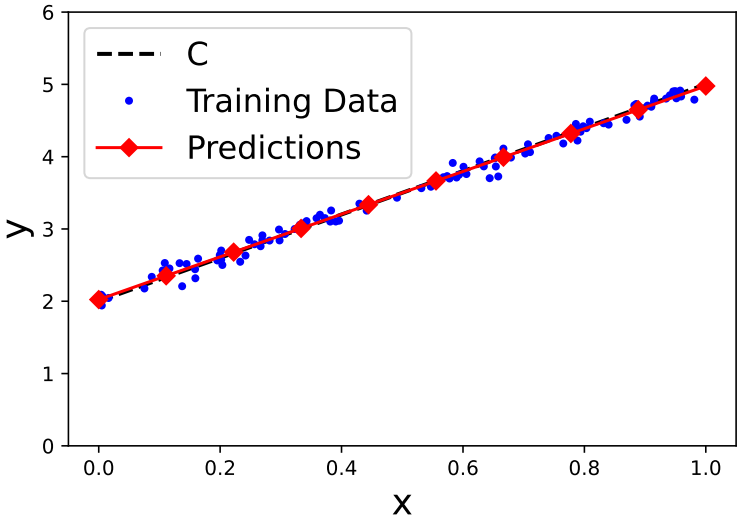
$$y_{pred} = f_{ANN}(x, w)$$

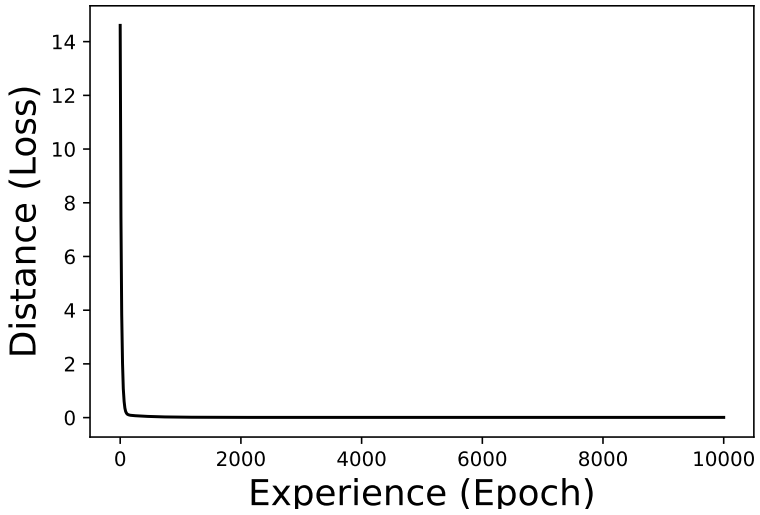




Using a simple Neural Network (pyTorch)

$$y_{pred} = f_{ANN}(x, w)$$





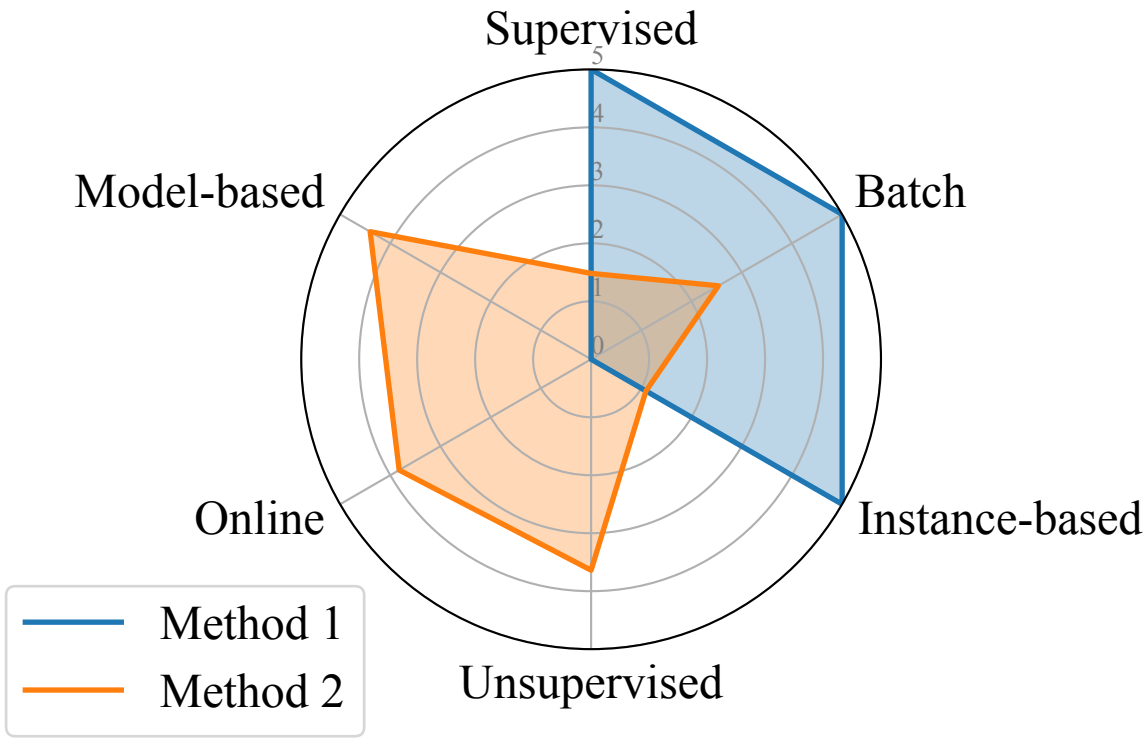
Categorization of ML methods

Categorization of ML methods

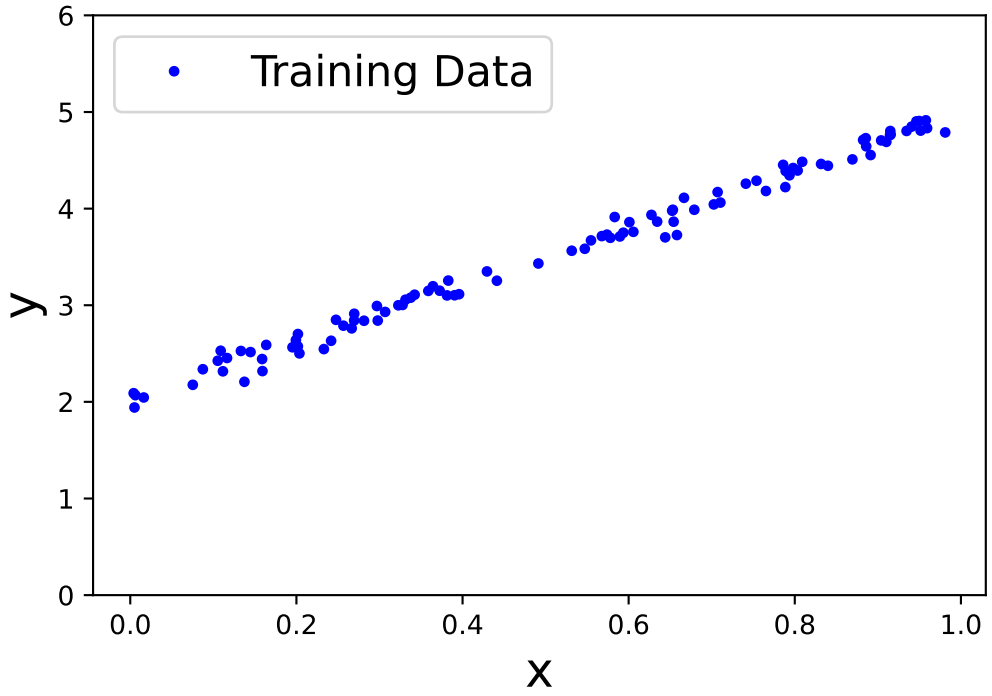
Supervised *vs* Unsupervised

Batch *vs* Online

Instance-based *vs* Model-based


Supervised
Batch
Instance-based
Unsupervised
Online
Model-based
5
4
3
2
1
0
Method 1
Method 2

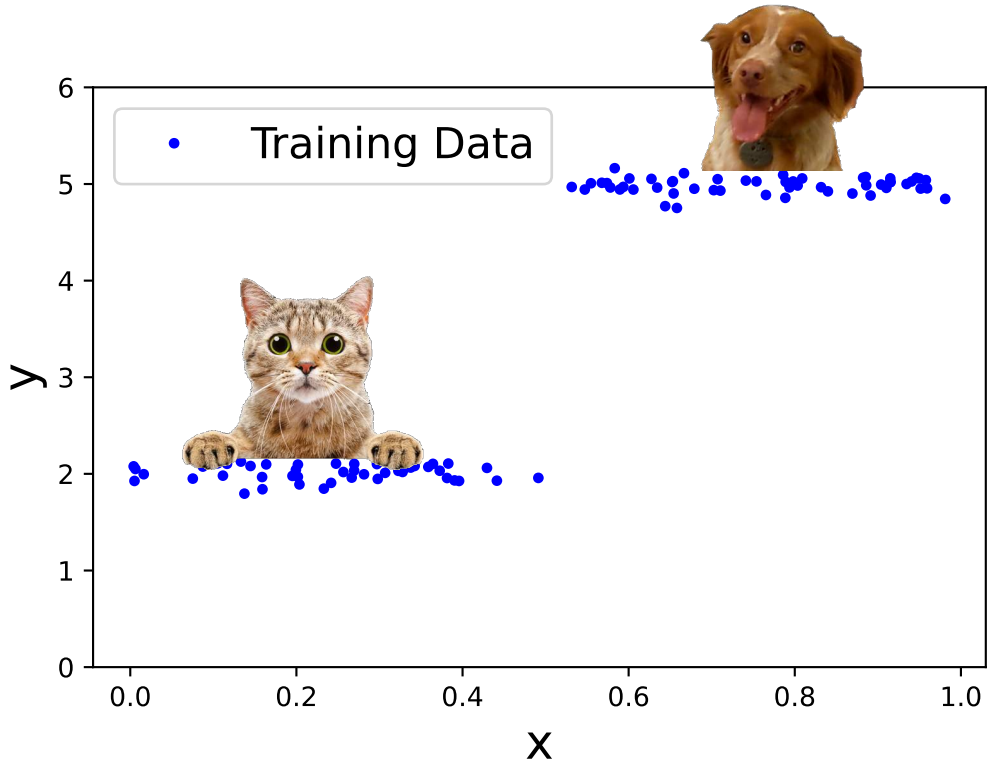
Supervised ML (e.g. regression)



Training data:
At each value x_i we have assigned a value (a label) y_i

$x_1 \rightarrow y_1$
$x_2 \rightarrow y_2$
…
$x_n \rightarrow y_n$

Supervised ML (e.g. classification)



Training data:

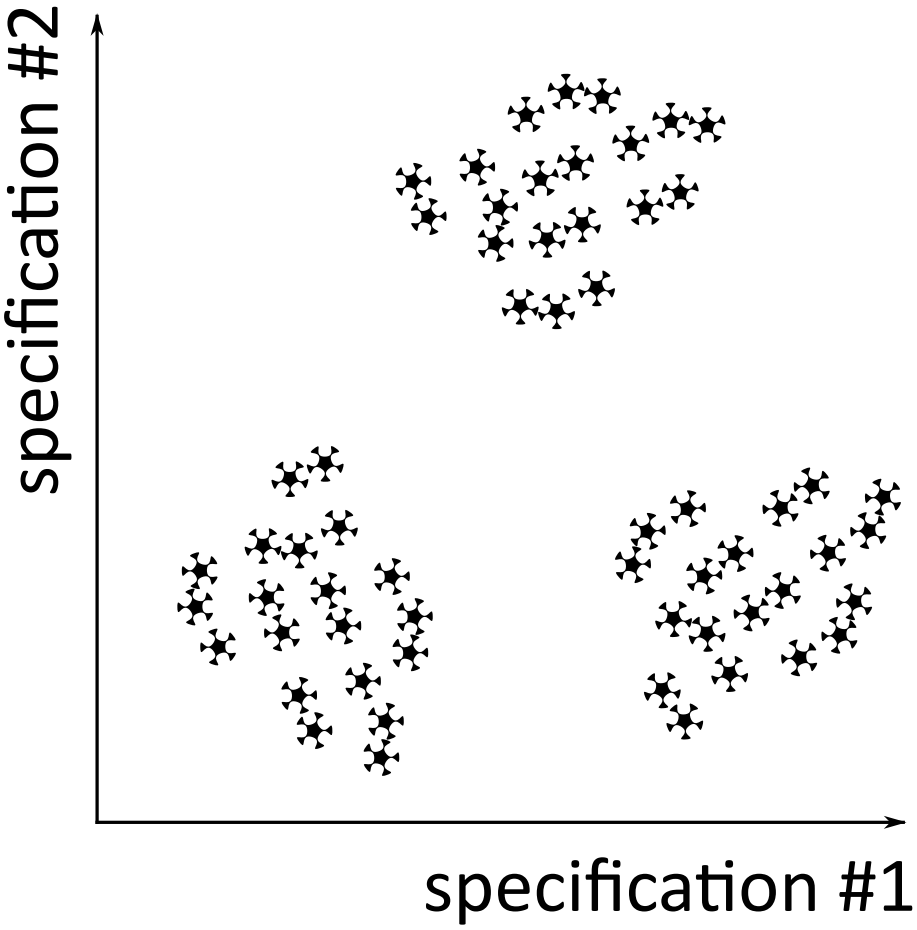
x_i: data instances
(pixels of photos or other characteristics)

y_i: what we call cat/dog

$x_1 \rightarrow$ cat
$x_2 \rightarrow$ dog
...
$x_n \rightarrow$ cat

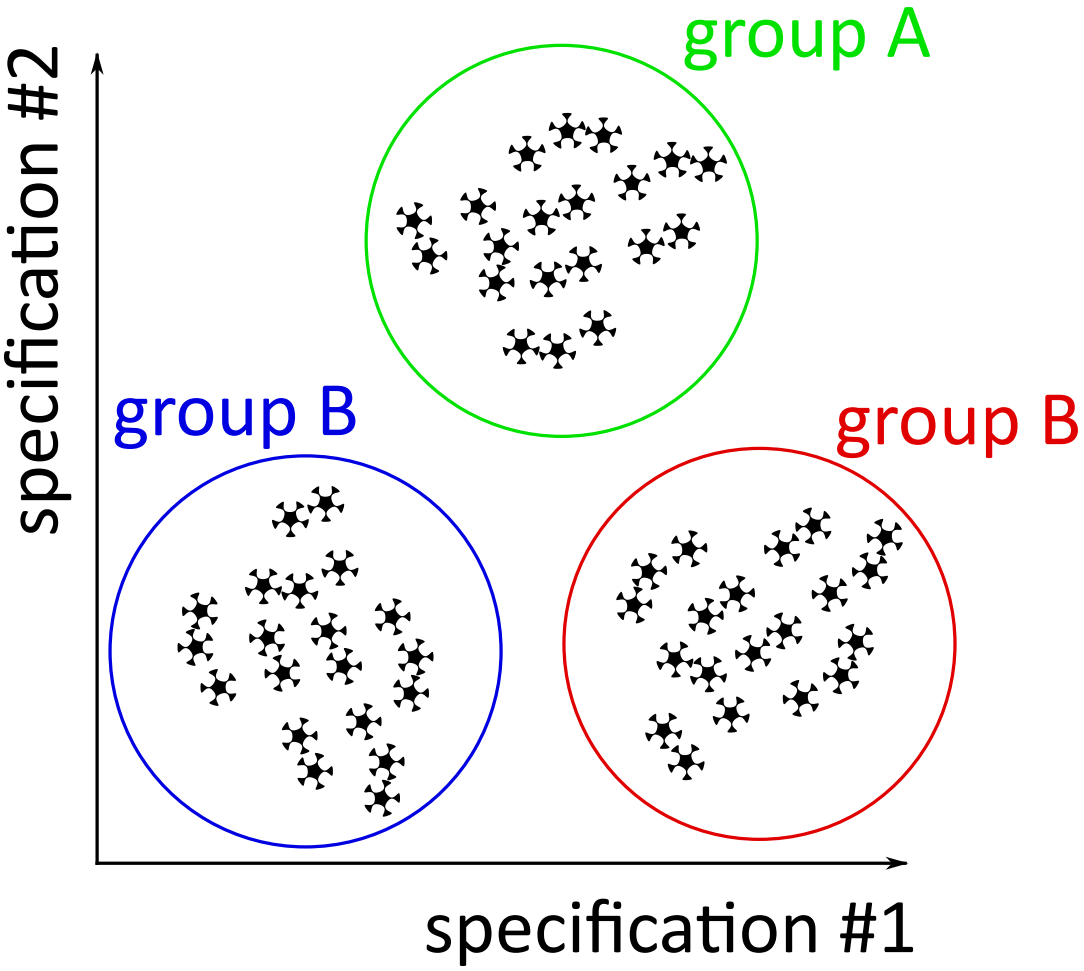
Unsupervised ML (e.g. clustering)

In unsupervised learning there are **no labels**.



Unsupervised ML (e.g. clustering)

In unsupervised learning there are **no labels**.



Clustering based on the distance between points.

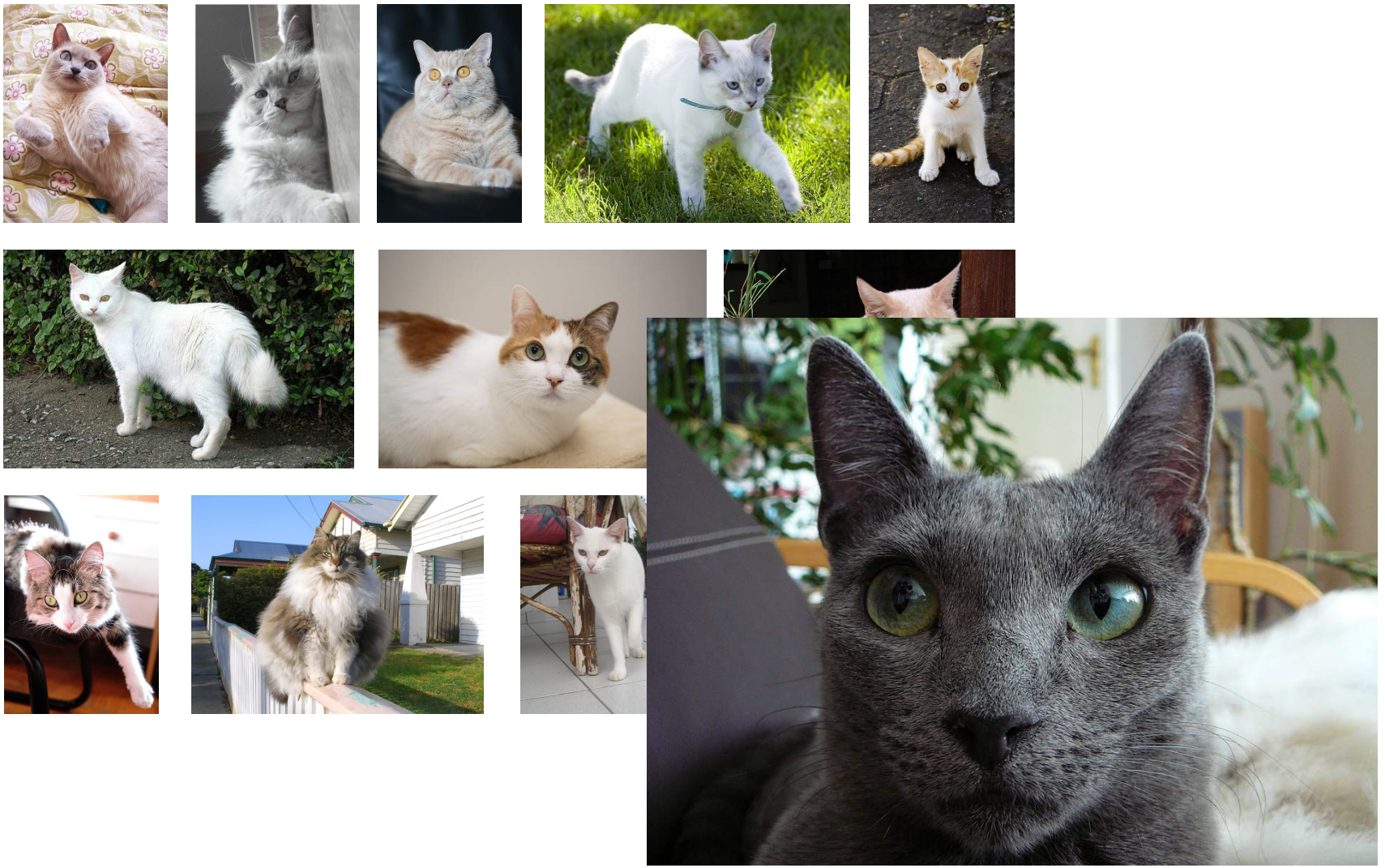
Examples of Supervised & Unsupervised ML methods

	Supervised	Unsupervised
Regression & Classification (linear, logistic, polynomial, ANN, Lasso, Ridge, k-Nearest neighbors, Decision trees, Random forests, SVM etc.)	✓	
Clustering (k-means, hierarchical clustering, Gaussian micture etc.)		✓
Dimensionality reduction (autoencoders, PCA, compression etc.)		✓
Anomaly detection		✓
…		

A case in between: Reinforcement Learning

Batch and online ML

Batch and online ML



Examples of Batch & Online ML methods

	Batch	Online
Linear regression	✓	
Polynomial regression	✓	
Lasso	✓	
ANN		✓
Autoencoders		✓
Reinforcement Learning		✓
...		

Instance-based and model-based ML

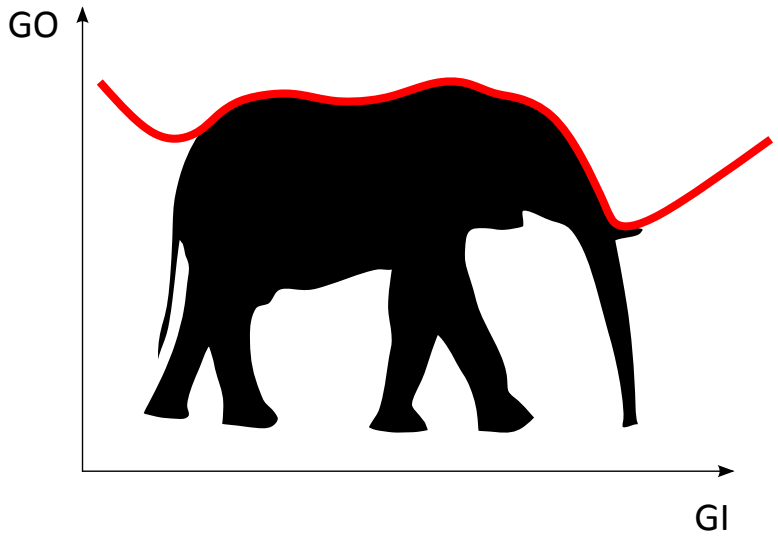
Instance-based
(also called memory-based or lazy)

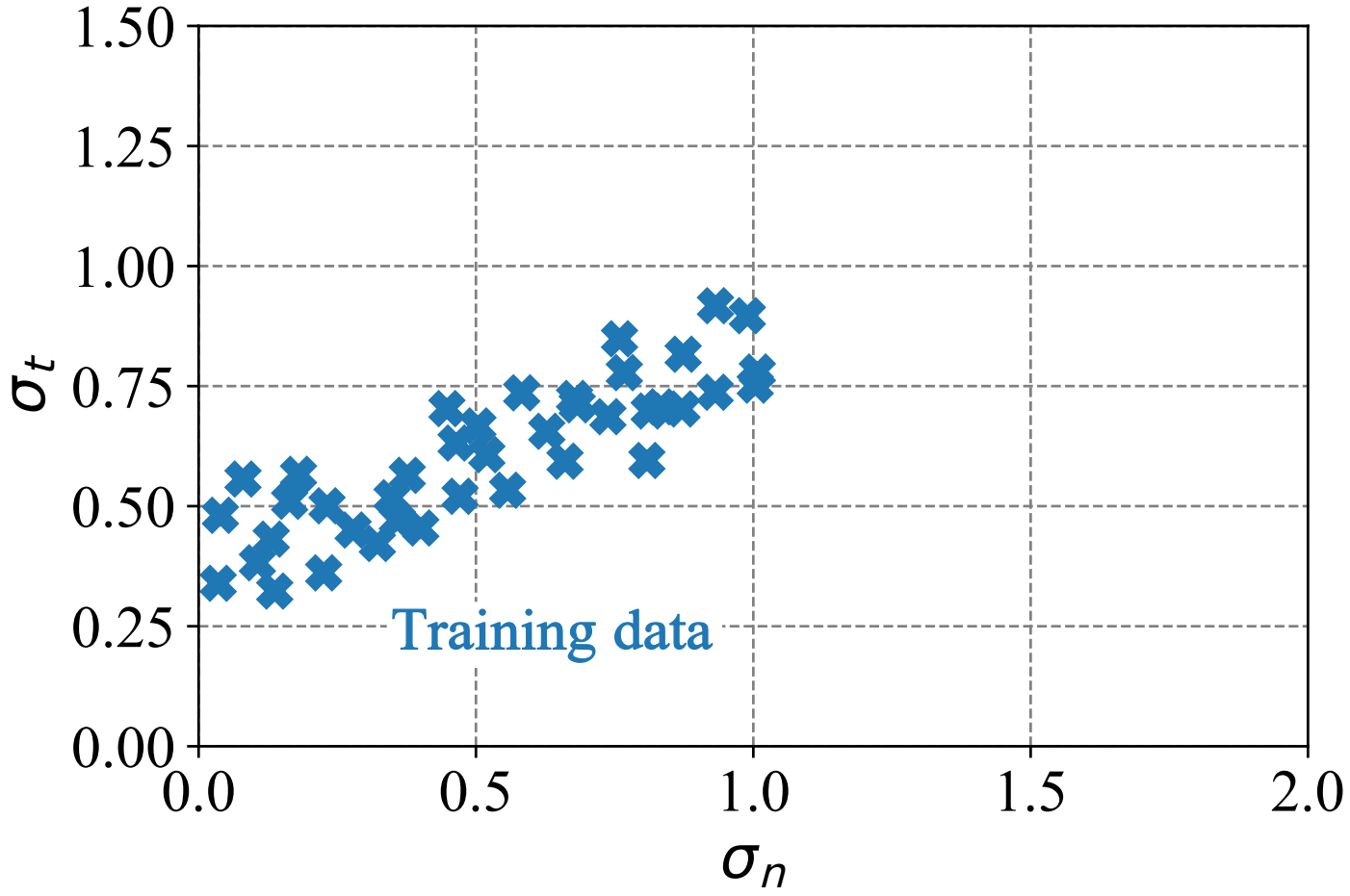
Model-based
(also called physics-based)

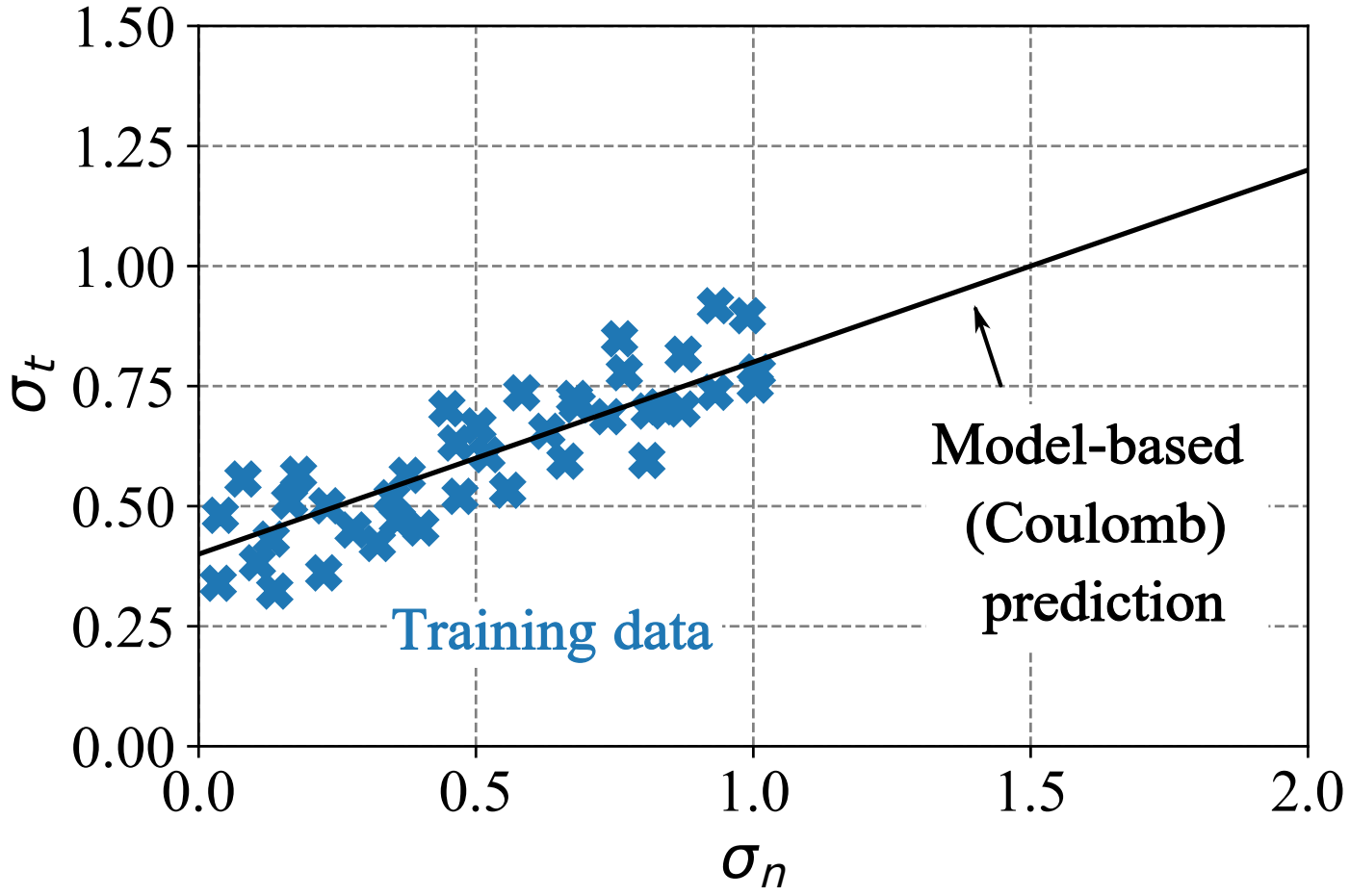
Instance-based and model-based ML

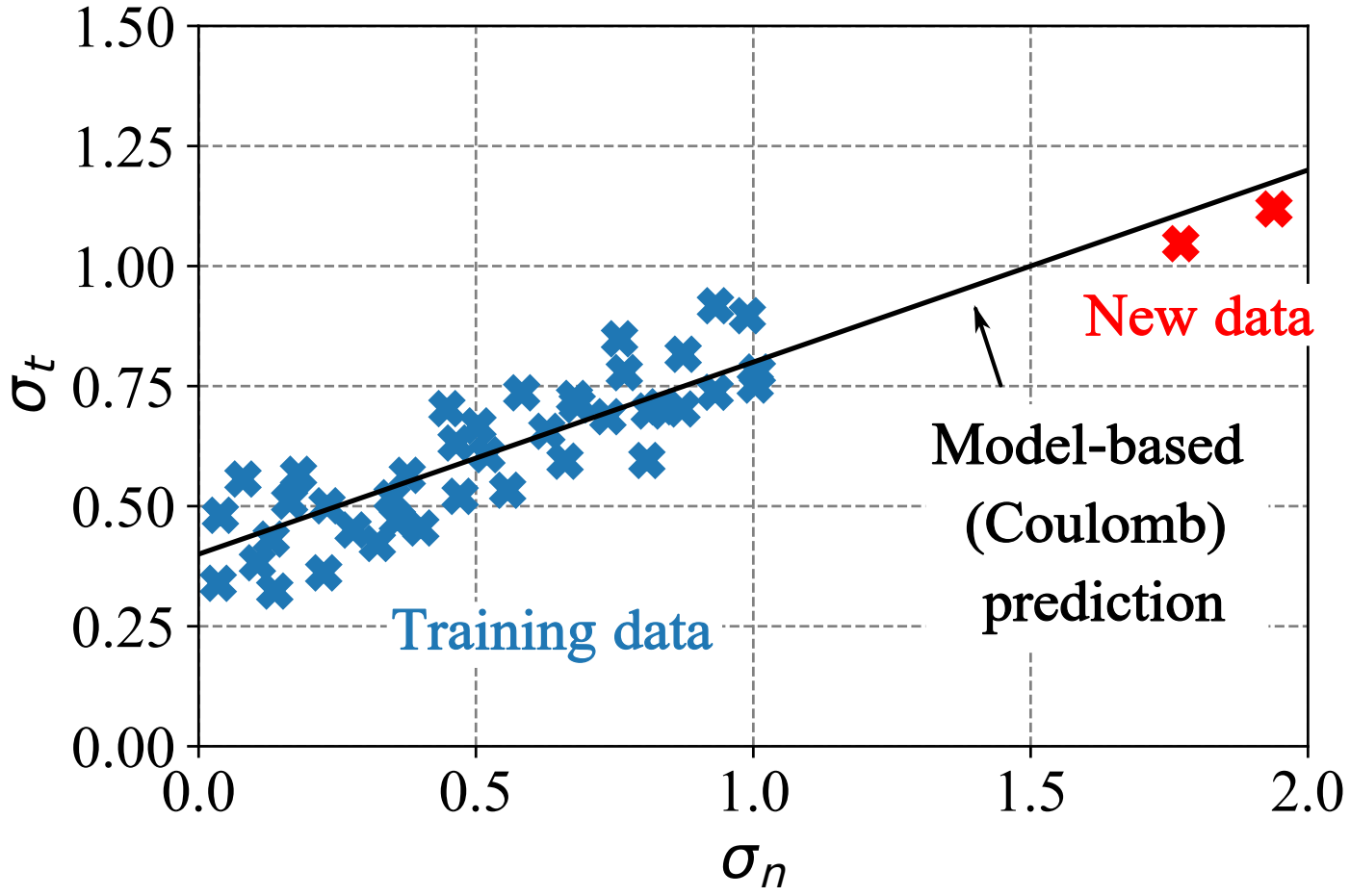
Instance-based
(also called memory-based or lazy)

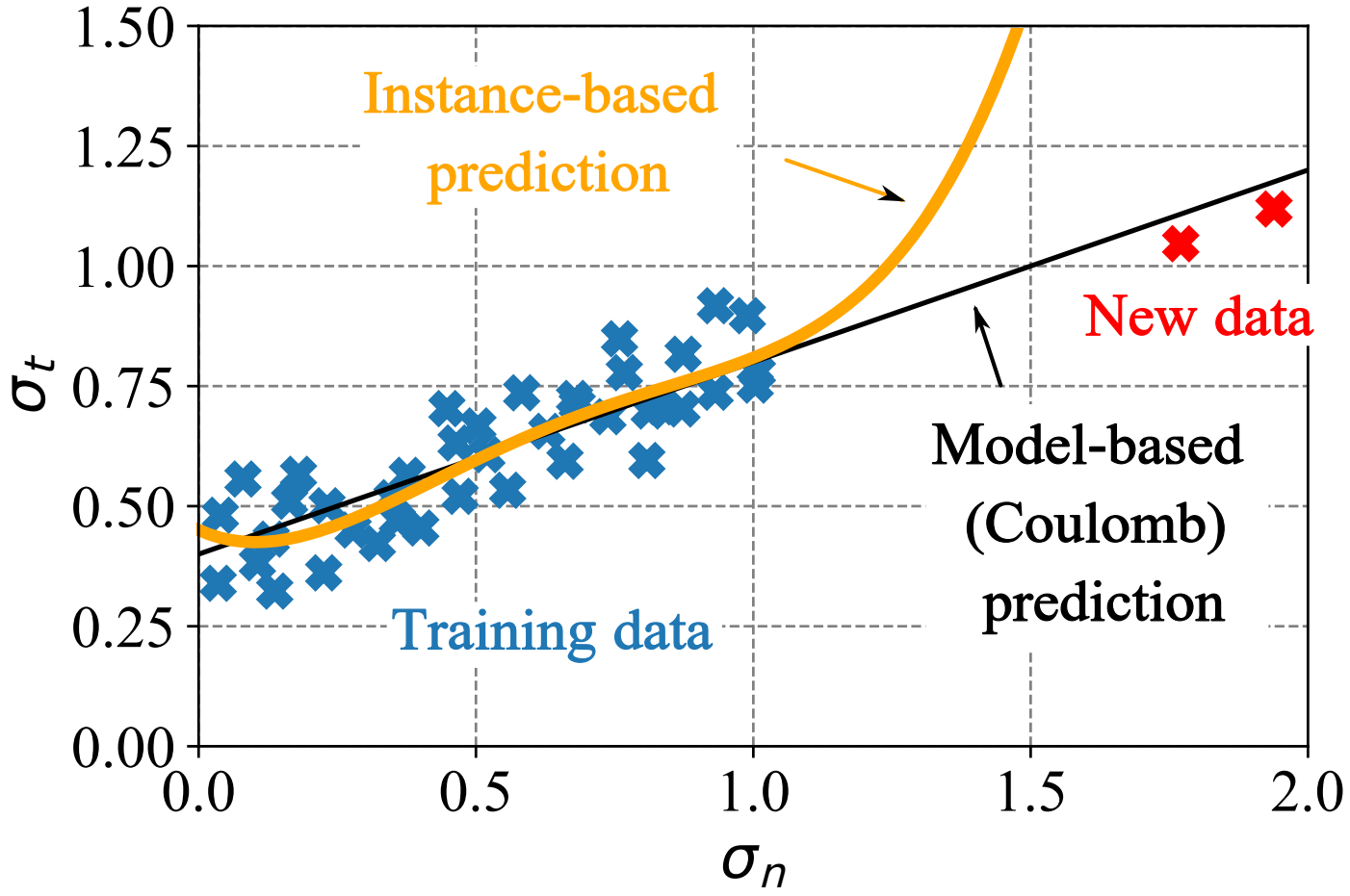
Model-based
(also called physics-based)




1.50
1.25
1.00
0.75
0.50
0.25
0.00
σ_t
0.0
0.5
1.0
1.5
2.0
σ_n
Training data


1.50
1.25
1.00
0.75
0.50
0.25
0.00
σ_t
0.0
0.5
1.0
1.5
2.0
σ_n
Training data
Model-based
(Coulomb)
prediction


1.50
1.25
1.00
0.75
0.50
0.25
0.00
σ_t
0.0
0.5
1.0
1.5
2.0
σ_n
New data
Model-based
(Coulomb)
prediction
Training data


Instance-based
prediction
New data
Model-based
(Coulomb)
prediction
Training data
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0.0
0.5
1.0
1.5
2.0
σ_t
σ_n

Examples of instance- & physics-based ML

	Instance-based	Physics-based
Artificial Neural Networks	✓	
Decision trees	✓	
Random forests	✓	
k-Nearest neighbors	✓	
Principal Component Analysis (PCA)	✓	
Thermodynamics-based ANN		✓
“Data-driven” mechanics		✓

ML and Geomechanics

ML and geomechanics

- Constitutive modeling
- Geotechnics
- Geophysics
- Image corelation

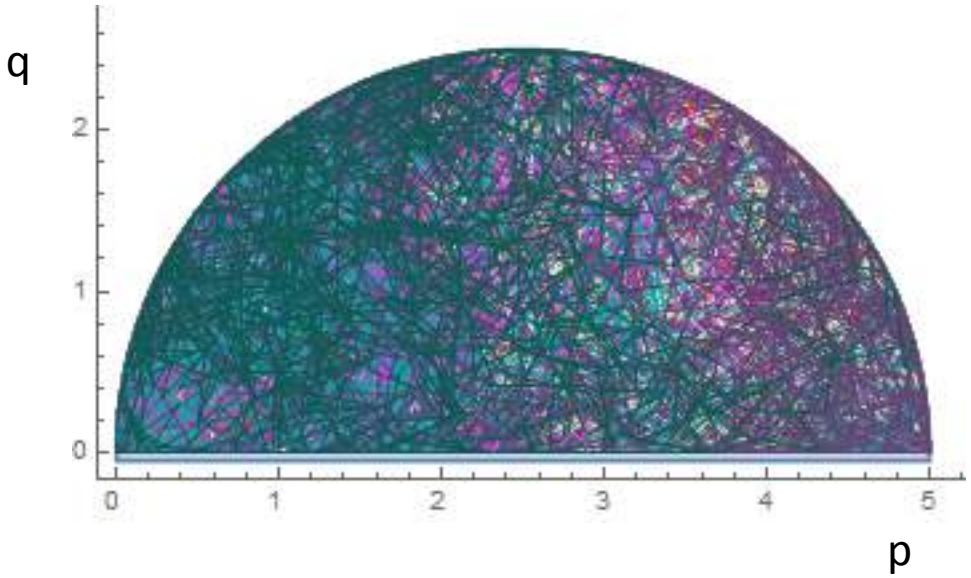
ML for Constitutive modeling

- Fit data with existing models
  (see Pin et al. 2021 for a review for soils;
  Bahmani et al. 2023 & Linka et al. 2023 for closed form approaches)

- Derive constitutive descriptions from data at multiple scales and identify (new) state variables

Example: Brute-force fitting

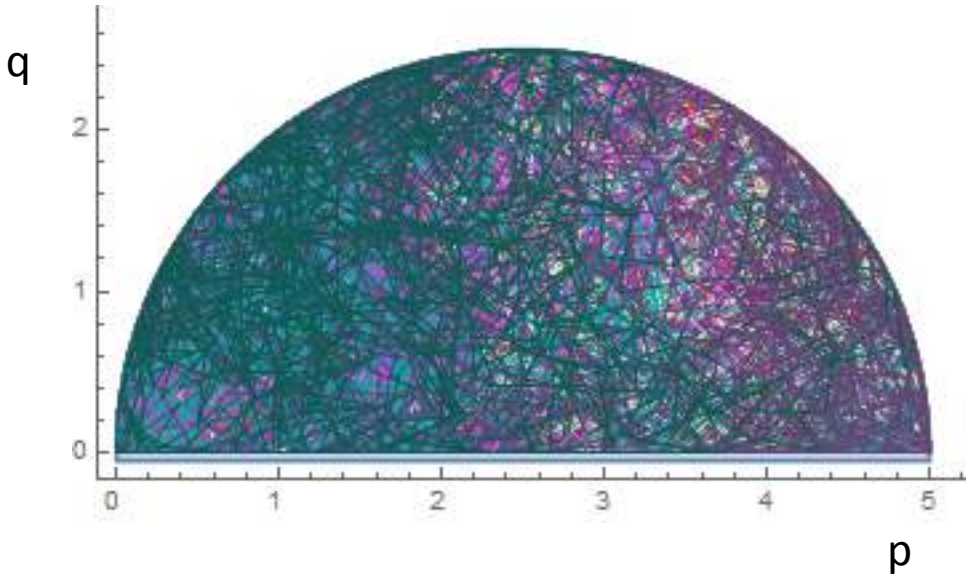
Training with random stress paths



Modified Cam-clay yield surface

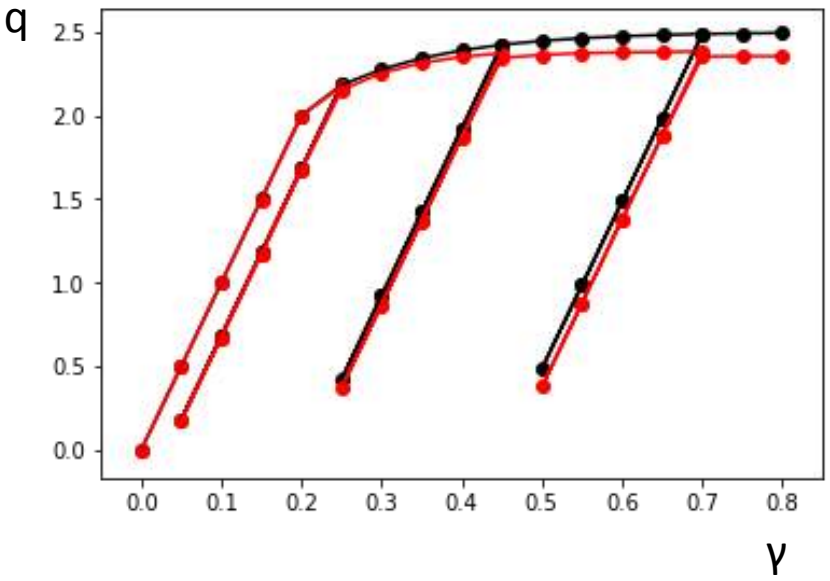
Example: Brute-force fitting

Training with random stress paths



Modified Cam-clay yield surface

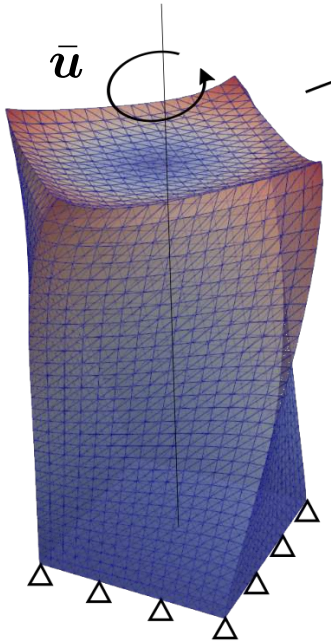
ANN predictions (red)



ML for Constitutive modeling & multi-scale

Macro-scale

Boundary Value Problem (BVP)



F

Strain

FE^2
FEM×DEM
FEM×...

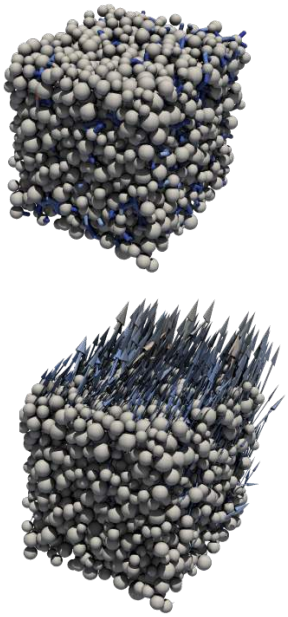
bottleneck
∀ point, time

Stress

S

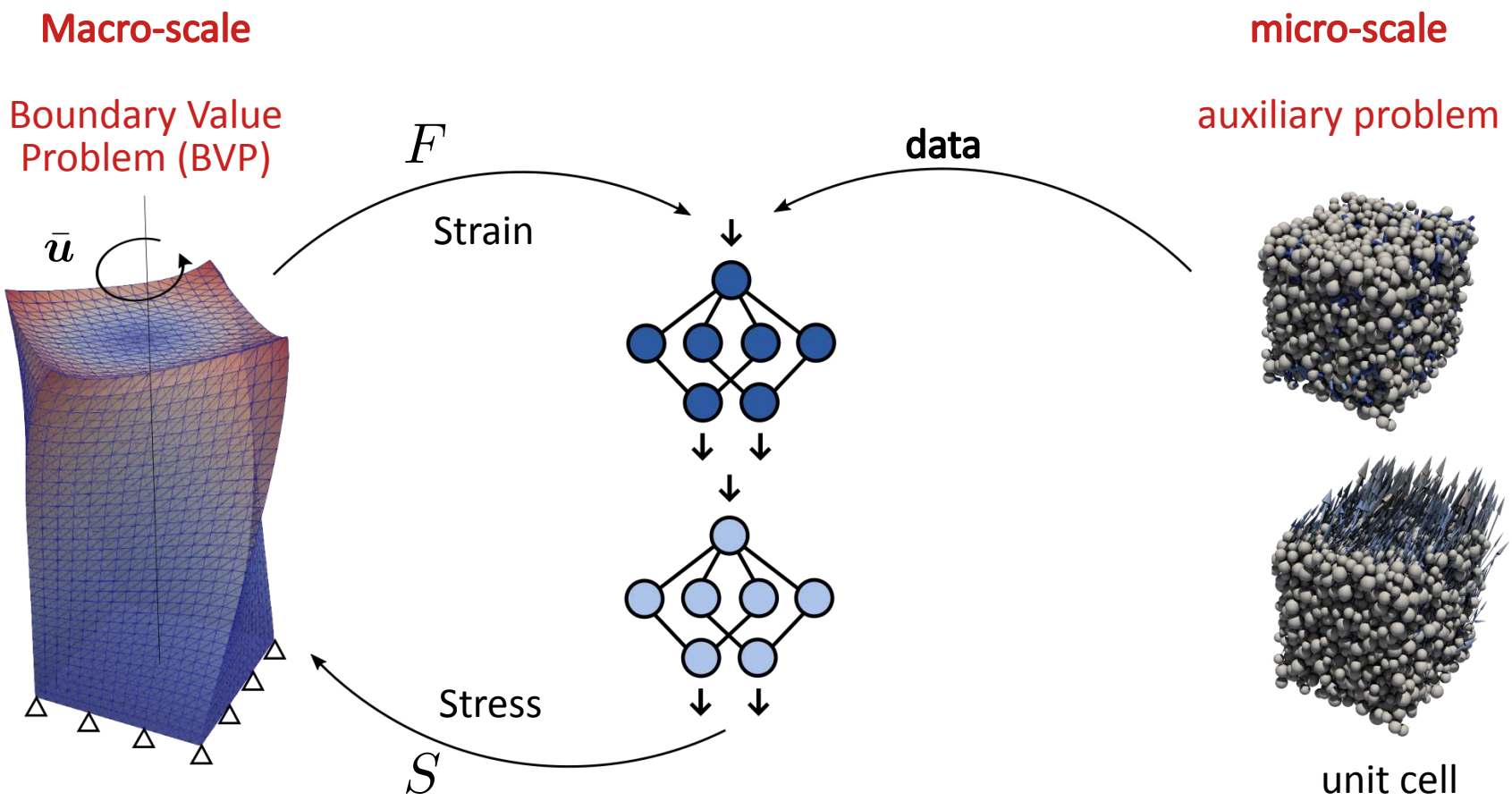
micro-scale

auxiliary problem



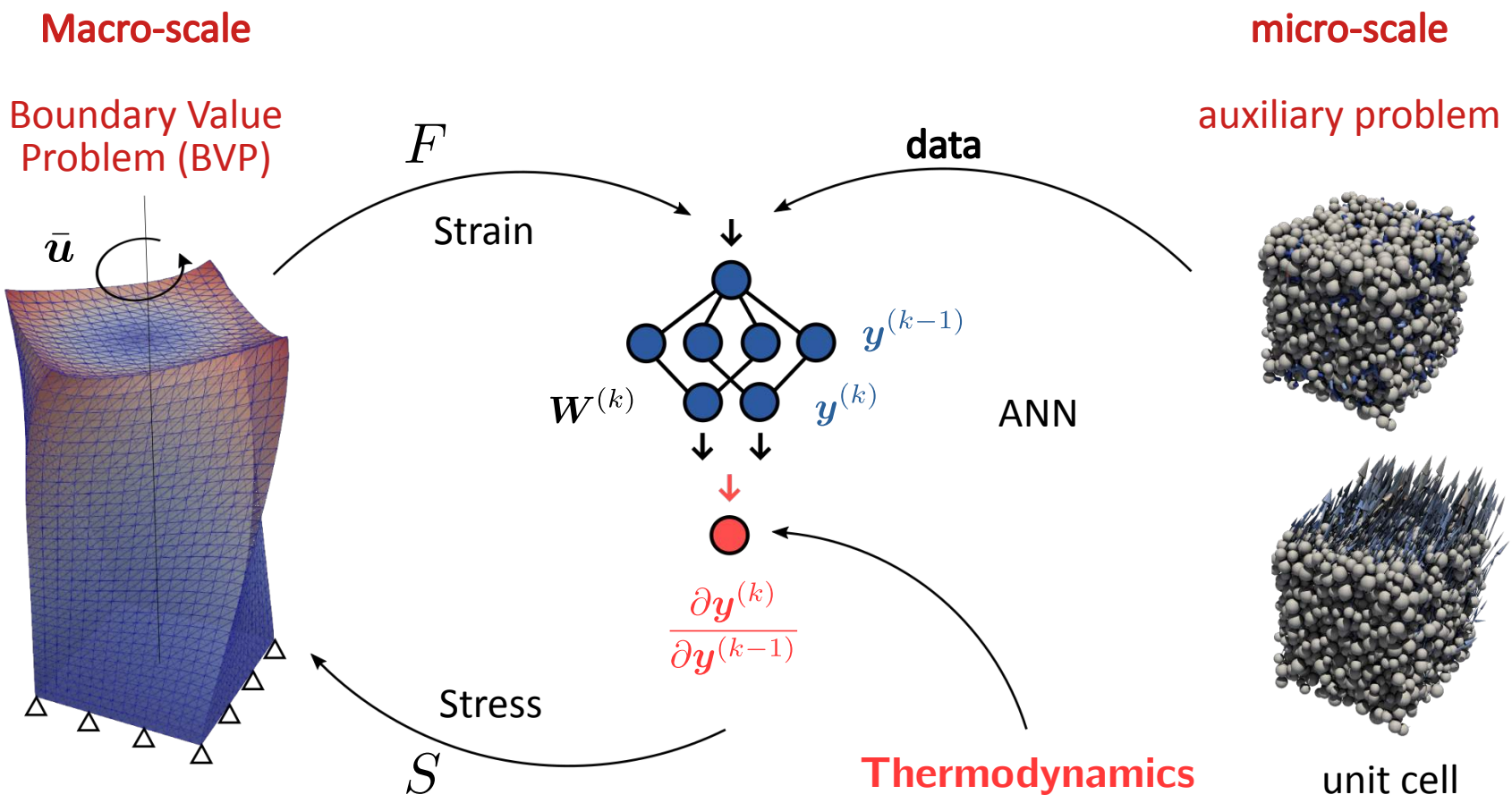
unit cell

Example #1: Multi-scale ANN



(Ghaboussi et al. 1991; Lefik and Schrefler 2003; Wang & Sun 2018; Mozaffar et al. 2019; Mianroodi et al. 2021 among many others)

Example #2: Thermodynamics-based ANN (TANN)

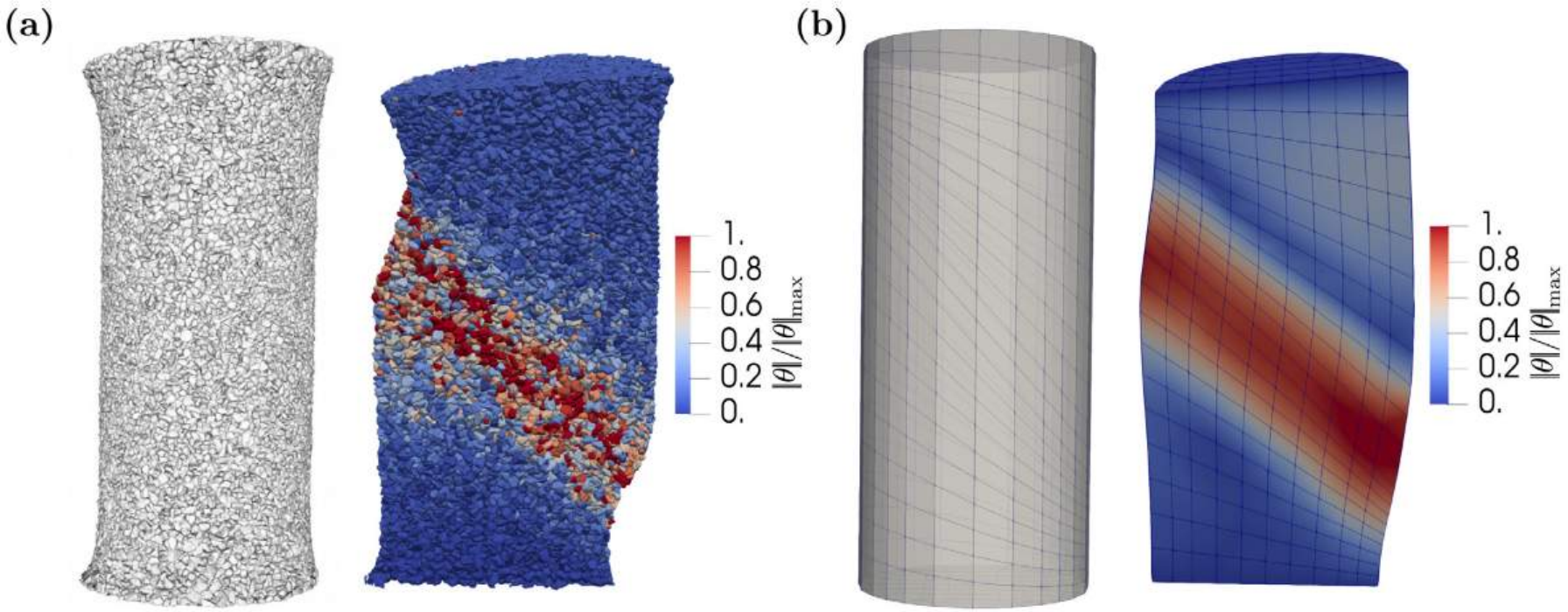


(Masi et al. 2021; Masi & Stefanou 2022; Masi & Stefanou 2023)

How?

See the program

Example #3: Data-Driven computational mechanics

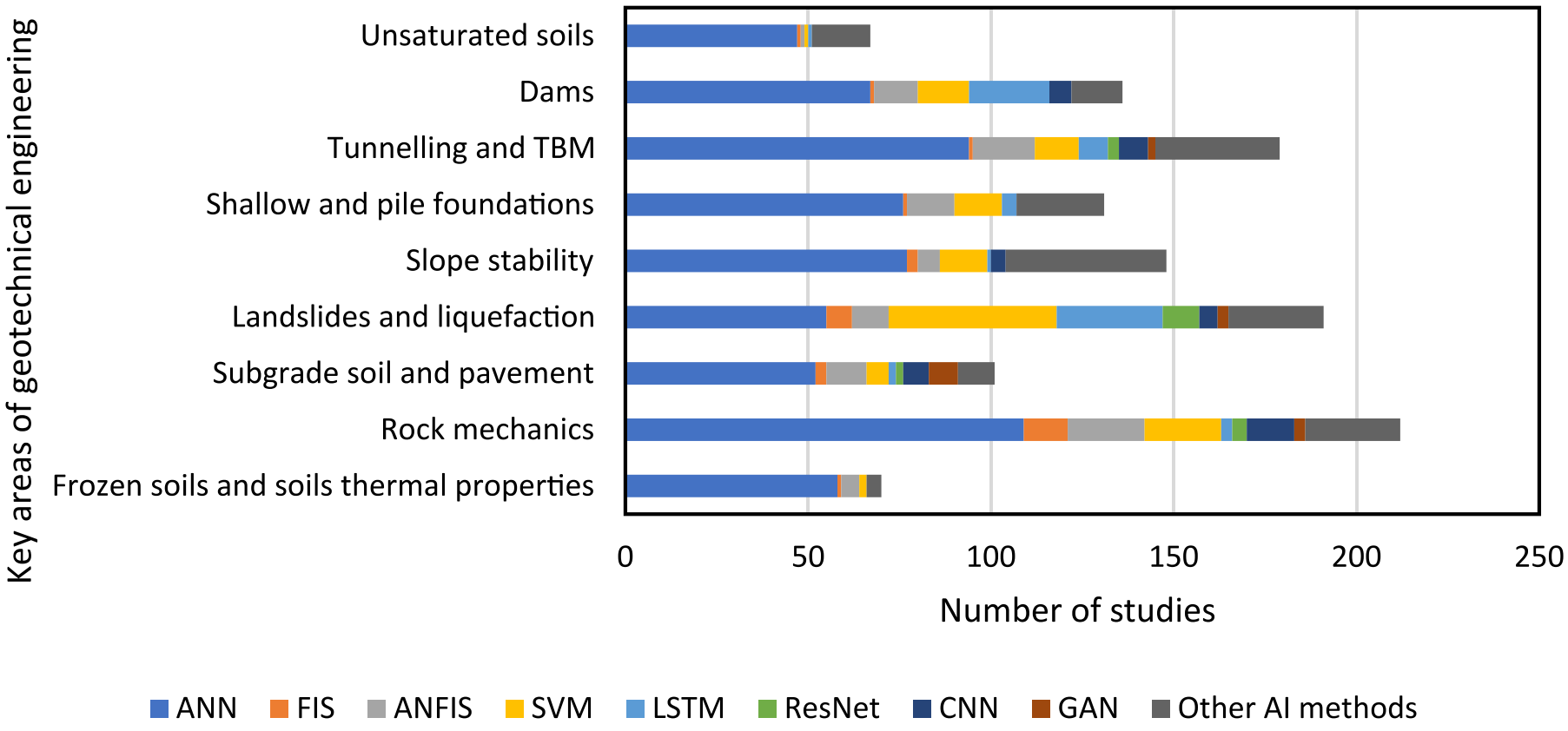


(Karapiperis et al. 2022)

How?

See the program

ML in Geotechnics



(Baghbani et al. 2022)

Example: Predicted periodic displacement for the Baishuihe landslide

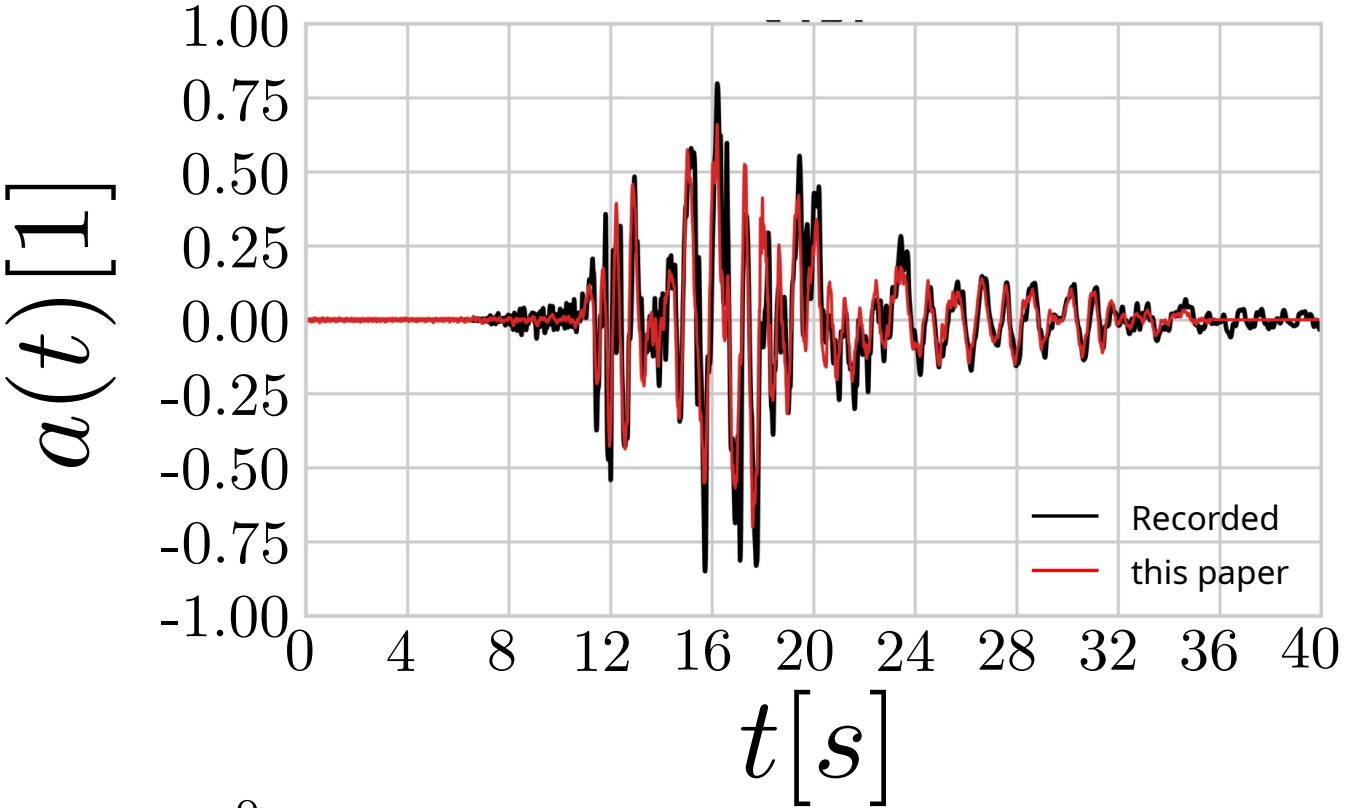
(Li et al. 2021)

Study	ML (or hybrid ML) algorithms*	Performance metrics**				
		MAPE/%	MAE/mm	RMSE/mm	MSE/mm	R
Li et al. (2020)	DBN	5.05	2.78	3.48	–	–
Liu et al. (2020a, b, c)	LSTM	5.0	–	7.5	–	–
Liu et al. (2020a, b, c)	Hybrid SVR-K-means-LSTM	–	3.86	6.76	–	–
Li et al. (2019)	ELM	–	–	17.41	–	0.968
Yang et al. (2019)	LSTM	10.43	–	7.11	–	–
Li et al. (2018)	Hybrid LASSO-ELM	1.30	26.44	35.63	2.28	–
Zhou et al. (2018a)	Hybrid PSO-KELM	0.083	18.104	–	–	0.983

(Tehrani et al. 2022)

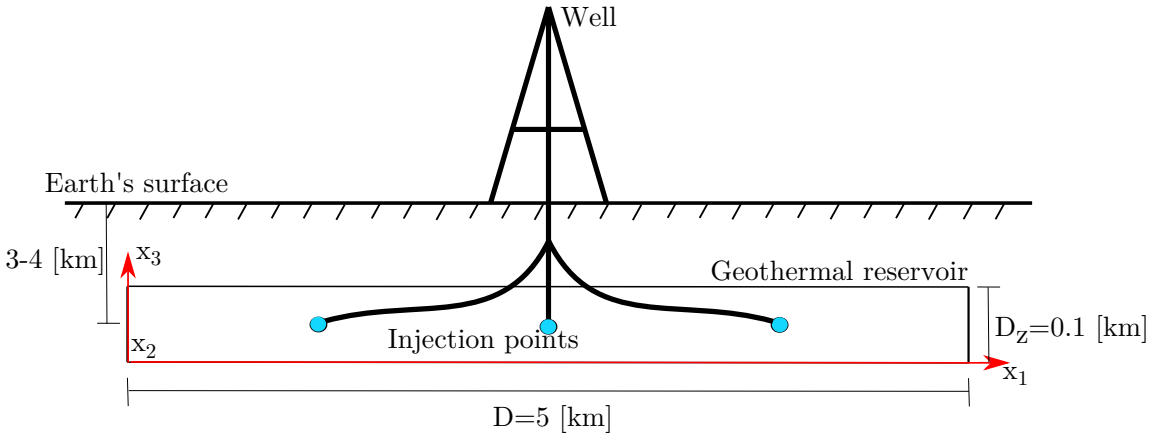
ML in Geophysics

Example #1: Creating realistic accelerograms



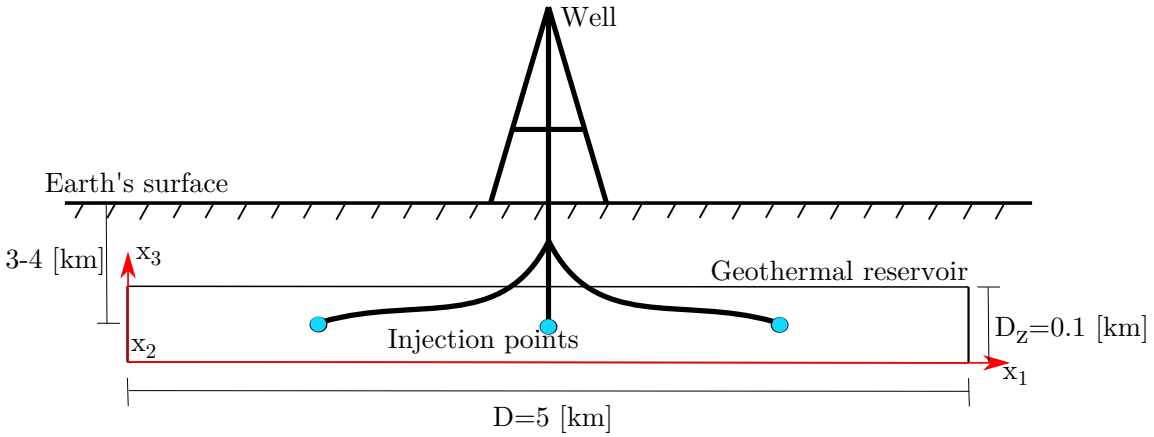
(Gatti & Clouteau 2020)

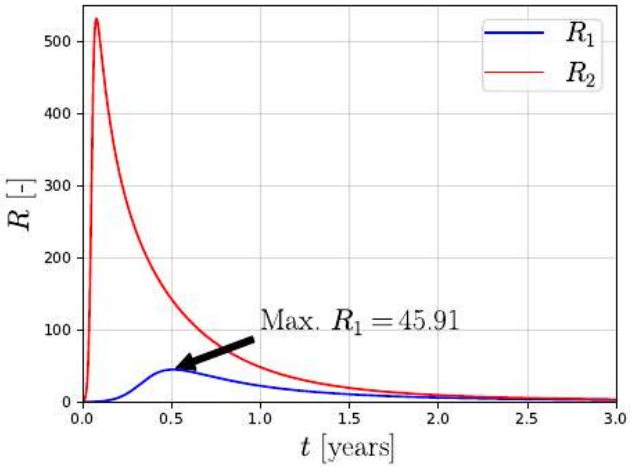
Example #2: Preventing induced seismicity in a Geothermal Reservoir with RL

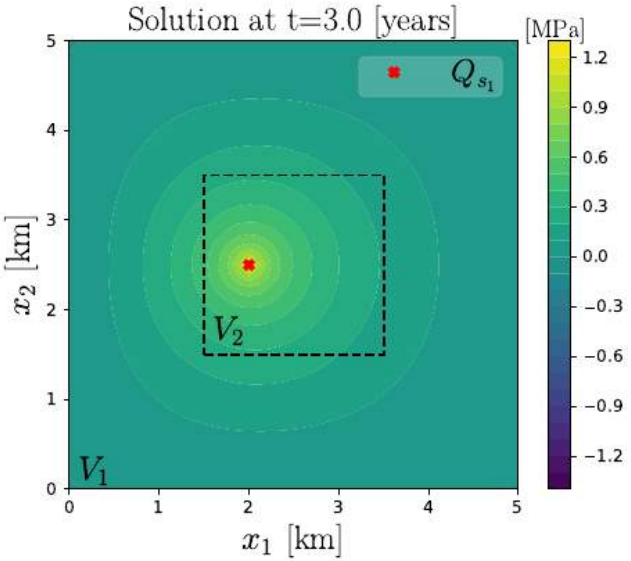


(Gutiérrez et al. 2023)

Example #2: Preventing induced seismicity in a Geothermal Reservoir with RL

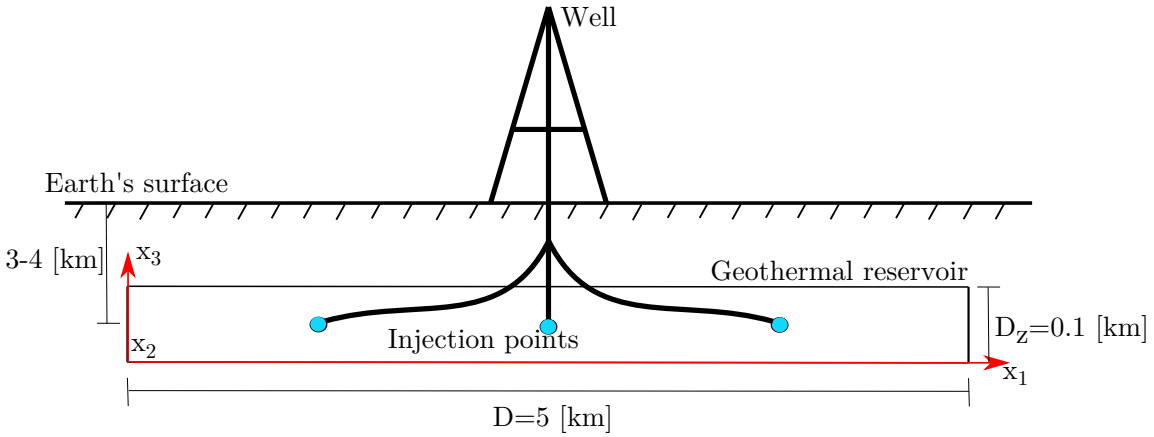


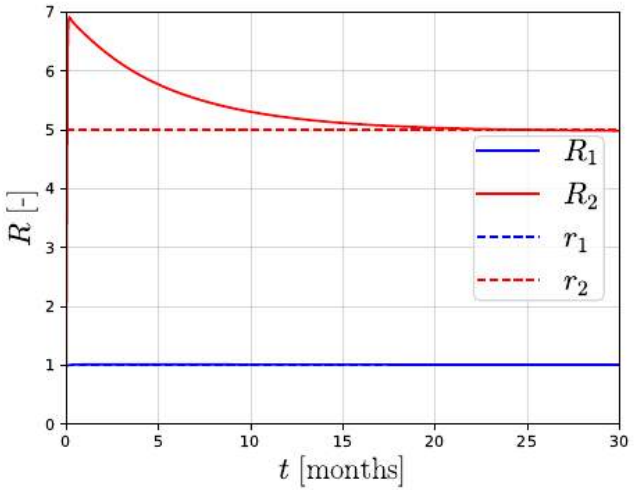


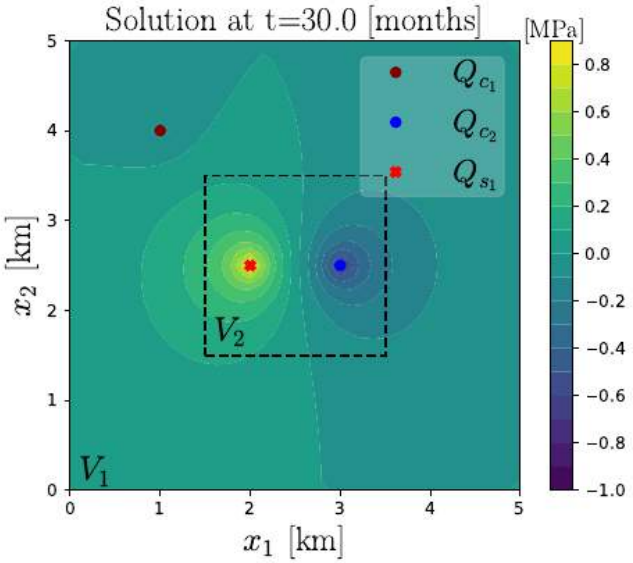


(Gutiérrez et al. 2023)

Example #2: Preventing induced seismicity in a Geothermal Reservoir with RL







(Gutiérrez et al. 2023)

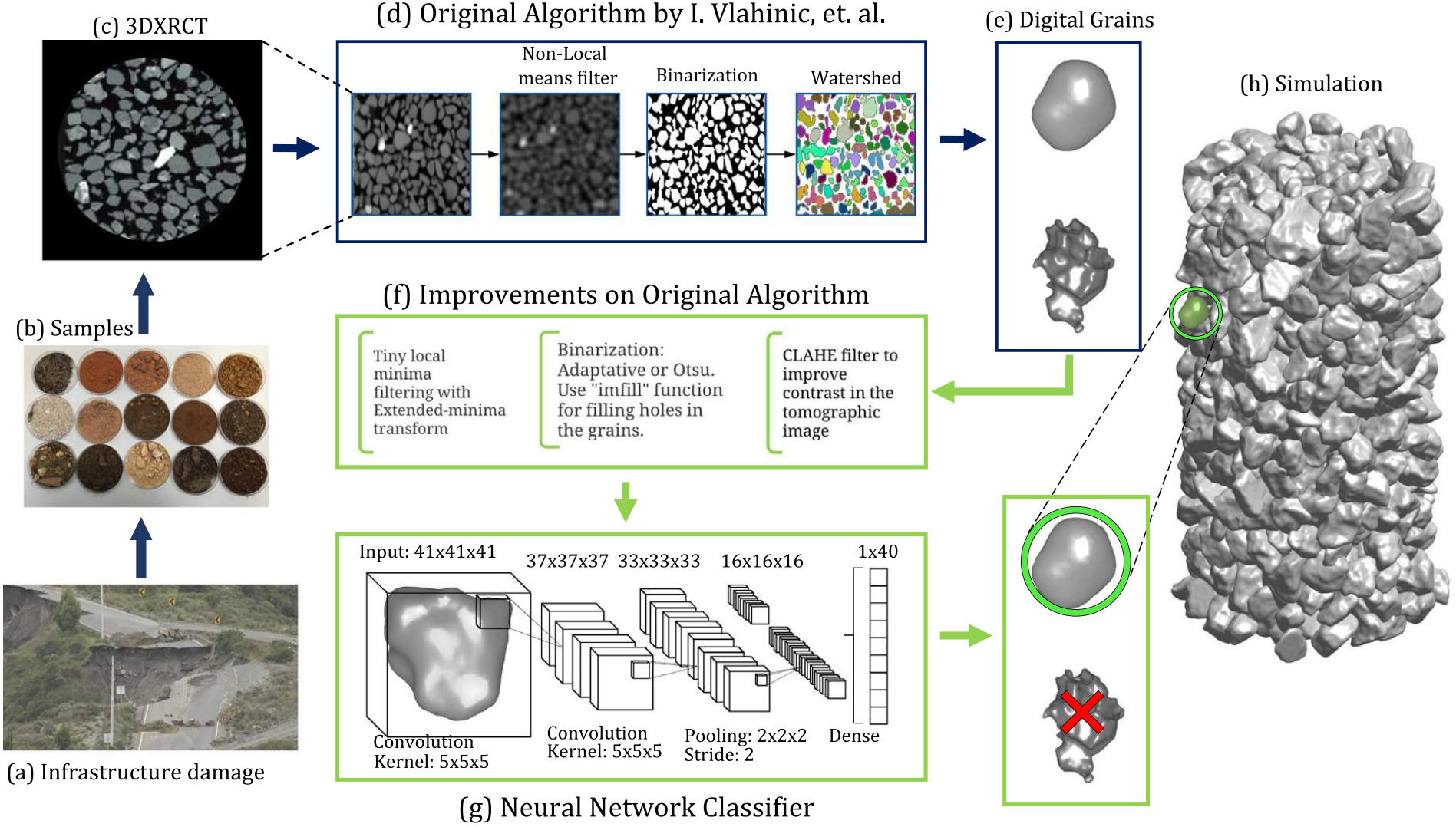
ML in Image correlation

Step 1: see ALERT Doctoral School 2022 for state of the art techniques in image correlation

Step 2: explore M

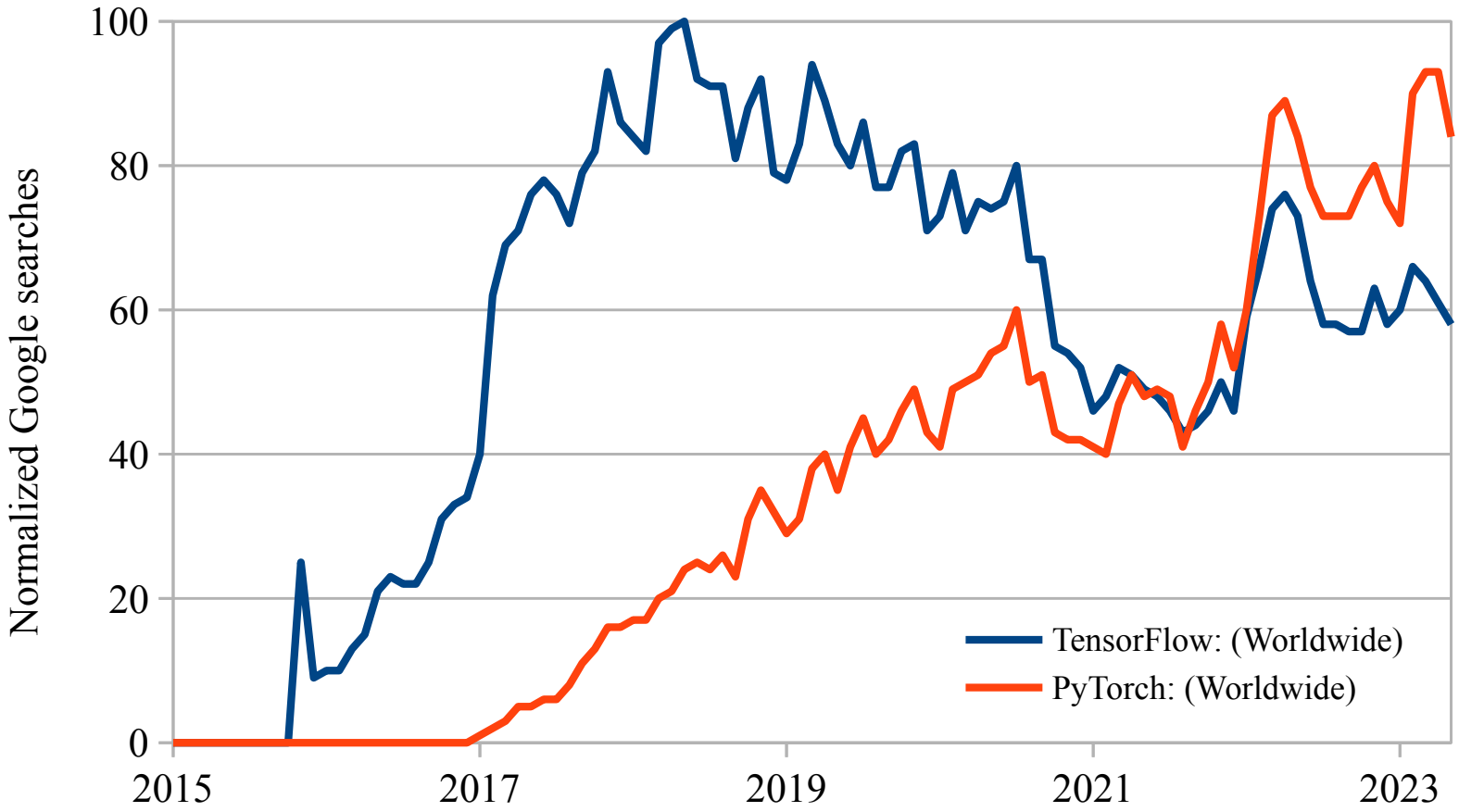
ML in Image correlation

(Cevallos et al. 2023)



Libraries for ML

Libraries for ML



Other libraries: https://en.wikipedia.org/wiki/Machine_learning#Software

PyTorch

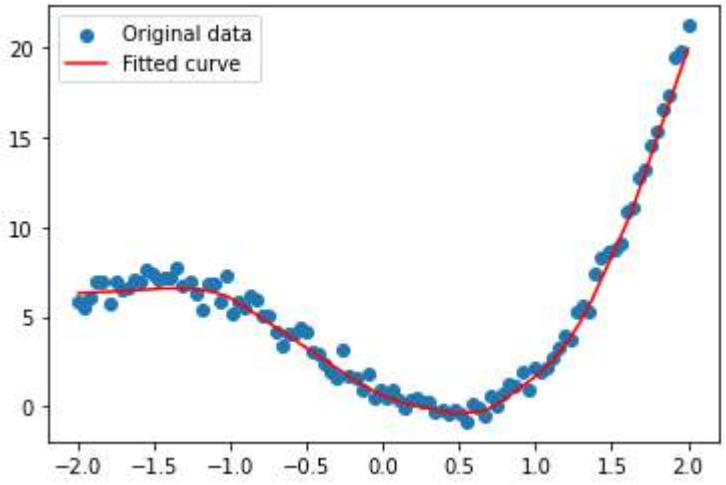
```python
import torch
import torch.nn as nn
import torch.optim as optim
import numpy as np
import matplotlib.pyplot as plt

# Generate synthetic data
np.random.seed(0)
x = np.linspace(-2, 2, 100)
y = 2 * x**3 + 3 * x**2 - 4 * x + 1 + 0.5 * np.random.randn(100)

# Convert data to PyTorch tensors
x = torch.tensor(x, dtype=torch.float32).view(-1, 1)
y = torch.tensor(y, dtype=torch.float32).view(-1, 1)

# Define a neural network model with 2 hidden layers and nonlinear activati
class CubicRegressionModel(nn.Module):
    def __init__(self):
        super(CubicRegressionModel, self).__init__()
        self.fc1 = nn.Linear(1, 64)  # Input layer to hidden layer 1
        self.relu1 = nn.ReLU()       # ReLU activation
        self.fc2 = nn.Linear(64, 64) # Hidden layer 1 to hidden layer 2
        self.relu2 = nn.ReLU()       # ReLU activation
        self.fc3 = nn.Linear(64, 1)  # Hidden layer 2 to output layer

    def forward(self, x):
        x = self.fc1(x)
        x = self.relu1(x)
        x = self.fc2(x)
        x = self.relu2(x)
        x = self.fc3(x)
        return x

# Initialize the model and optimizer
model = CubicRegressionModel()
optimizer = optim.SGD(model.parameters(), lr=0.01)
criterion = nn.MSELoss()

# Training loop
num_epochs = 1000
for epoch in range(num_epochs):
    # Forward pass
    outputs = model(x)
    loss = criterion(outputs, y)

    # Backward pass and optimization
    optimizer.zero_grad()
    loss.backward()
    optimizer.step()

    if (epoch + 1) % 100 == 0:
        print(f'Epoch [{epoch + 1}/{num_epochs}], Loss: {loss.item():.4f}')
```

TensorFlow

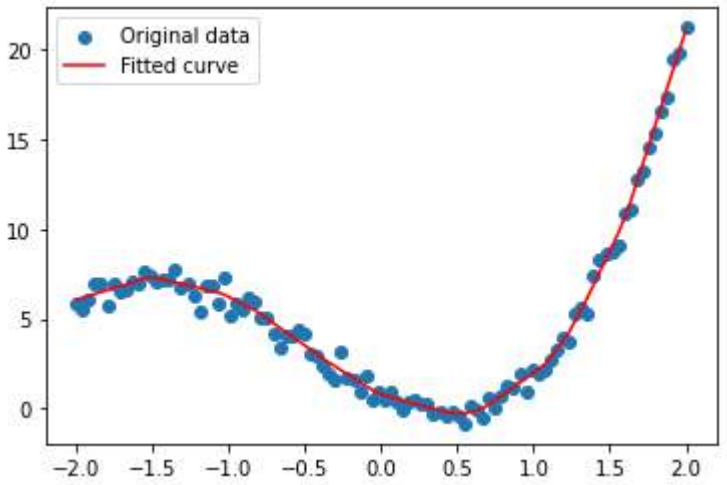
```python
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt

# Generate synthetic data
np.random.seed(0)
x = np.linspace(-2, 2, 100)
y = 2 * x**3 + 3 * x**2 - 4 * x + 1 + 0.5 * np.random.randn(100)

# Convert data to TensorFlow tensors
x_tf = tf.constant(x, dtype=tf.float32)
y_tf = tf.constant(y, dtype=tf.float32)

# Define a neural network model with 2 hidden layers and nonlinear a
model = tf.keras.Sequential([
    tf.keras.layers.Dense(64, input_shape=(1,), activation='relu'),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(1, activation='linear')
])

# Compile the model
model.compile(optimizer='sgd', loss='mean_squared_error')

# Train the model
history = model.fit(x_tf, y_tf, epochs=1000, verbose=1)

# Plot the results
predicted = model.predict(x_tf)
plt.scatter(x, y, label='Original data')
plt.plot(x, predicted, label='Fitted curve', color='red')
plt.legend()
plt.show()
```

PyTorch

```
53
54 # Plot the results
55 predicted = model(x).detach().numpy()
56 plt.scatter(x.numpy(), y.numpy(), label='Original data')
57 plt.plot(x.numpy(), predicted, label='Fitted curve', color='red')
58 plt.legend()
59 plt.show()
```



TensorFlow

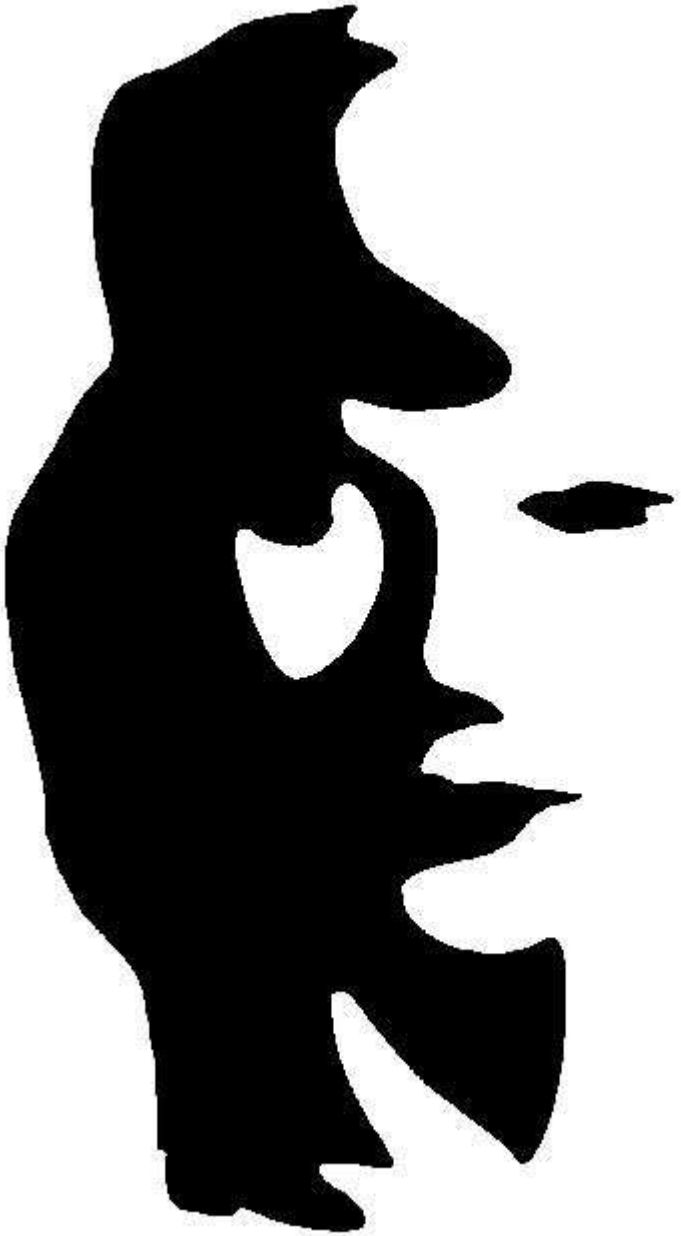


> show me an example of regression of a cubic function with pytorch and tensorflow using 2 layers and nonlinear activation functions

Bias in ML and limitations

Cookies or dogs?

(from Cino & Instagram)



(eBaumsWorld)

Bias in ML and limitations

SCIENCE ADVANCES | RESEARCH ARTICLE

RESEARCH METHODS

The accuracy, fairness, and limits of predicting recidivism

Julia Dressel and Hany Farid*

Algorithms for predicting recidivism are commonly used to assess a criminal defendant's likelihood of committing a crime. These predictions are used in pretrial, parole, and sentencing decisions. Proponents of these systems argue that big data and advanced machine learning make these analyses more accurate and less biased than humans. We show, however, that the widely used commercial risk assessment software COMPAS is no more accurate or fair than predictions made by people with little or no criminal justice expertise. In addition, despite COMPAS's collection of 137 features, the same accuracy can be achieved with a simple linear predictor with only two features.

(see Varha 2023 for a comprehensive review)

Bias in ML and limitations

SCIENCE ADVANCES | RESEARCH ARTICLE

RESEARCH METHODS

The accuracy, fairness, and li… of predicting recidivism

Julia Dressel and Hany Farid*

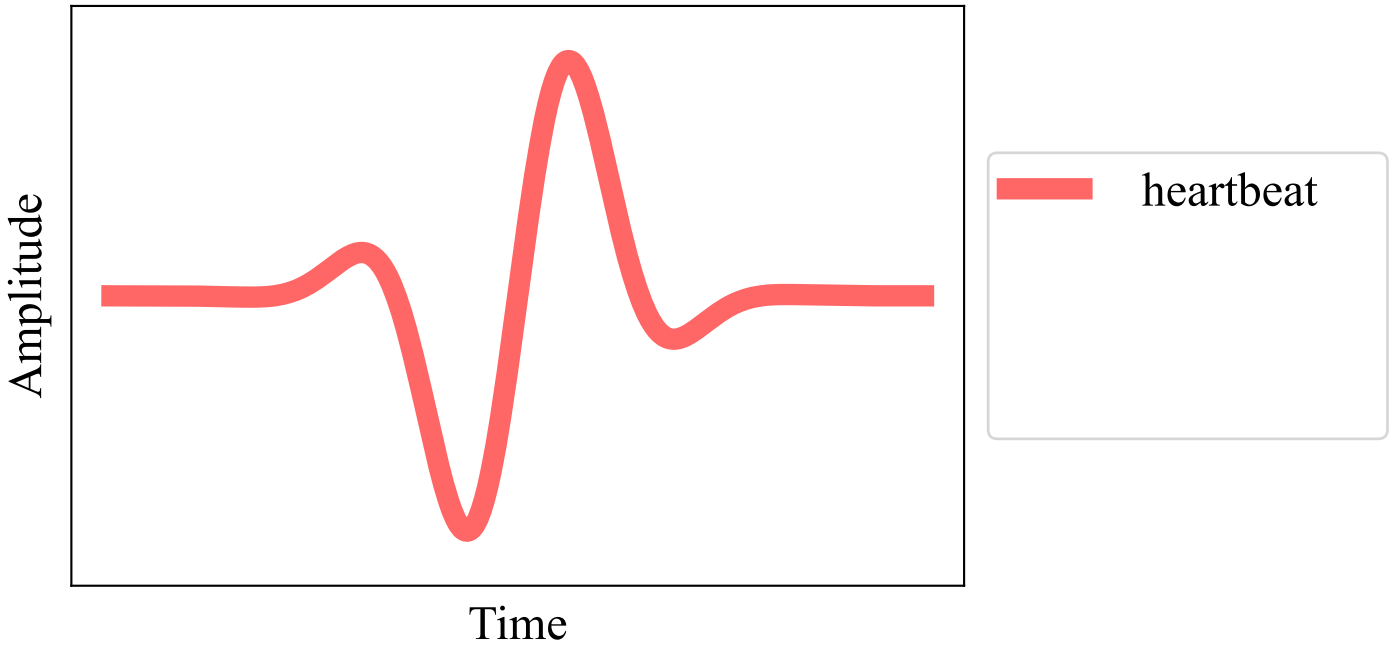
Algorithms for predicting recidivism are commonly used to a… crime. These predictions are used in pretrial, parole, and senter… big data and advanced machine learning make these analyses… however, that the widely used commercial risk assessment sof… tions made by people with little or no criminal justice expe… features, the same accuracy can be achieved with a simple lin…

Table 1. Human versus COMPAS algorithmic predictions from 1000 defendants. Overall accuracy is specified as percent correct, AUC-ROC, and criterion sensitivity (d') and bias (β). See also Fig. 1.

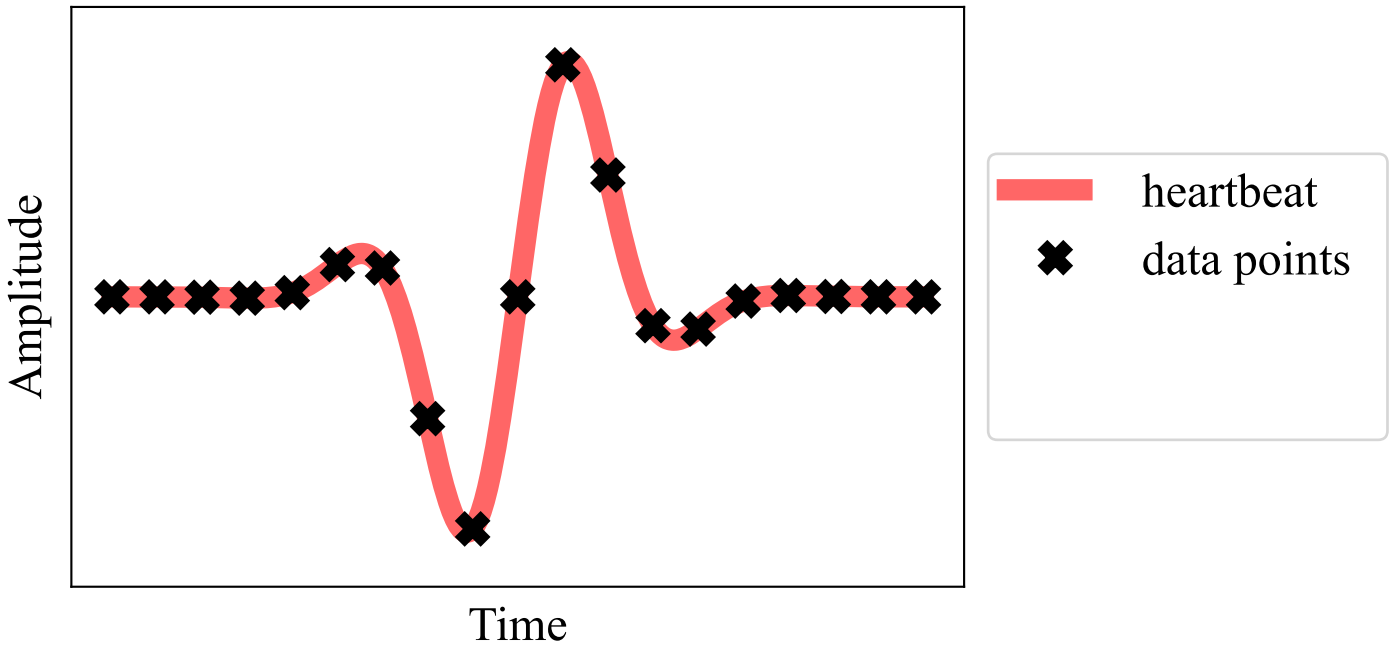
	(A) Human (no race)	(B) Human (race)	(C) COMPAS
Accuracy (overall)	67.0%	66.5%	65.2%
AUC-ROC (overall)	0.71	0.71	0.70
d'/β (overall)	0.86/1.02	0.83/1.03	0.77/1.08
Accuracy (black)	68.2%	66.2%	64.9%
Accuracy (white)	67.6%	67.6%	65.7%
False positive (black)	37.1%	40.0%	40.4%
False positive (white)	27.2%	26.2%	25.4%
False negative (black)	29.2%	30.1%	30.9%
False negative (white)	40.3%	42.1%	47.9%

(see Varha 2023 for a comprehensive review)

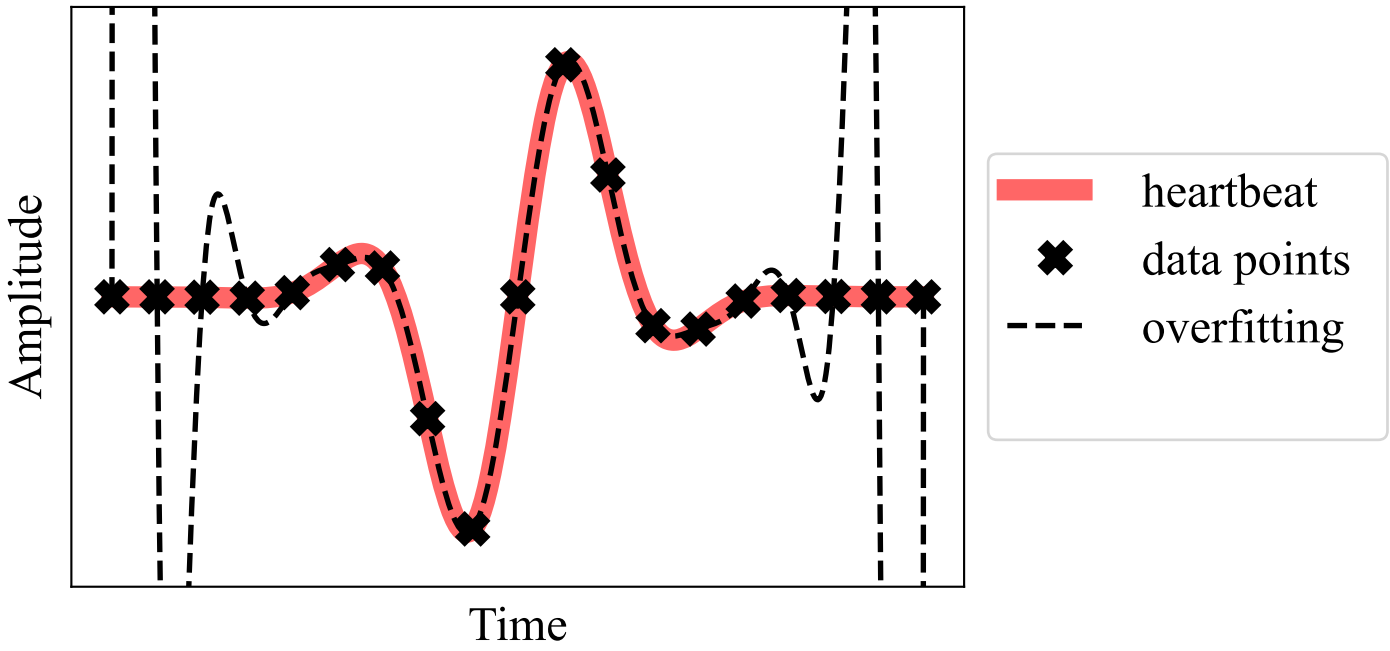
Example: Heartbeat



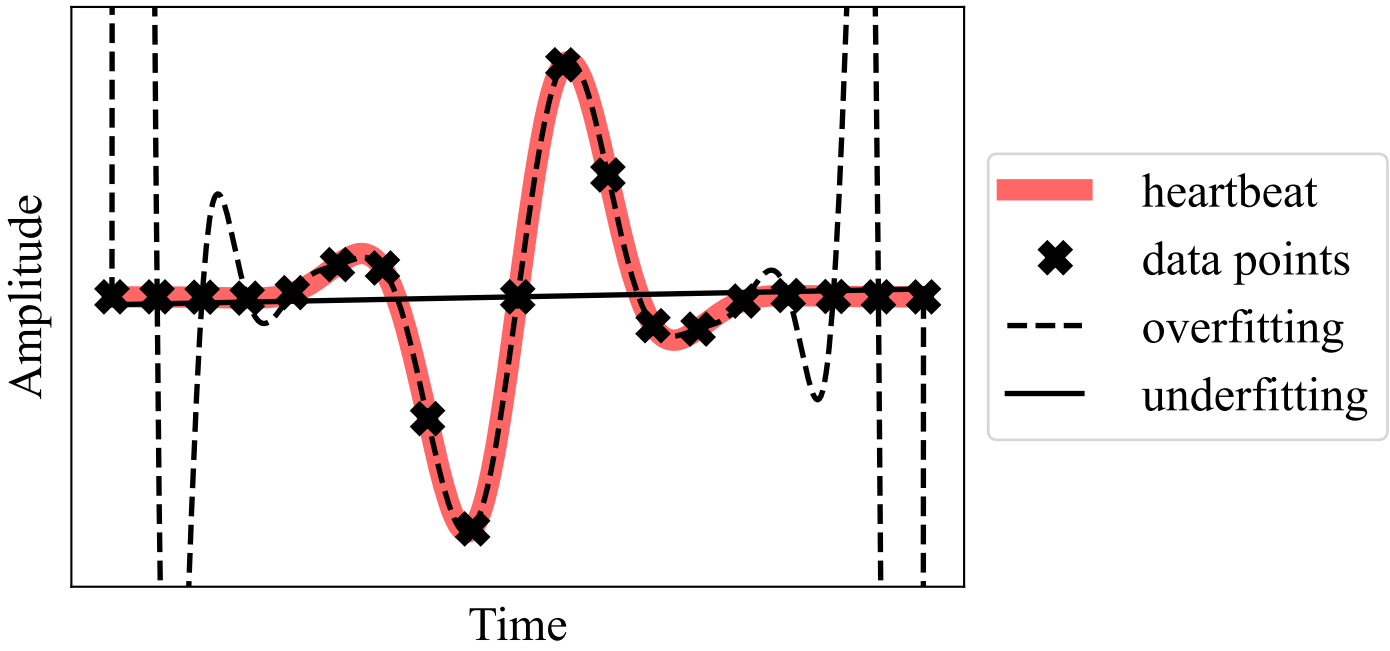
Example: Heartbeat

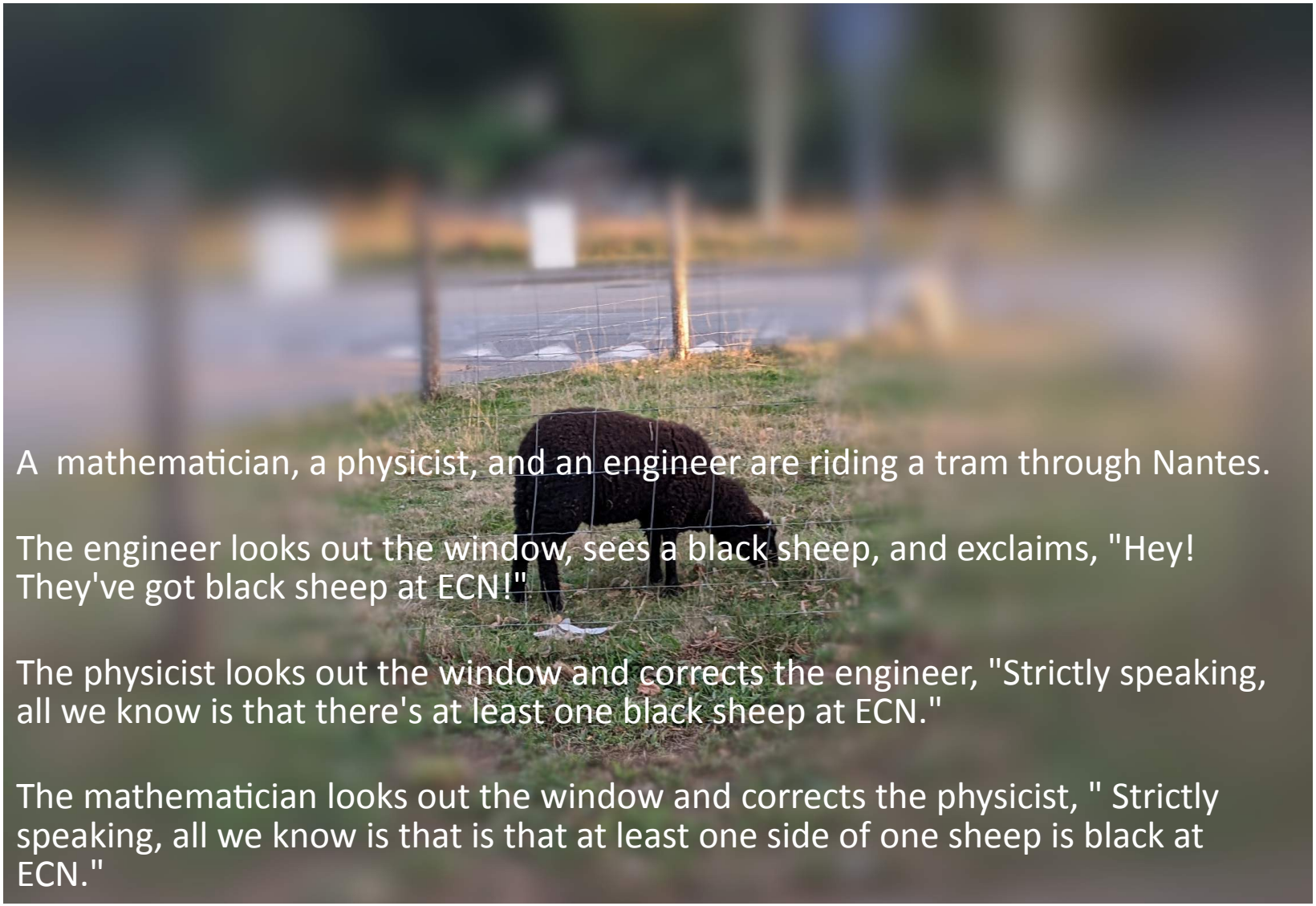


Example: Heartbeat



Example: Heartbeat





A mathematician, a physicist, and an engineer are riding a tram through Nantes.

The engineer looks out the window, sees a black sheep, and exclaims, "Hey! They've got black sheep at ECN!"

The physicist looks out the window and corrects the engineer, "Strictly speaking, all we know is that there's at least one black sheep at ECN."

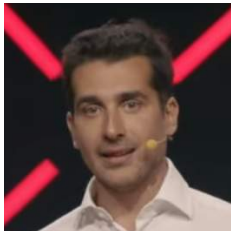
The mathematician looks out the window and corrects the physicist, " Strictly speaking, all we know is that is that at least one side of one sheep is black at ECN."

What's next?

Today

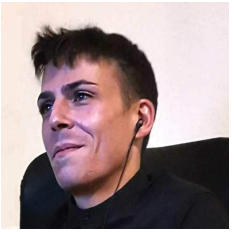
8:30-9:00 **Introduction to the school** (*Pr. Felix Darve*)

9:00-10:00 **Overview of Machine Learning** (*Pr. Ioannis Stefanou*)



10:00-10:30 Coffee Break

10:30-12:00 **Introduction to regression methods** (*Dr Filippo Masi*)



12:00-13:30 Lunch

13:30-15:00 **Hands-on introduction to regression methods** (*Dr Filippo Masi*)

15:00-15:30 Coffee Break

15:30-16:30 **Unsupervised topological learning** (*Pr. Noël Jakse*)

16:30-17:45 **Introduction to classification methods** (*Pr. Noël Jakse*)

17:45-19:00 **"Data-Driven" modeling of geomaterials** (*Dr. Konstantinos Karapiperis*)

Friday 29/9/2023

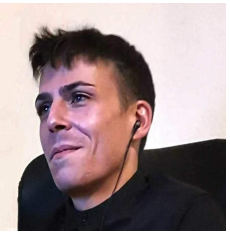
9:00-10:00 **Introduction to Artificial Neural Networks** (*Dr Filippo Gatti*)

10:00-10:30 Coffee Break

10:30-12:30 **Hands-on introduction to Artificial Neural Networks I** (*Dr Filippo Gatti*)

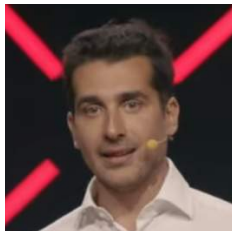
12:30-14:00 Lunch

14:00-15:00 **Hands-on introduction to Artificial Neural Networks II** (*Dr Filippo Gatti*)



15:00-15:30 Coffee Break

15:30-17:30 **Hands-on Physics and Thermodynamics based Artificial Neural Networks** (*Dr Filippo Masi & Pr. Ioannis Stefanou*)



17:30-19:00 **Surprise!**

Saturday 30/9/2023

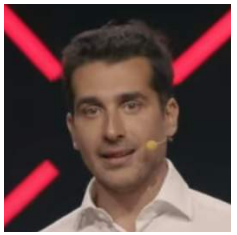
8:30-10:00 **Introduction to Reinforcement Learning**
(D*r Alexandros Stathas, Dr Diego Gutierrez-Oribio & Pr. Ioannis Stefanou*)

10:00-10:30 Coffee Break

10:30-12:00 **Hands-on Introduction to Reinforcement Learning**
(*Dr Diego Gutierrez-Oribio, Dr Alexandros Stathas & Pr. Ioannis Stefanou*)

12:00-12:30 **Closure** (*Pr. Felix Darve & Pr. Ioannis Stefanou*)






Objectives

By the end of this school we expect from you to have:

- demystified and understood what ML is;
- be conscious of the fundamental notions of the most important ML methods;
- used ML in simple examples, got aware of pitfalls and understood the need for physics- and (geo-)mechanics-based ML methods.

Requirements:

- basic knowledge of Python;
- basic concepts in mathematics;
- have some nice problems in mind that could combine ML and geomechanics!
- PARTICIPATE!

Online material

- Updated version of the book
- Codes and hands-on

  download them at: https://github.com/alert-geomaterials/2023-doctoral-school

Enjoy!

https://www.youtube.com/watch?v=NM-zWTU7X-k

The authors acknowledge the support of the European Research Council (ERC) under the European Union Horizon 2020 research and innovation program (Grant agreement ID 757848 CoQuake).