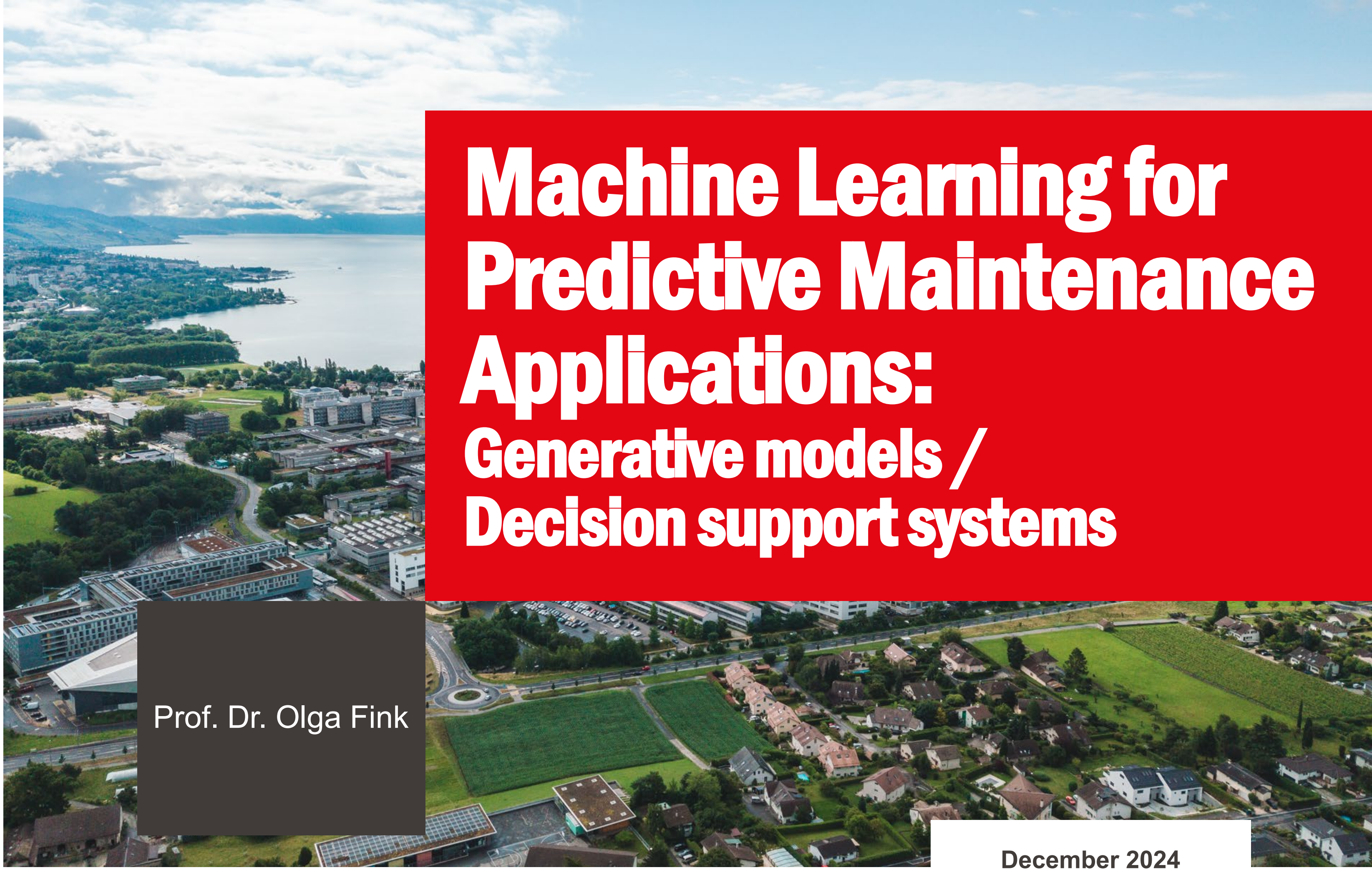


Machine Learning for Predictive Maintenance Applications:

Generative models / Decision support systems

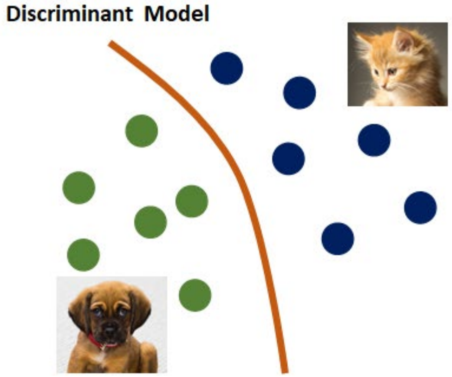
Prof. Dr. Olga Fink

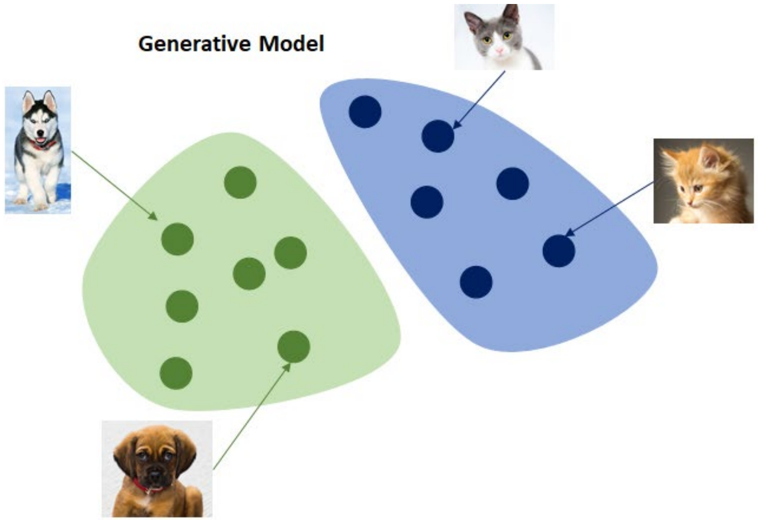
December 2024

École polytechnique fédérale de Lausanne

Discriminative VS Generative

- Discriminative
  - Model P(y|x)
  - Learn the boundary between classes (in classifiers)
  - Usually better performance in low-data regime
- Generative
  - Model P(x,y)
  - Model the distribution of classes
  - Usually needs more data
  - Can be used to generate new samples

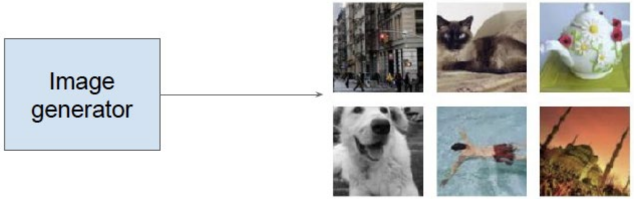




Why generative models?

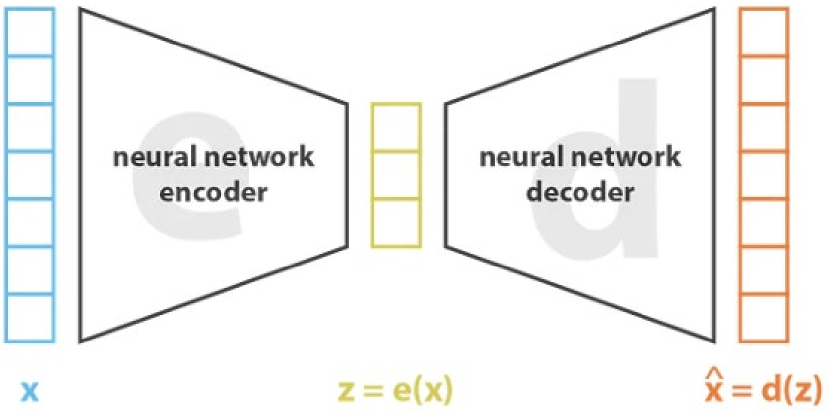
- Model complex and high-dimensional distributions
- Generate realistic synthetic samples
  - Data augmentation
  - Simulation scenarios for learning algorithms
- Fill the blanks in the data
  - Manipulate real samples
- Learn a latent representation useful for other tasks

original enhanced



Variational Autoencoders

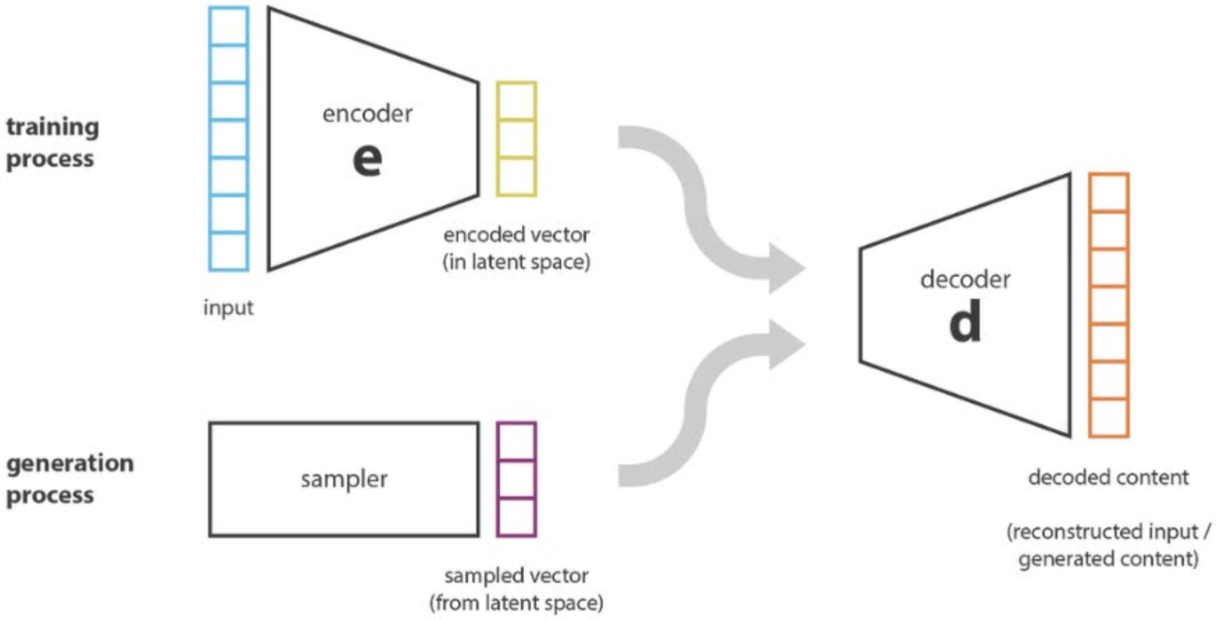
Recap: Autoencoders



$$\text{loss} = \| x - \hat{x} \|^2 = \| x - d(z) \|^2 = \| x - d(e(x)) \|^2$$

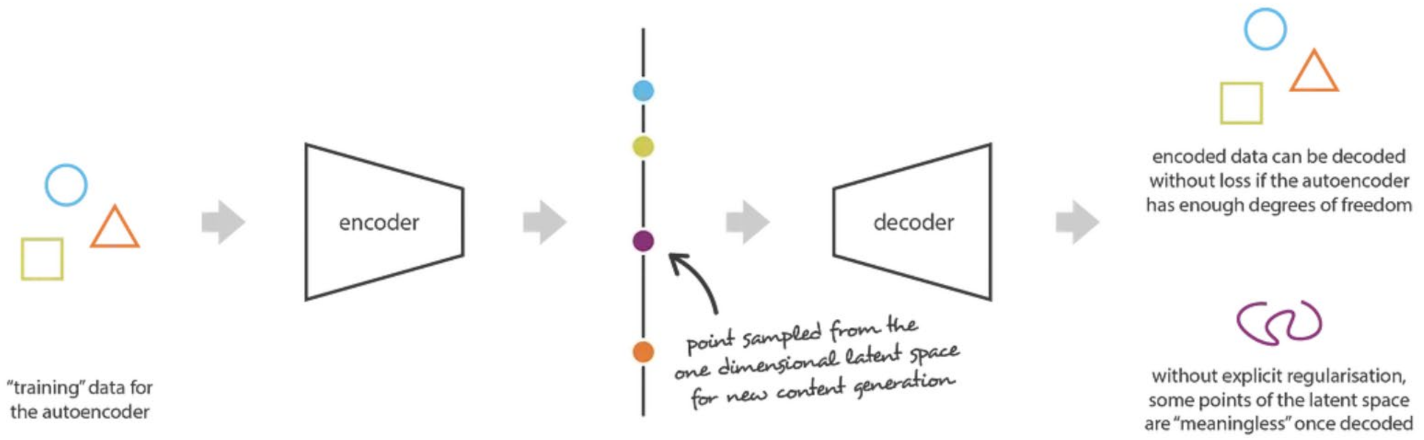
Source: https://towardsdatascience.com

Potential generation process



Source: https://towardsdatascience.com

AE for data generation

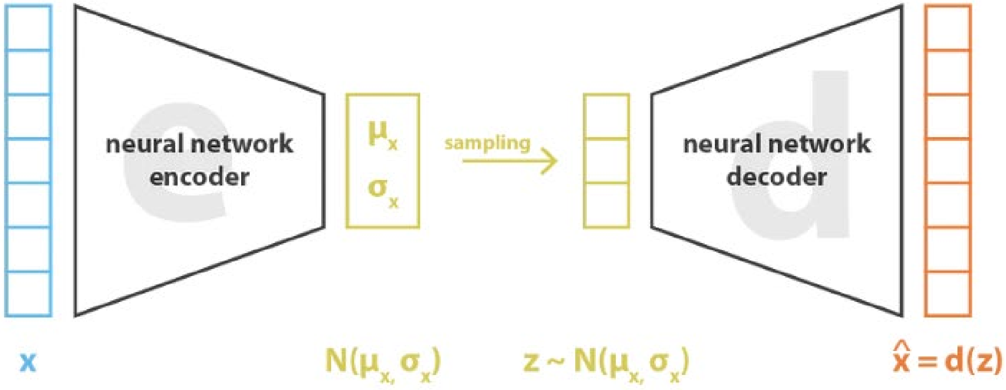


Source: https://towardsdatascience.com

Variational Autoencoders

- Definition: Variational Autoencoders are a type of generative model that use machine learning to produce new data points that are statistically similar to a given dataset.
- Key Concept: VAEs are based on the principles of probability and statistics, aiming to model the underlying data distribution.
- Purpose of VAEs
  - Data Generation: Generate new data instances that mimic the original data.
  - Dimensionality Reduction: Compress data into a more manageable latent space.

General structure of VAE



$$\text{loss} = \| x - \hat{x} \|^2 + \text{KL}[\, N(\mu_x, \sigma_x), N(0, I)\,] = \| x - d(z) \|^2 + \text{KL}[\, N(\mu_x, \sigma_x), N(0, I)\,]$$

Source: https://towardsdatascience.com

Objective function of VAE

$$\mathcal{L}_{\text{VAE}} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}))$$

Reconstruction loss

KL Divergence Loss

Simple AE vs. VAE

simple autoencoders

input x $\xrightarrow{\text{encoding}}$ latent representation $z = e(x)$ $\xrightarrow{\text{decoding}}$ input reconstruction $d(z)$

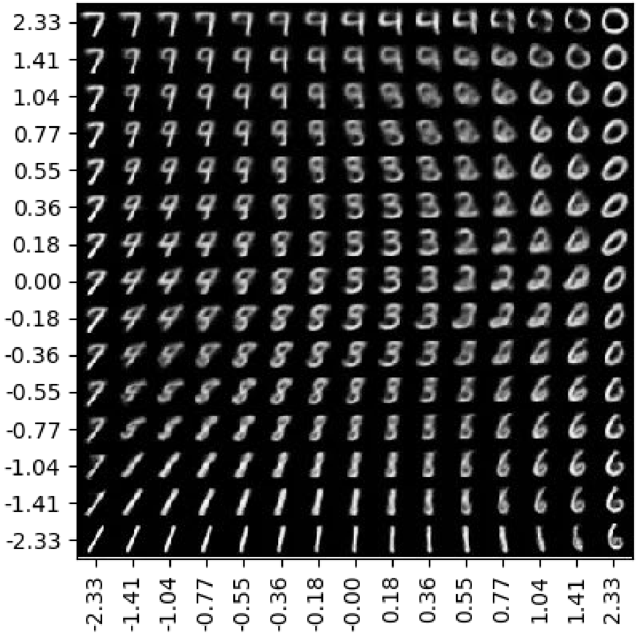
---

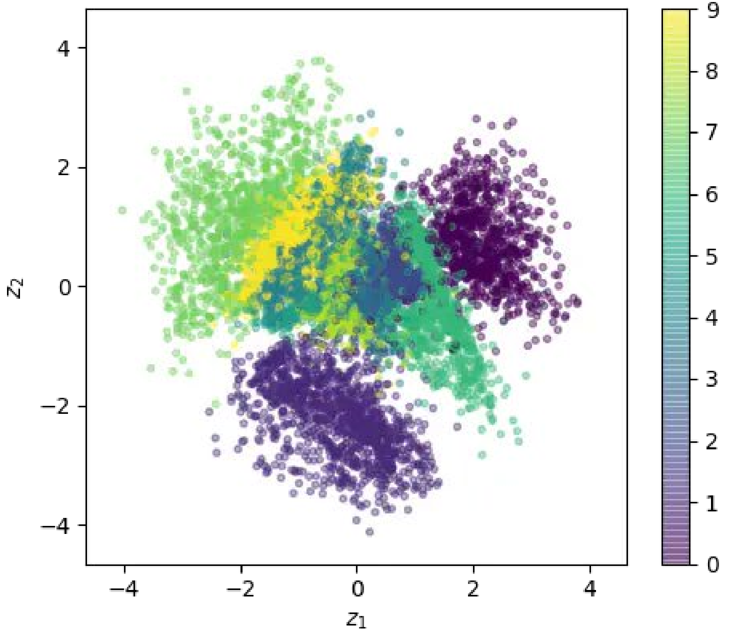
variational autoencoders

input x $\xrightarrow{\text{encoding}}$ latent distribution $p(z|x)$ $\xrightarrow{\text{sampling}}$ sampled latent representation $z \sim p(z|x)$ $\xrightarrow{\text{decoding}}$ input reconstruction $d(z)$

Source: https://towardsdatascience.com

VAE on MNIST

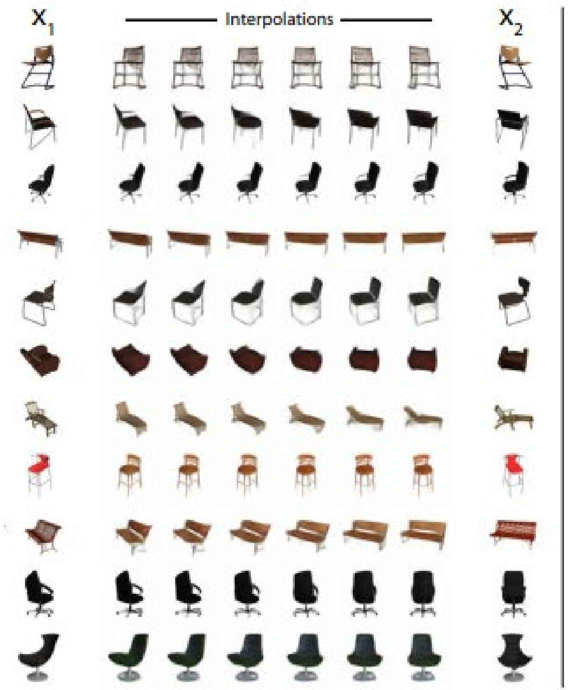

2.33
1.41
1.04
0.77
0.55
0.36
0.18
0.00
-0.18
-0.36
-0.55
-0.77
-1.04
-1.41
-2.33
-2.33 -1.41 -1.04 -0.77 -0.55 -0.36 -0.18 -0.00 0.18 0.36 0.55 0.77 1.04 1.41 2.33

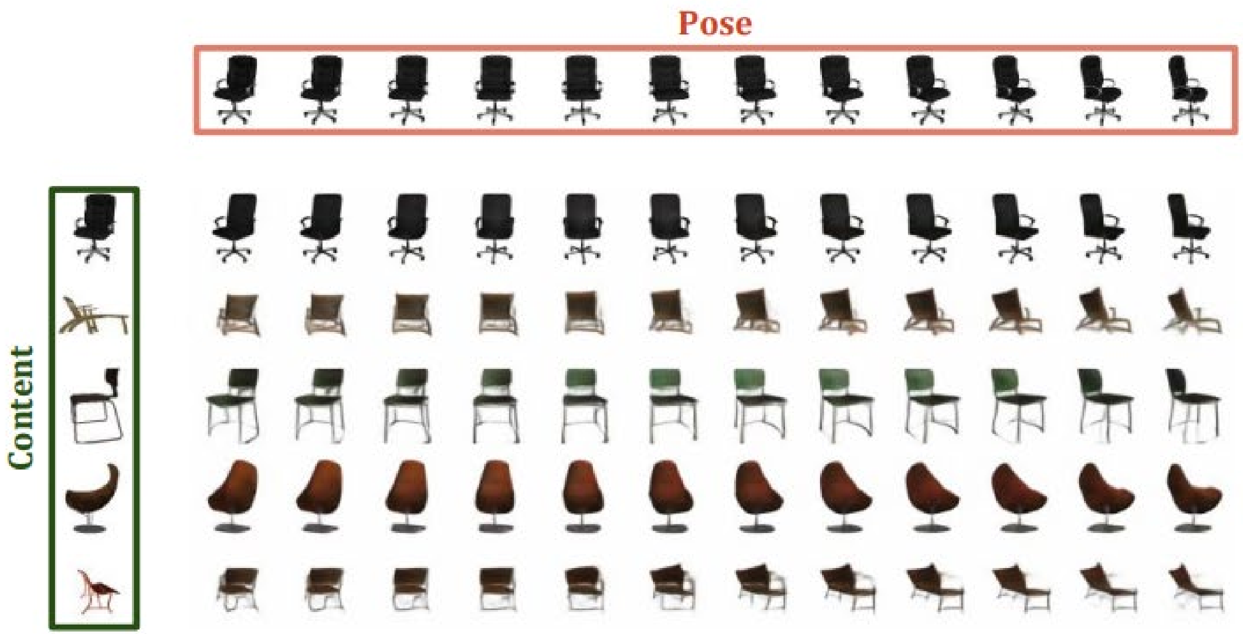

4
2
0
−2
−4
z_2
−4 −2 0 2 4
z_1
9
8
7
6
5
4
3
2
1
0

Idea of dientanglement

- The concept of disentanglement is based on the hypothesis that real-world data is generated by a few independent explanatory factors of variation
- Can be sued for controlled data generation:
  - learn a disentangled feature representation of the data
  - use these disentangled features representing independent factors of variation to generate data samples with desired characteristics in controlled ways

Disentangled features: generation





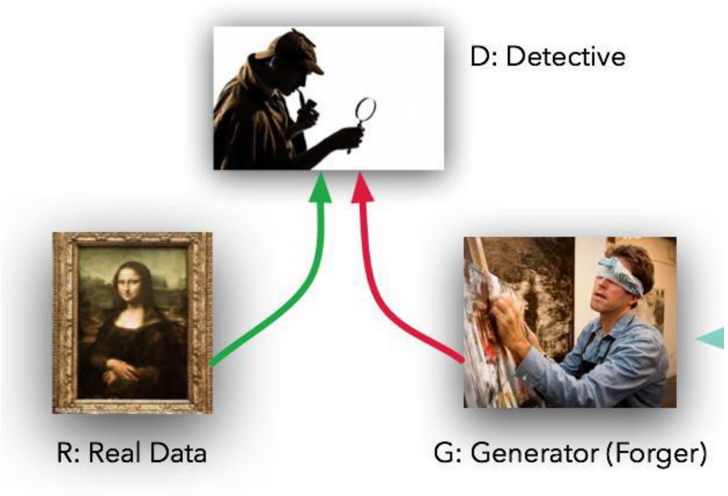
Denton, Emily L. "Unsupervised learning of disentangled representations from video." *Advances in neural information processing systems* 30 (2017).

Generative Adversarial Networks (GANs)

GANs

- 2014: a major milestone in generative models
- GAN: Generative Adversarial Network
- Take a look at http://thispersondoesnotexist.com by Style-GAN

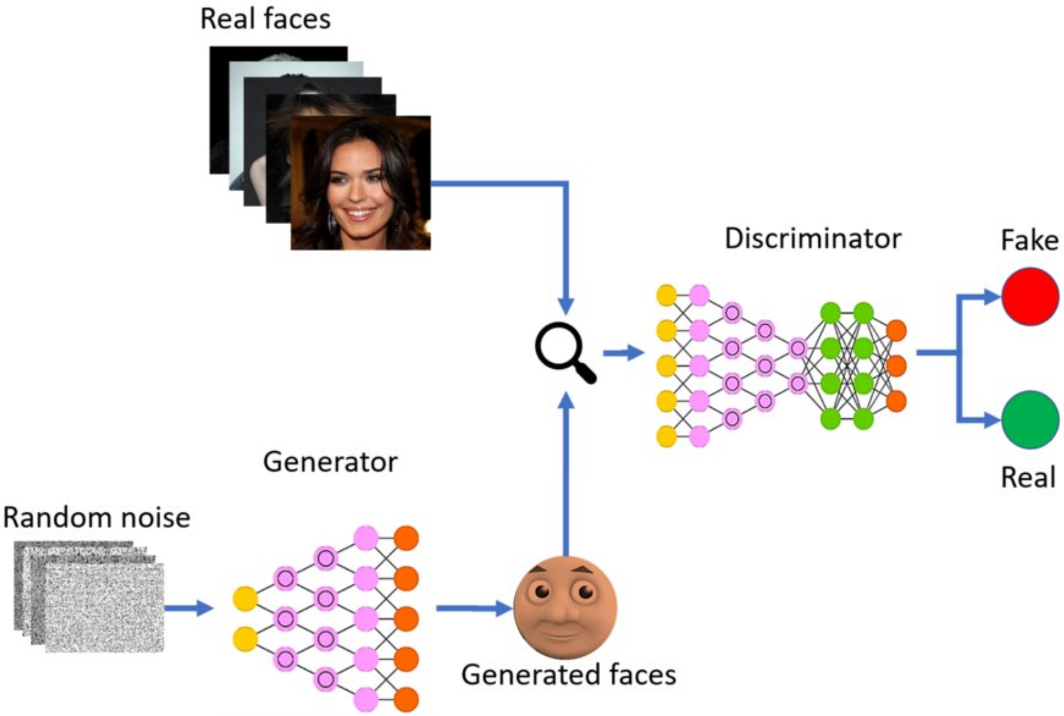
Adversarial Training



Generative Adversarial Networks (GAN)

Figure 1 https://github.com/devnag/pytorch-generative-adversarial-networks

GANs



Source: https://medium.com/sigmoid/a-brief-introduction-to-gans-and-how-to-code-them-2620ee465c30

GAN loss functions

Min-Max Loss:

$$E_x[log(D(x))] + E_z[log(1 - D(G(z)))]$$

Discriminator Loss:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right]$$

Generator Loss:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)$$

Example

Different DA setups

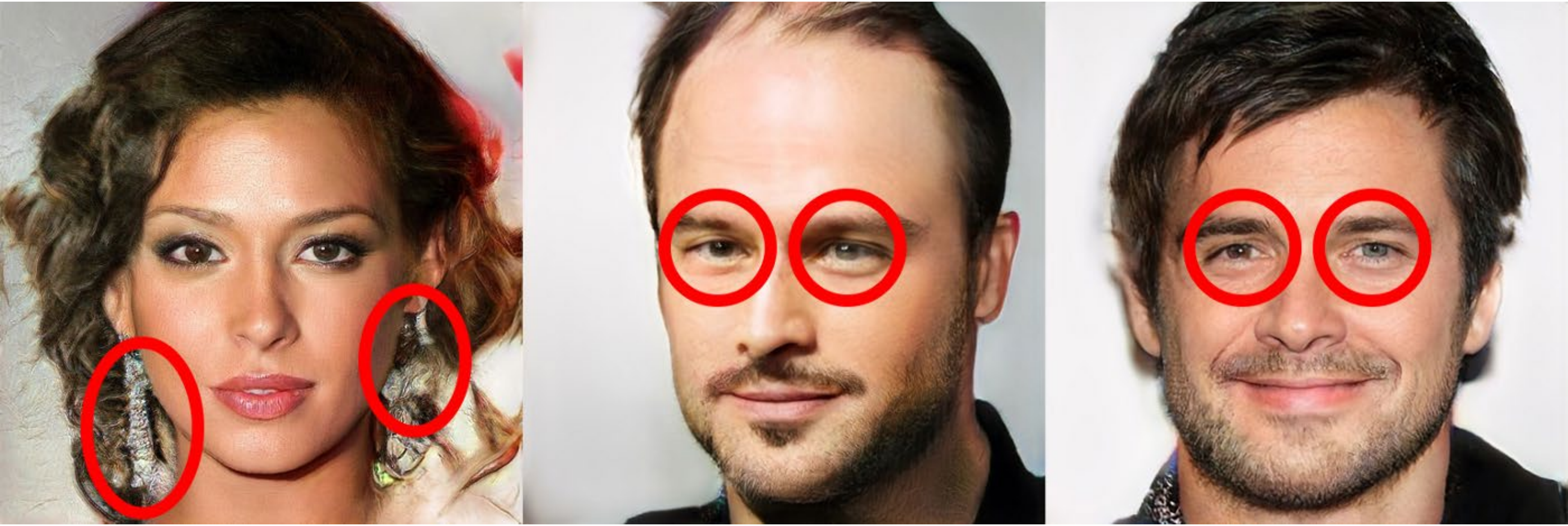
Partial DA under «extreme» setup

Source - Train	Target - Train	Target - Test
Healthy	Healthy	Healthy
Fault 1		Fault 1
Fault 2		Fault 2
Fault 3		Fault 3
Fault x		Fault x

Open-Partial DA for Fault Diagnosis

Source - Train	Target - Train	Target - Test	Source - Test
Healthy	Healthy	Healthy	Healthy
Fault 1	---	Fault 1	Fault 1
Fault 2	---	Fault 2	Fault 2
----	Fault 3	Fault 3	Fault 3
----	Fault 4	Fault 4	Fault 4

GAN Generated images



Source: https://kcimc.medium.com/

Challenges / requirements for synthetic faults

- Need to be physically plausible / interpretable
- Need to be specific to the considered system and specific to the operating conditions
- Usually, samples of all fault types in one system cannot be assumed!

EPFL

Controlled Generation of Unseen Faults

Generate unseen Faults in a new domain (new unit or new operating condition) and “exchange fault patterns between different domains”

→ Open-Partial Domain adaptation

General Idea

- Hypothesis: faulty signal can be represented by combining
  - domain-specific variations of the normal steady state operation (represented by healthy data)
  - and a domain-independent signal representing solely the characteristic from a fault.
- Hypothesis in the Fourier domain:
  - Fourier spectrum of fault data can be expressed as the sum of

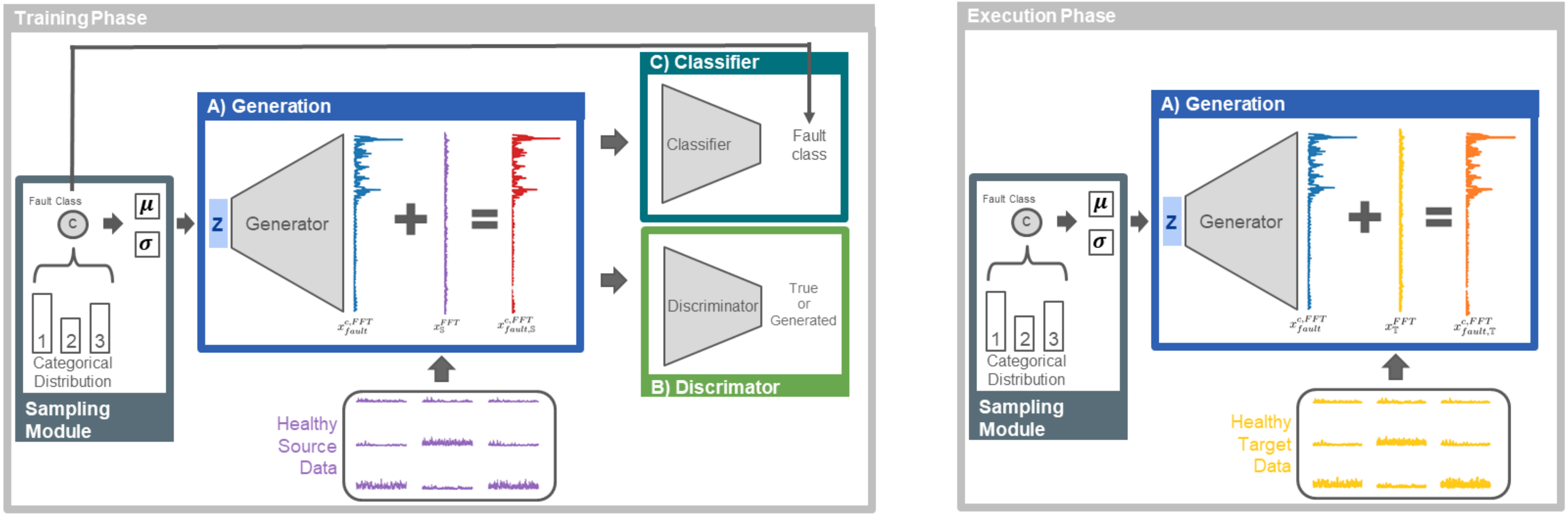
  1) the spectrum of a signal from normal operation and

  2) the spectrum of a signal representing the domain-independent faulty condition.

$$x_{fault,\mathbb{X}}^{c,FFT} = x_{\mathbb{X}}^{FFT} + w * x_{fault}^{c,FFT}$$

Rombach, K., G. Michau & O. Fink: Controlled Generation of Unseen Faults for Partial and Open-Partial Domain Adaptation, Reliability Engineering and System Safety

FaultSignatureGAN



Training Phase:

→ Training the A) generative model to generate domain independent fault characteristics while imposing B) plausibility with the discriminator in the source domain and C) semantic consistency with the classifier.

Execution Phase

→ generation of unseen target data

Rombach, K., G. Michau & O. Fink: Controlled Generation of Unseen Faults for Partial and Open-Partial Domain Adaptation, Reliability Engineering and System Safety

Dataset: Paderborn Bearing

Domain	Rotational speed [rpm]	Load Torque [Nm]	Radial Force [N]
0	1500	0.7	1000
1	900	0.7	1000
2	1500	0.1	1000
3	1500	0.7	400

- Six Classes
  One healthy class, two OR fault severities and three IR fault severities
- Four Domains
- Generate faulty data in different domains
- Domain 1: differs significantly from the other conditions

Open-Partial Setup for the Paderborn Dataset

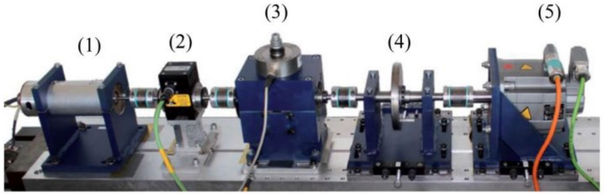
Real datasets

Real and generated training datasets

Real test datasets

Rombach, K., G. Michau & **O. Fink**: Controlled Generation of Unseen Faults for Partial and Open-Partial Domain Adaptation, Reliability Engineering and System Safety

Transfer between OCs knowing the health + 1 OC in each of the datasets (without OC1)



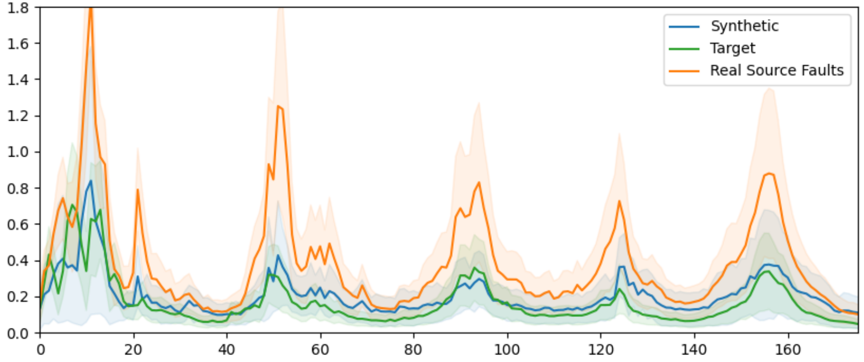
Paderborn Dataset

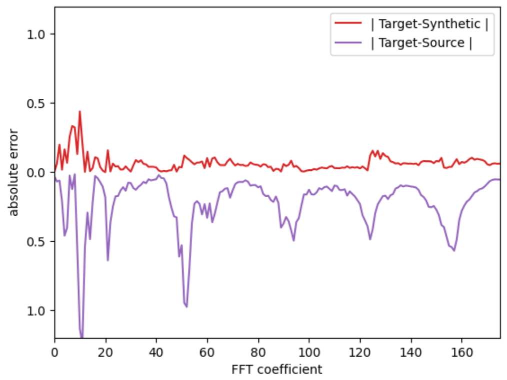
	Baseline	FaultSignature GAN
Mean Accuracy (over all test datasets and transfer directions)	83.4%	**96.0%**

*The test datasets comprise the real missing fault data as well as of a 30% of known health conditions

Rombach, K., G. Michau & **O. Fink**: Controlled Generation of Unseen Faults for Partial and Open-Partial Domain Adaptation, Reliability Engineering and System Safety

Paderborn data visualization of the OR severity 1 fault comparing real fault data with generated fault data



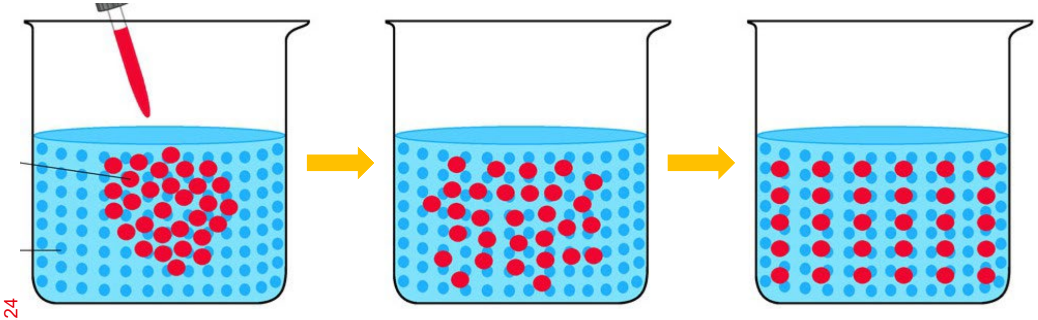


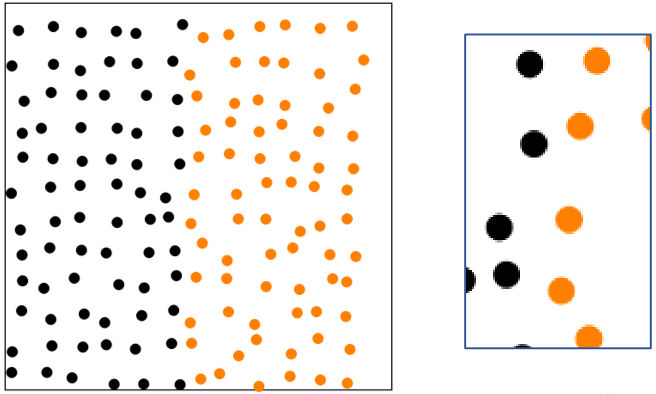
Rombach, K., G. Michau & **O. Fink**: Controlled Generation of Unseen Faults for Partial and Open-Partial Domain Adaptation, Reliability Engineering and System Safety

Diffusion models

Diffusion Process

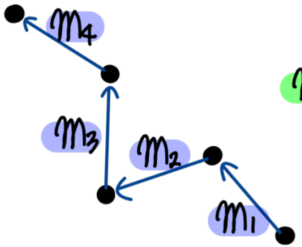
- Diffusion models are inspired by non-equilibrium thermodynamics.
- For a small fraction of the time, it is difficult to determine whether particles are moving in the direction of mixing or in the opposite direction.





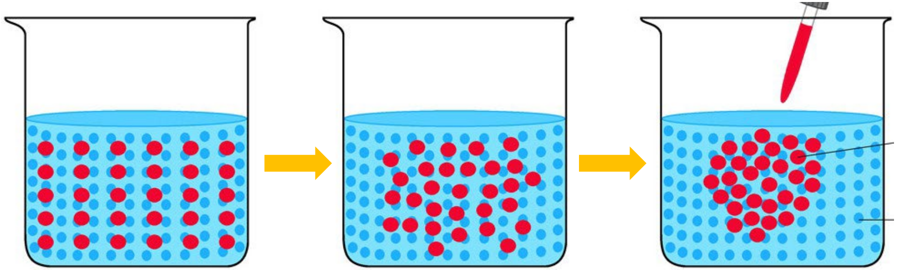
Source: Jumin Lee

Diffusion Process



$m_t \sim N(\mu, \sigma)$

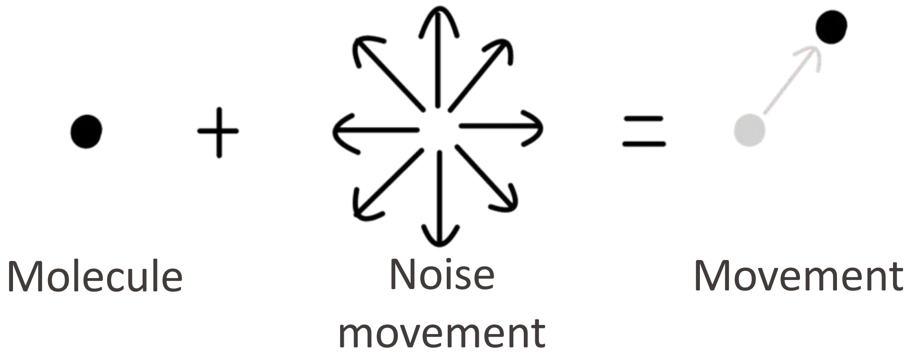
- If we look at the movement of a single molecule on a very short time scale, it follows a Gaussian distribution.
- Since the direction of mixing and the opposite direction are the same in a very short time, the opposite direction also follows a Gaussian distribution.

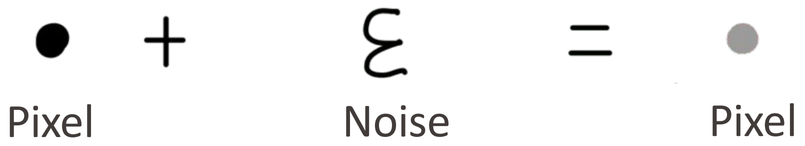


Source: Jumin Lee

Diffusion Process

- Just as we viewed the molecule's motion as a Gaussian-distributed noise, we add a Gaussian-distributed noise to the pixel.



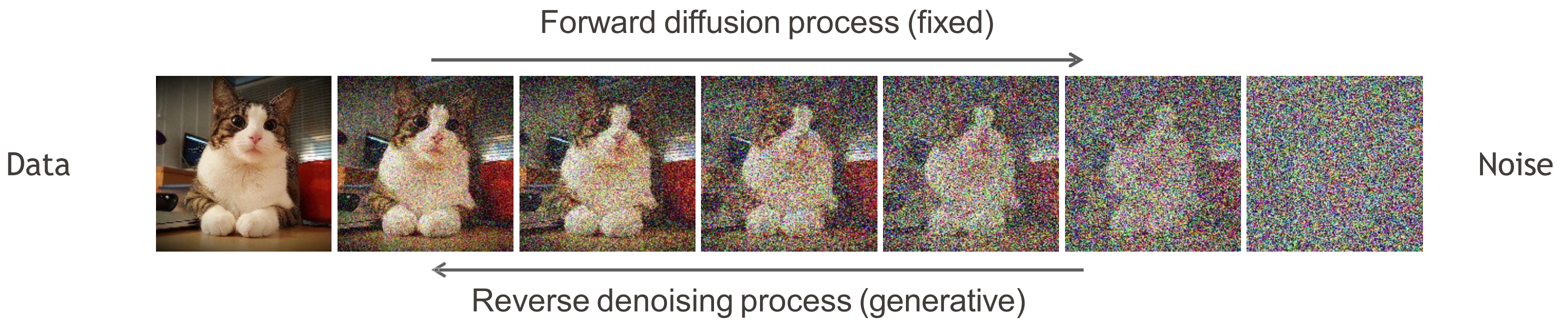


Source: Jumin Lee

Denoising Diffusion Models

Denoising diffusion models consist of two processes:

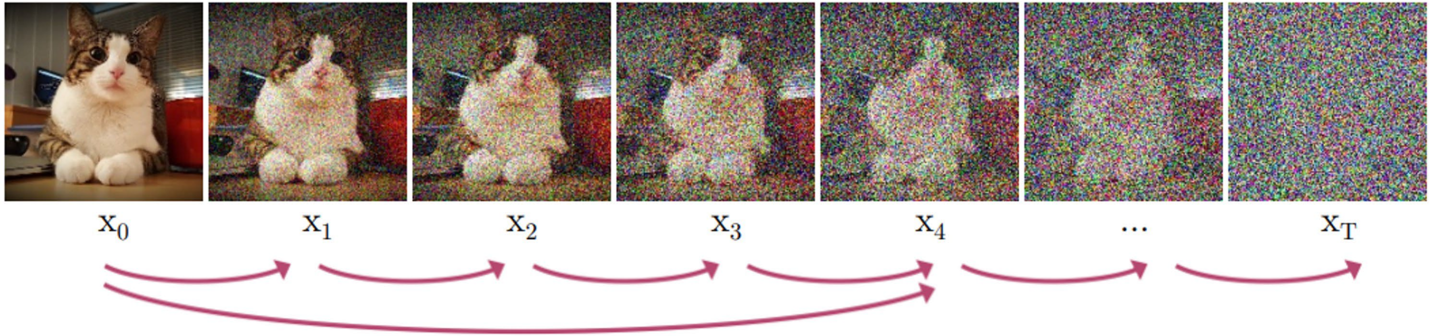
- Forward diffusion process that gradually adds noise to input
- Reverse denoising process that learns to generate data by denoising



Source: Jumin Lee

Forward Diffusion Process

The formal definition of the forward process in T steps:



Markov Property

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I})$$ ← Diffusion Kernel

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\ \mathbf{x}_0 + \sqrt{(1-\bar{\alpha}_t)}\ \epsilon \qquad \text{where} \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

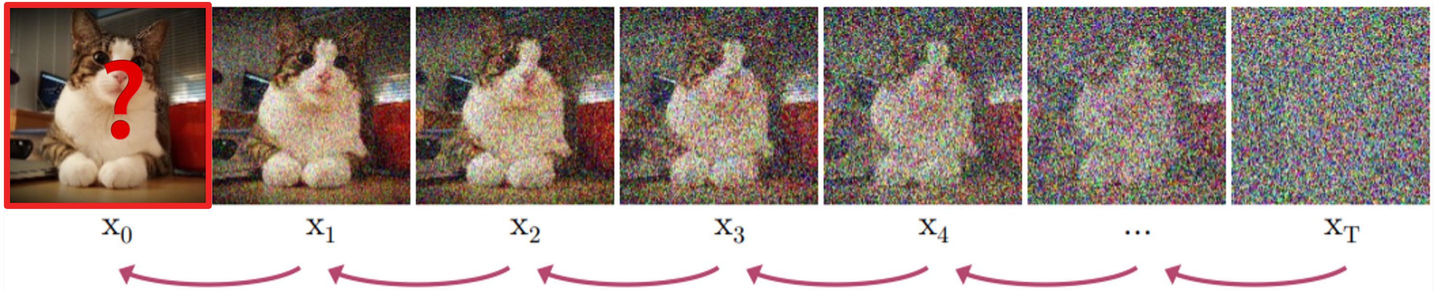
$$\alpha_t := 1 - \beta_t \text{ and } \bar{\alpha}_t := \prod_{s=0}^{t} \alpha_s$$

Source: Jumin Lee

EPFL

Reverse Denoising Process

Formal definition of forward and reverse processes in T steps:



$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I})$$

Model

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \boxed{\mu_\theta(x_t, t)}, \Sigma_\theta(x_t, t))$$

Source: Jumin Lee

Decision support

PHM Process

DataAcquisition (DA)

↓

Data Manipulation(DM)

↓

State Detection(SD)

↓

Health Assessment(HA)

↓

Prognostics Assessment(PA)

↓

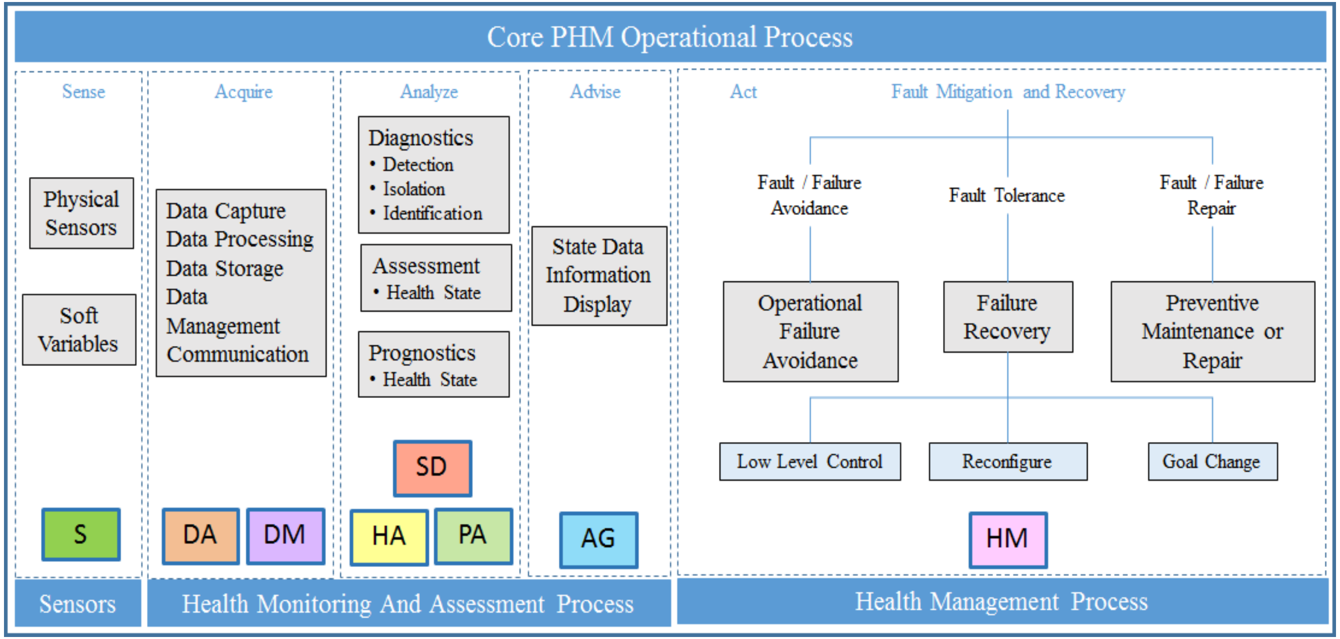
Advisory Generation (AG)

What we focused on up to now

- Up to now:
  - Generate an alarm in case of an anomaly
  - Provide information which fault type has occurred (or at least which signals have shown the largest deviation from normal behaviour)
  - Provide a prediction of the remaining useful life

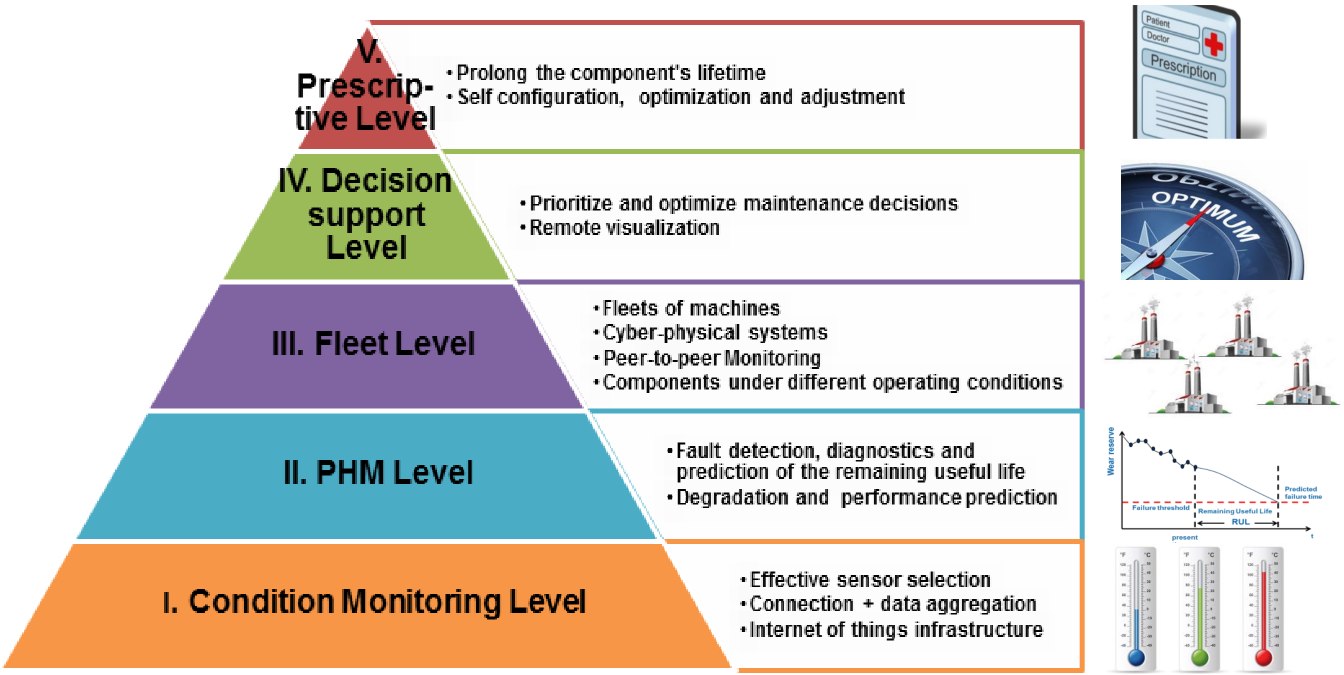
  → Overcoming challenges of the lack of labels, diversity of operating conditions, uncertain measurements…
- Predictions / detections at component / (sub-)system level

Reminder: PHM system operational view



Source: P1856™/D31 Standard

Five levels of condition-based and predictive maintenance



Why decision support systems for PHM required?

- Required recommendation at system, fleet and enterprise level:
  - Optimal specific action (what needs to be done)
  - Optimal point in time
  - Required ressource usage (including personel, tools and material)
  - Under the given constraints from the ressource availability and operational requirements

(Prognostics and) Health Management

- Health management utilizes prognostic information to make **decisions** related to safety, condition-based maintenance, ensuring adequate inventory, and product life extension.
- Health management goes beyond the predictions of failure times

→supports optimal maintenance and logistics decisions

→by considering the available resources +

→the operating context +

→the economic consequences of different faults.

- Health management process of taking timely and optimal maintenance actions based on outputs from diagnostics and prognostics, available resources and operational demand

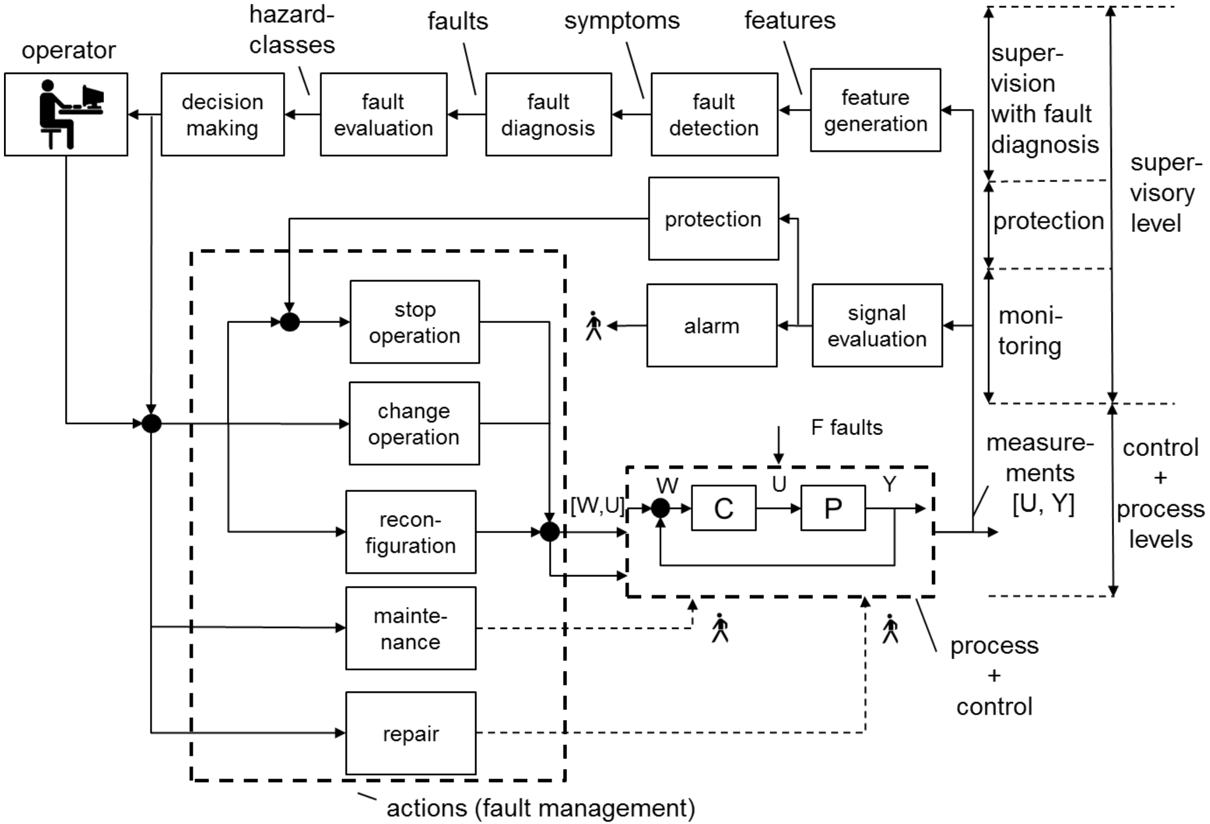
Types of decisions

- Aggregate information at the system level (taking boundary conditions into consideration)
- Health-aware control → operation
- Adjustment of operations with respect to the equipment’s health state
- Optimization of maintenance scheduling
- Take decision at fleet level (e.g. mission scheduling)
- …

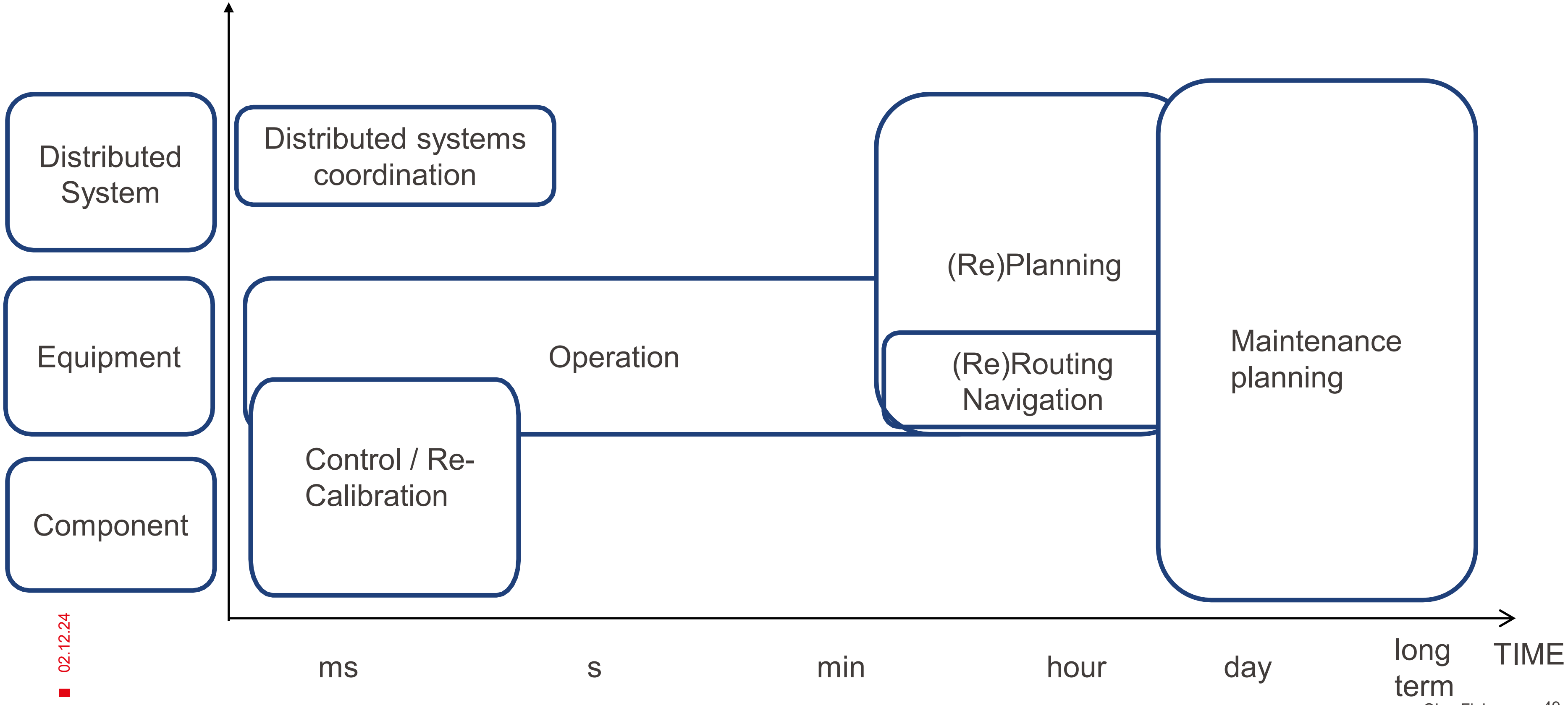
Decision-Making Tools

- **Optimization Algorithms:** Determining the most cost-effective maintenance schedules and resource allocations.
- **Scenario Analysis:** Evaluating the impact of different maintenance strategies or operational changes.
- **Risk Assessment:** Assessing the likelihood and consequences of potential failures to prioritize actions.

Health management



Decision making in complex systems



Steps to perform

- Establish objectives (e.g. availability, reliability, safety, performance and energy consumption), constraints (e.g. ressources), decision variables
- Incorporate operational and maintenance requirements on single system and on the fleet level (plus possible flexibility)
- Establish priorities and decision variables
- Quantify critical metrics
- Perform trade-off study incl. risk

Elements of decision support systems

Fleets of different assets (being composed of different components) incl. their criticality

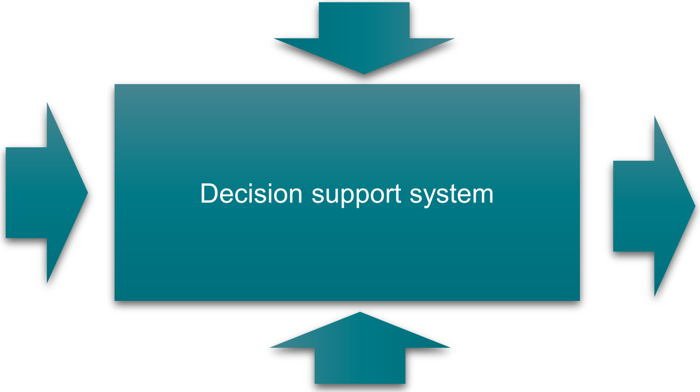
Health indicators or RULs, detection alarms (ideally including the uncertainty level)

Scheduled maintenance + inspections (e.g. due to safety requirements)

Maintenance ressources (maintenance infrastructure, tools, spare parts, logistics)

Human ressources (incl. qualification)

Missions / operational requirements



Performance objectives (availability, cost targets, efficiency …)

Operational schedules

Maintenance planning + schedule (grouping of several components / systems)

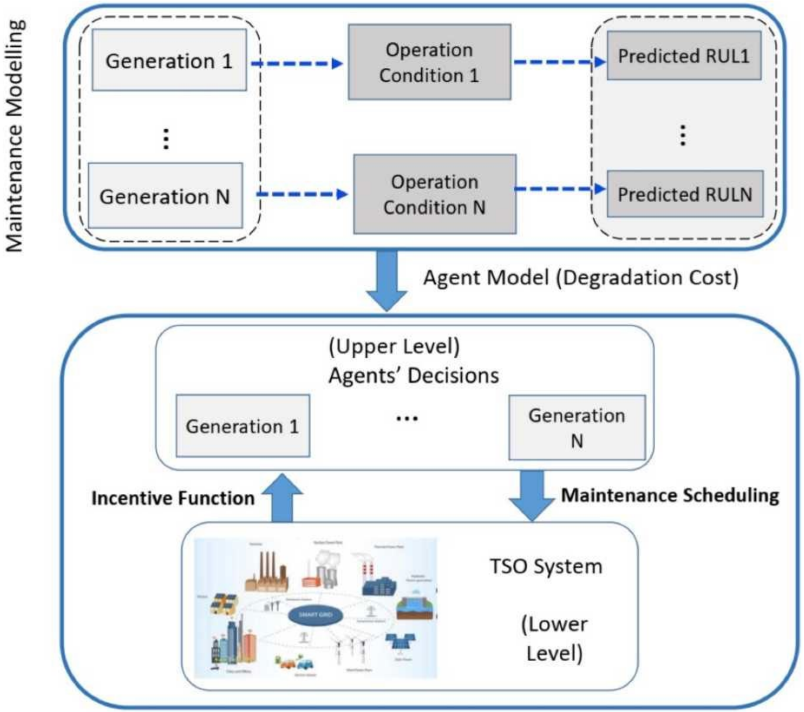
Personnel planning

Requirements for spare parts /logistics

Occupation of maintenance infrastructure

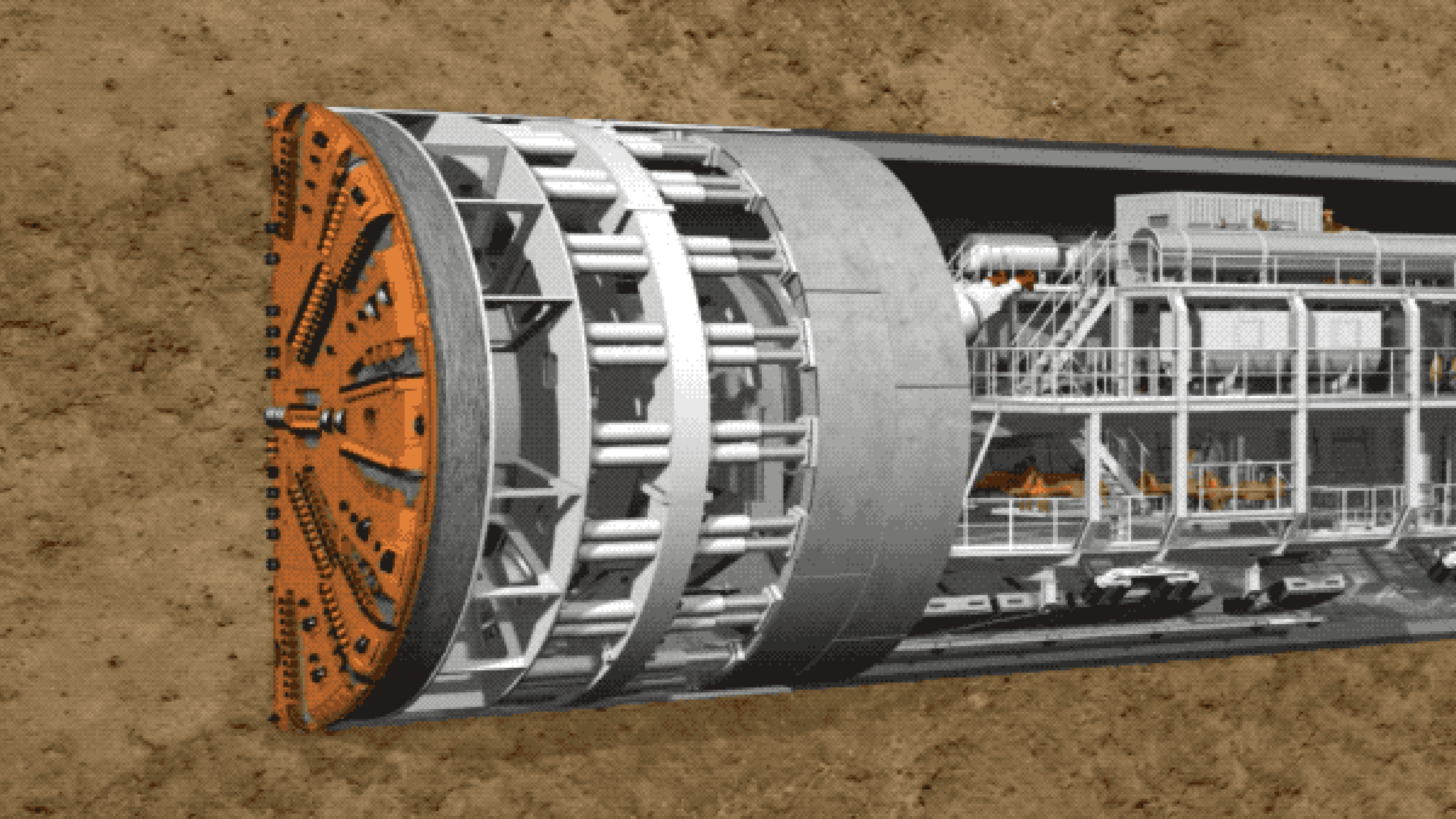
Performance indicators (costs, operational availability, efficiency, product quality, quality of service, etc.)

Multi-agent predictive maintenance scheduling in an electricity market



Rokhforoz, P., Gjorgiev, B., Sansavini, G., & Fink, O. (2021). Multi-agent maintenance scheduling based on the coordination between central operator and decentralized producers in an electricity market. *Reliability Engineering & System Safety*, *210*, 107495.

Decision support for optimal operation



Consequences of uncertainty?

High uncertainty + high complexity → often cost and time overruns (+safety critical risks)

Importance of experienced and skilled operators

Drilling efficiency differs between different operators and often depends on the level of expertise and experience → difficult to train novices

Tasks with incomplete observations / high uncertainty

How can we leverage the the experience and expertise of domain experts?

Under the condition that we don’t know how good the decisions / actions are

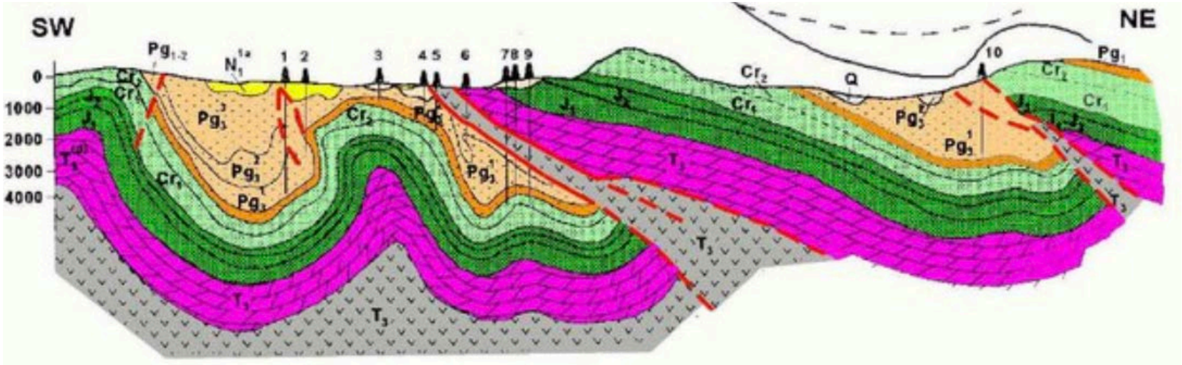
By only observing the data that their decisions generate

Mechanized tunneling: Tunnel construction using Tunnel Boring Machines (TBMs)

- Massive machines employed to build critical infrastructure for modern life
- Large number of sensors installed on a TBM (torque, rotational speed, thrust, ...)
- Geology affects sensor values

Image credit: https://www.theb1m.com/video/first-tbm-for-grand-paris-express-launched

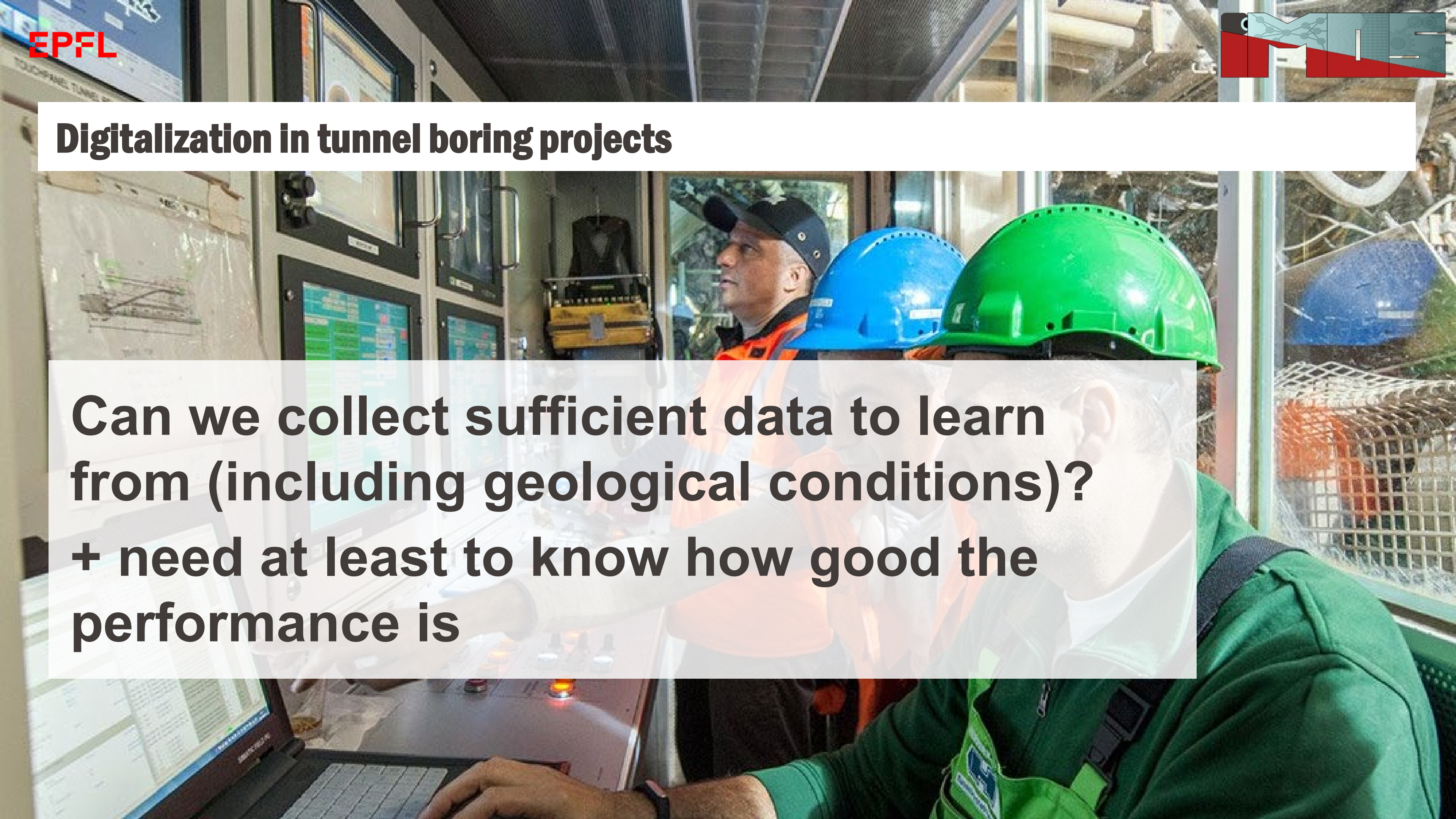
Geological conditions



Source: Bushati S. et al. , «Gophysical outlook on structure of the Albanides»

EPFL

Digitalization in tunnel boring projects



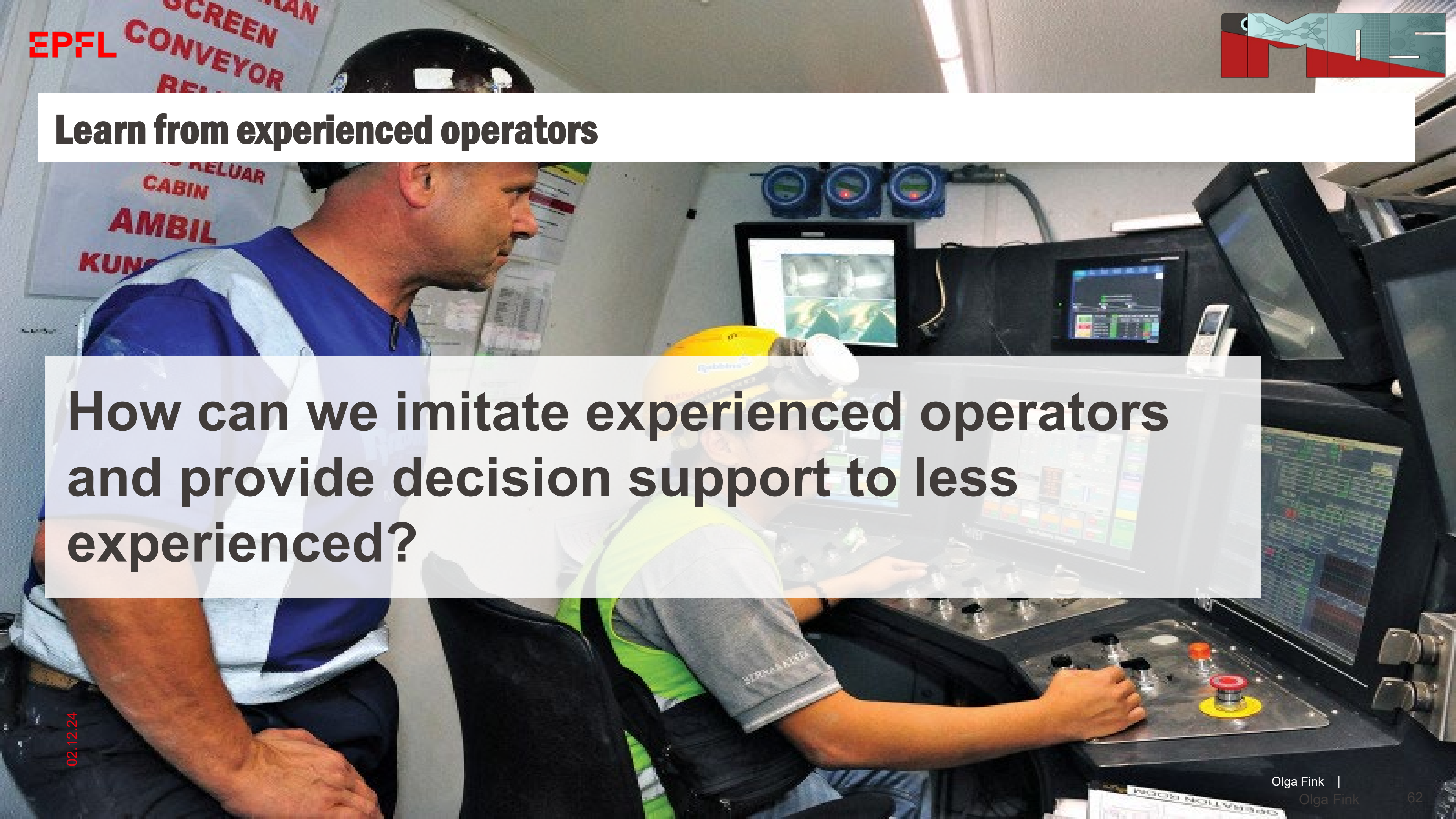
Can we collect sufficient data to learn from (including geological conditions)?

+ need at least to know how good the performance is

Humans – AI - Humans

Learning from observing the decision of domain experts → support novices

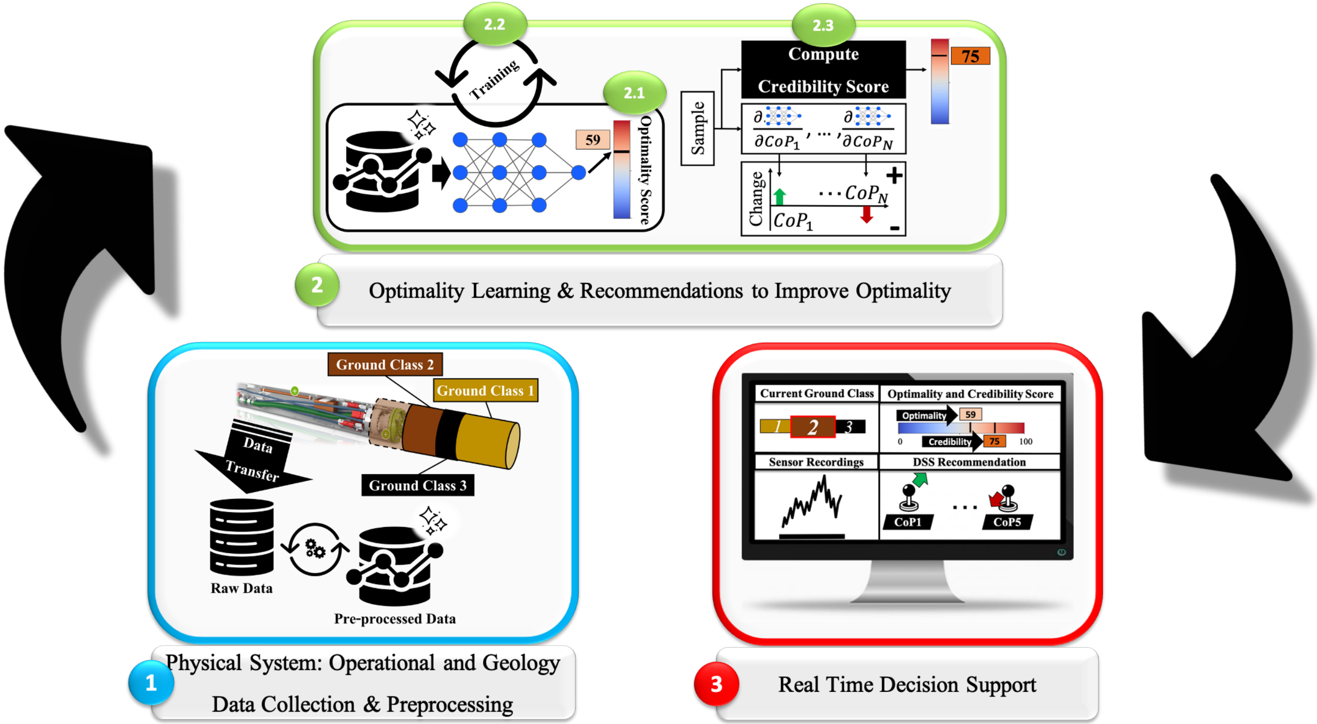
Learn from experienced operators



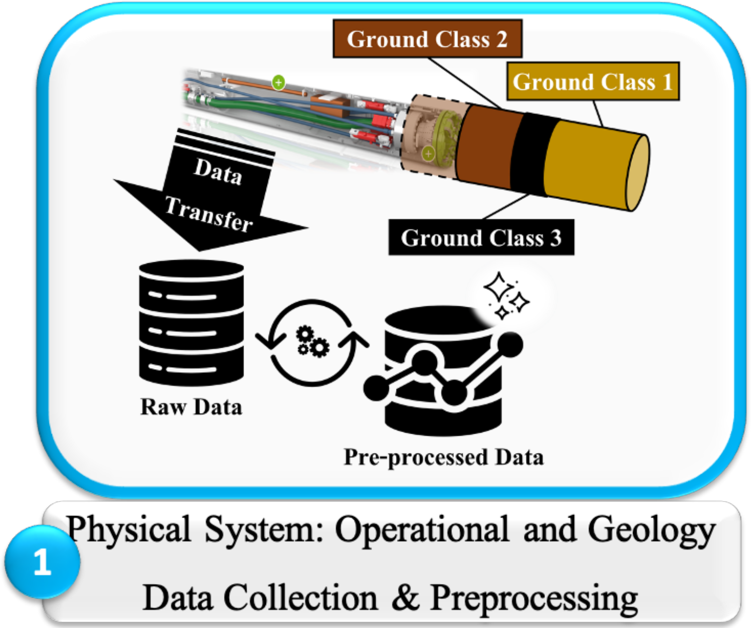
How can we imitate experienced operators and provide decision support to less experienced?

Olga Fink

Decision support system for an intelligent operator of utility tunnel boring machines → resembles imitation learning



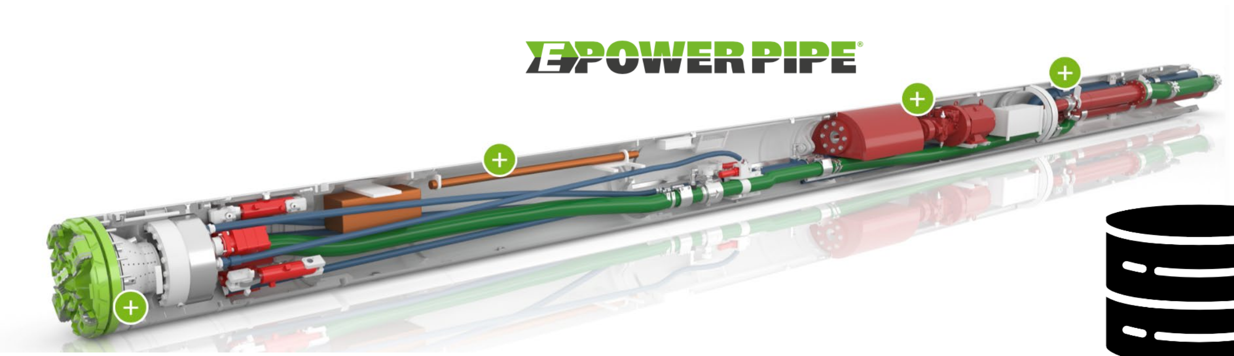
Framework

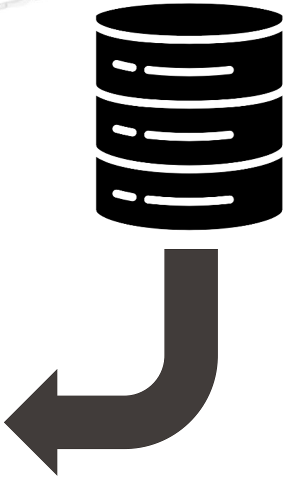

Ground Class 2
Ground Class 1
Data Transfer
Ground Class 3
Raw Data
Pre-processed Data
1
Physical System: Operational and Geology
Data Collection & Preprocessing

Use Case

Collect raw sensor data

1





Cutting Wheel Rot Speed	Feed pump Pressure	…	Oil Temperature
20.1012	1.29482	…	50.1232
21.0333	1.22112	…	50.21999

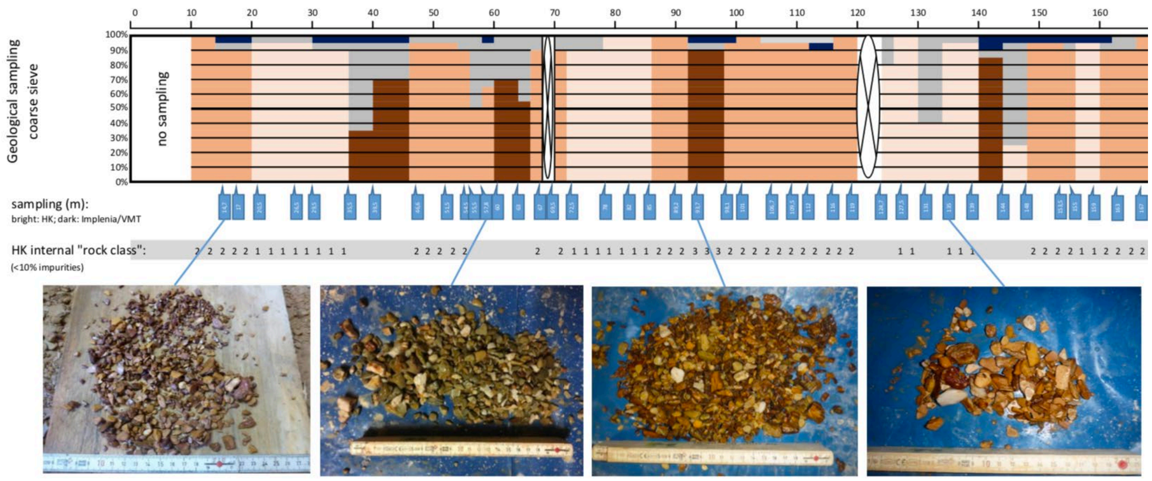
Use Case: Utility tunnels

Geology Profiles

1

- Highly weathered schist (HWS)
- Moderately weathered schist (MWS)
- Slightly weathered Schist (SWS)
- Quartzite
- Iron (hydr-)oxide
- No Material

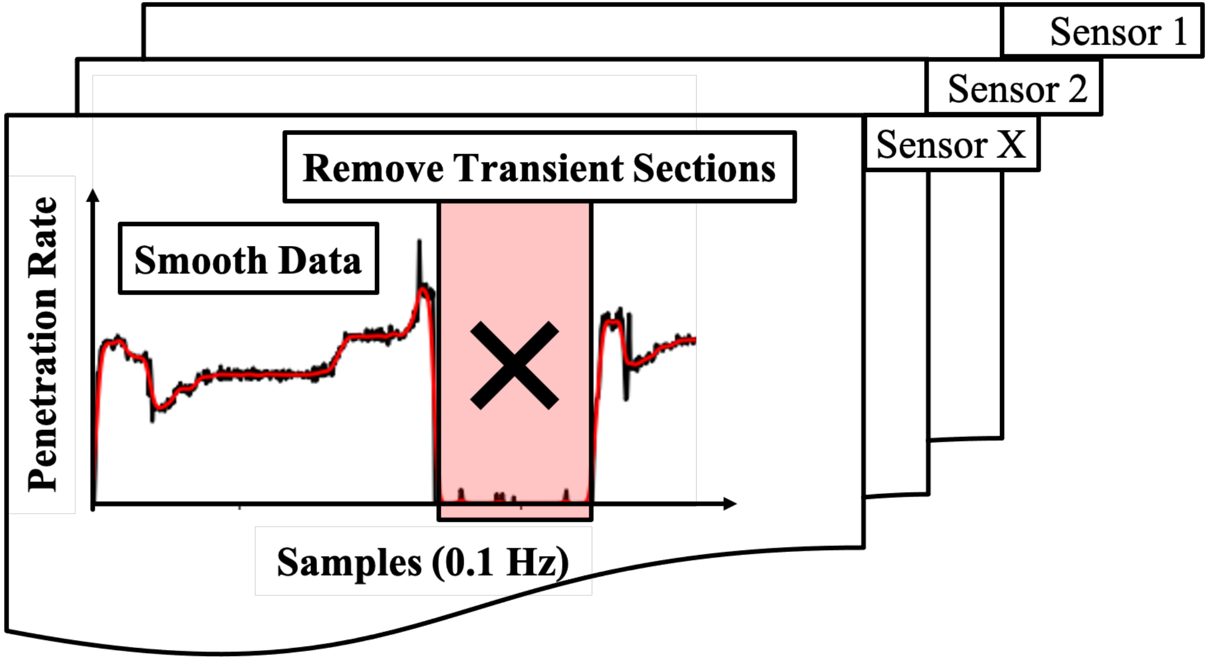
$$GC_1 = Homogeneous\ HWS$$
$$GC_2 = Homogeneous\ MWS$$
$$GC_3 = Homogeneous\ SWS$$



Garcia, G. R., Michau, G., Einstein, H. H., & Fink, O. (2021). Decision support system for an intelligent operator of utility tunnel boring machines. *Automation in Construction*, *131*, 103880.

Use Case
Data Preprocessing

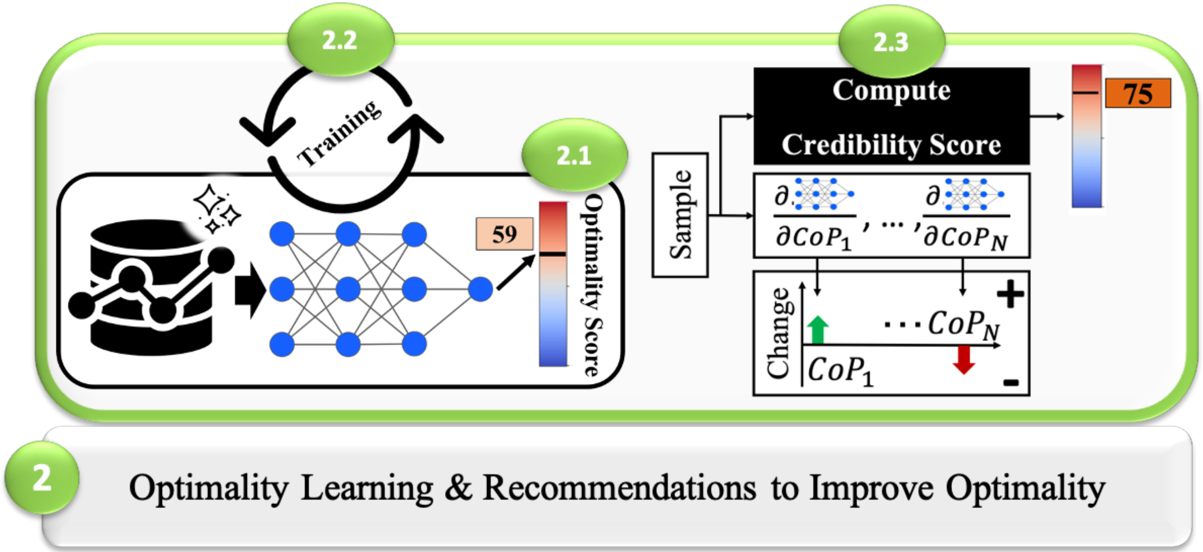
1



List of selected parameters

Parameter Name	Unit
Context Parameters (CxP)	
Steering cylinder 1 pressure	bar
Steering cylinder 2 pressure	bar
Steering cylinder 3 pressure	bar
Steering cylinder 3 pressure (3-B)	
Feed line pressure (on TBM)	bar
Feed line pressure (on pump)	bar
Suction line pressure	bar
High pressure pump pressure (on pump)	
Bentonite pump pressure	bar
Feed line flow rate	m^3/s
Drive line flow rate	m^3/s
High pressure nozzle flow rate	m^3/s
High pressure pump rotational speed	rpm
Bentonite pump rotational speed	rpm
Steering cylinder 1 extension	mm
Steering cylinder 2 extension	mm
Steering cylinder 3 extension	mm
Machine oil temperature	celsius
TBM axial rotation	degrees
Control Parameters (CoP)	
Cutter head rotational speed (CoP_1)	rpm
High pressure water nozzle speed (CoP_2)	bar
Drive line pressure (CoP_3)	bar
Jacking frame thrust (CoP_4)	kN
Feed pump rotational speed (CoP_5)	rpm
Target Parameters (TP)	
Working pressure	bar
Penetration rate	mm/min
(Optimality)	–

Framework



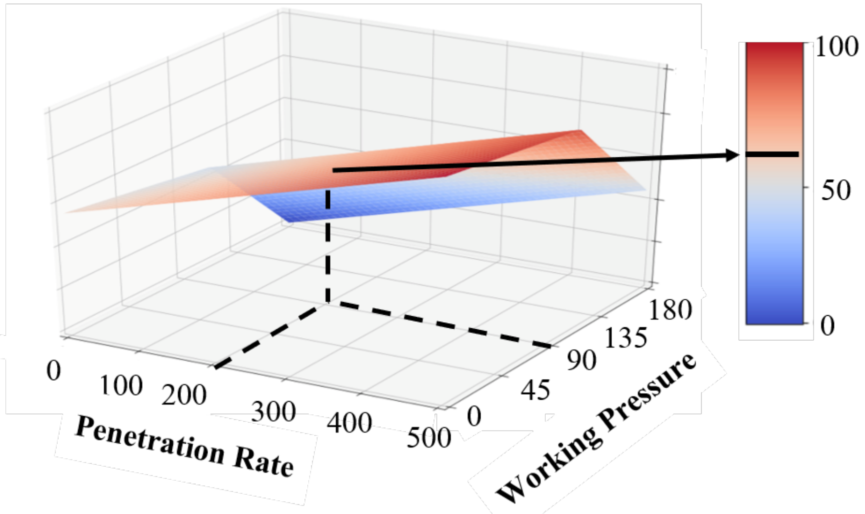
2.2
Training
2.1
59
Optimality Score
Sample
2.3
Compute
Credibility Score
75
∂CoP_1, ..., ∂CoP_N
Change
$\cdots CoP_N$
CoP_1
+
-
2
Optimality Learning & Recommendations to Improve Optimality

Use Case

Define Optimality

2.1

$$f_{GC_i}^{opt}(t) = \begin{cases} \frac{AR_t}{MAR_i} - w_1 \cdot \frac{WP_t}{UB} & \text{if } WP_t \leq MB_i, \\ \frac{AR_t}{MAR_i} - w_1 \cdot \frac{MB_i}{UB} - w_2 \cdot \frac{WP_t - MB_i}{UB} & \text{otherwise.} \end{cases}$$



- AR_t is the advance rate [mm/min] at time t
- WP_t is the working pressure [bar] at time t
- UB is the upper bound of the working pressure (safety threshold before automatic shutdown)
- MB_i is the working pressure margin bound [bar] defined as the observed 90th percentile for the i-th ground class
- MAR_i is the observed maximum advance rate within i-th ground class
- w_1 is the negative penalizing weight on the working pressure when the working pressure is below the margin bound MB_i.
- w_2 is the negative penalizing weight on the working pressure when the working pressure is above the margin bound MB_i. Typically we have $w_2 \gg w_1$.

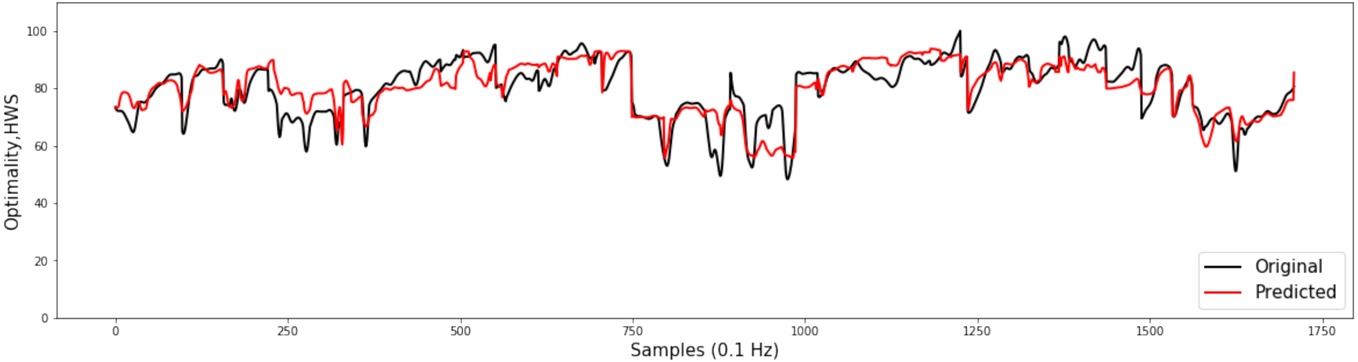
Garcia, G. R., Michau, G., Einstein, H. H., & Fink, O. (2021). Decision support system for an intelligent operator of utility tunnel boring machines. *Automation in Construction*, *131*, 103880.

Use Case

Predict Optimality

2.2

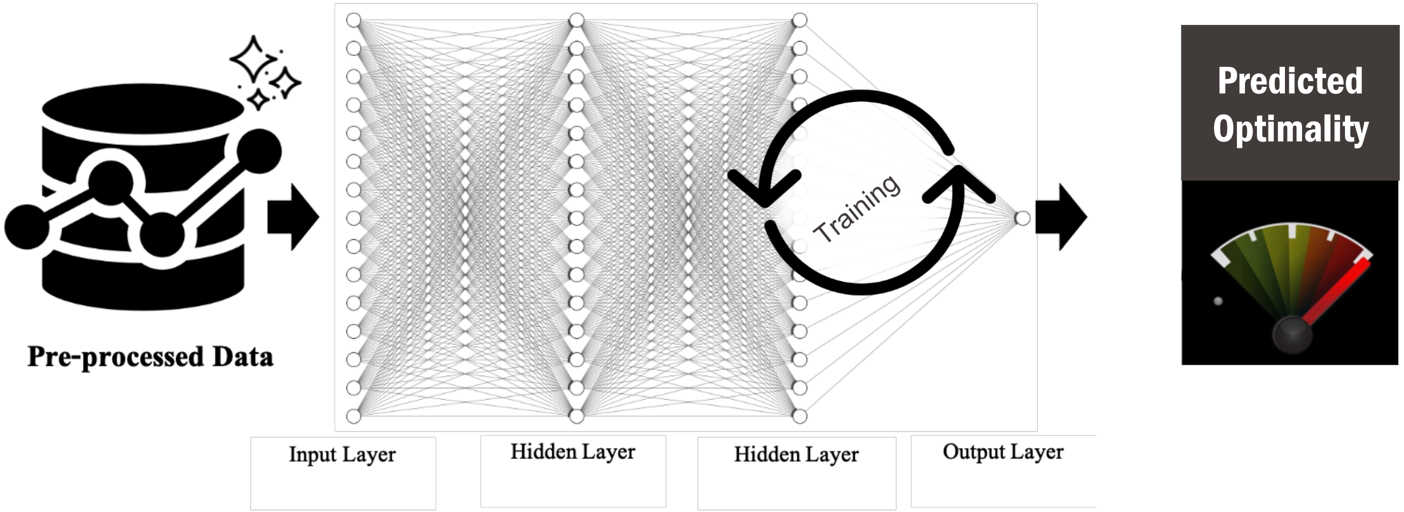
Predicted Optimality



Use Case

Train the Neural Network

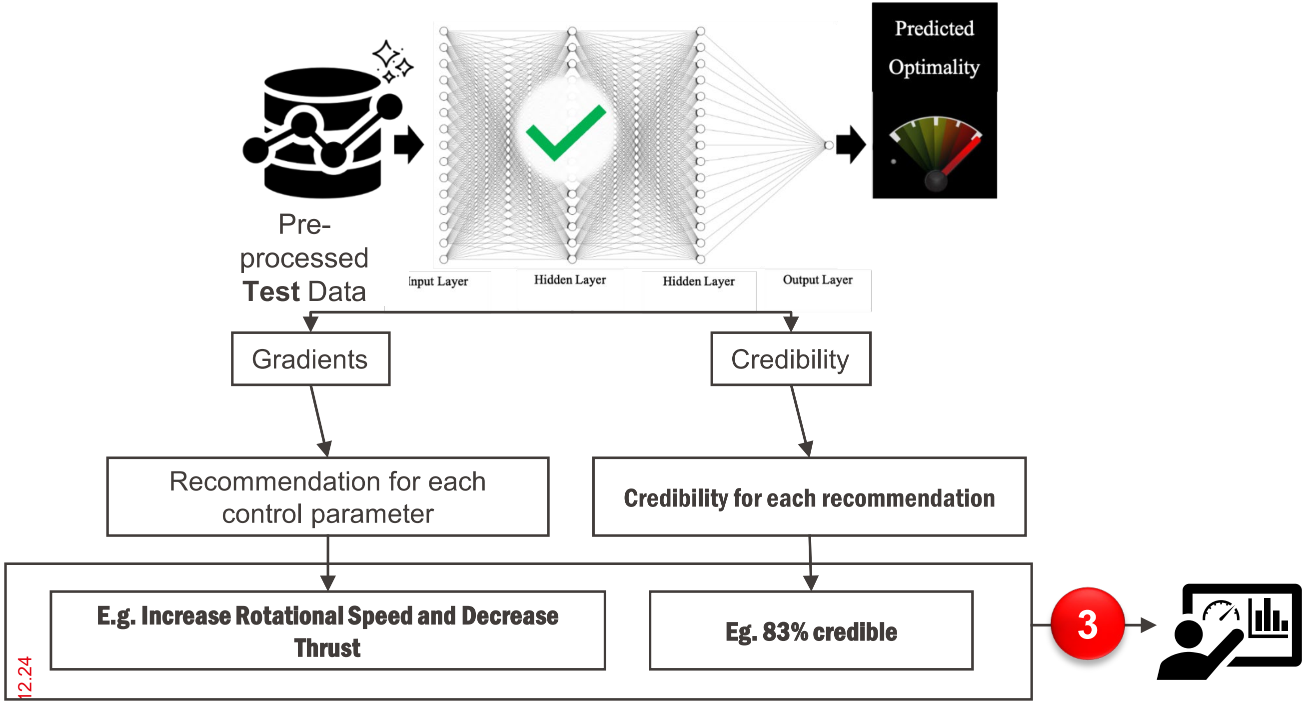
2.2



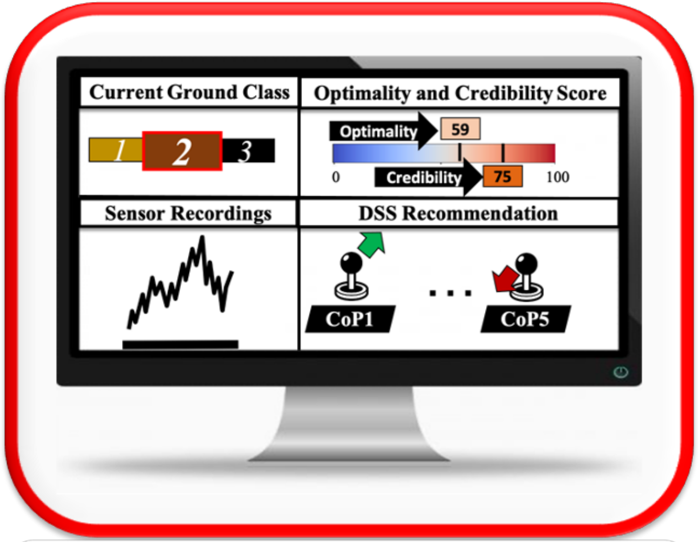
Use Case

Make recommendations and assess Credibility

2.3



Framework



3 Real Time Decision Support

Business models

Possible business models in predictive maintenance

- Sensors a service
- Subscription
- Performance based contracting
- Pay per use
- Guaranteed availability
- Freemium
- Add-on
- Solution provider