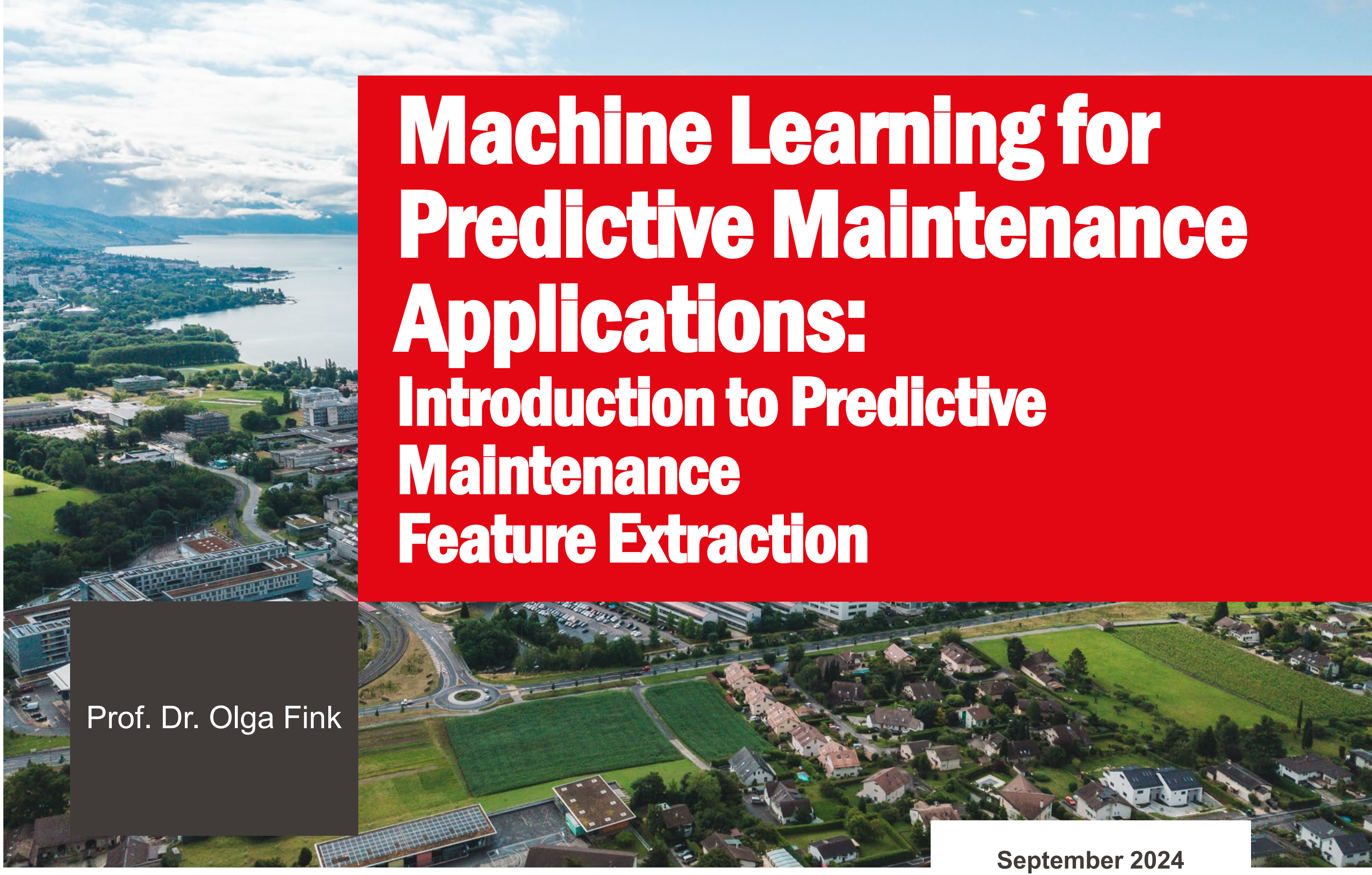


EPFL

Machine Learning for Predictive Maintenance Applications:
Introduction to Predictive Maintenance
Feature Extraction

Prof. Dr. Olga Fink

September 2024

École polytechnique fédérale de Lausanne

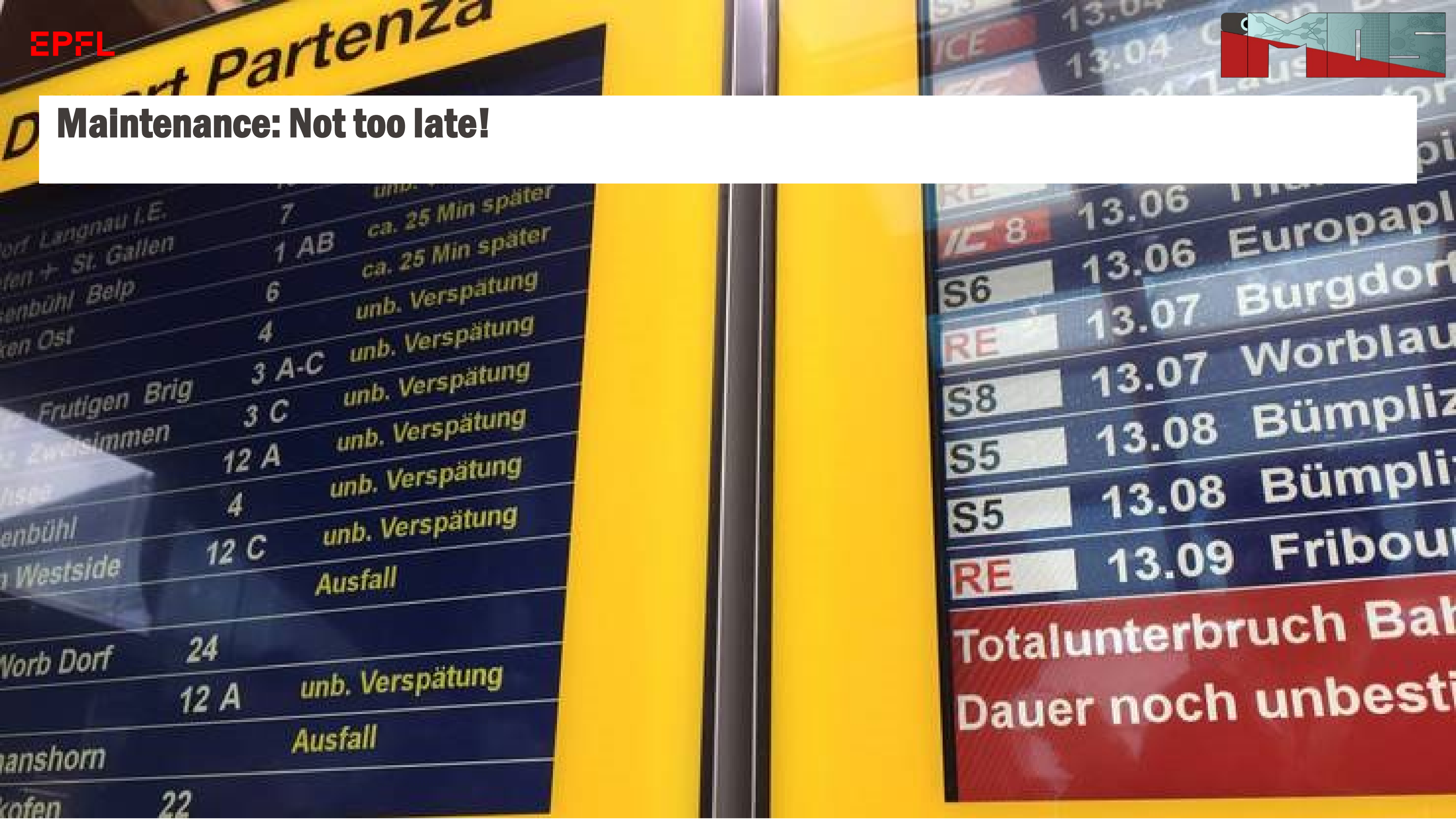
Maintenance

- Combinatoin of all technical administrative and managerial actions during the life cycle of an item intended to retain it in, or restore it to, a state in which it can perform the required function.

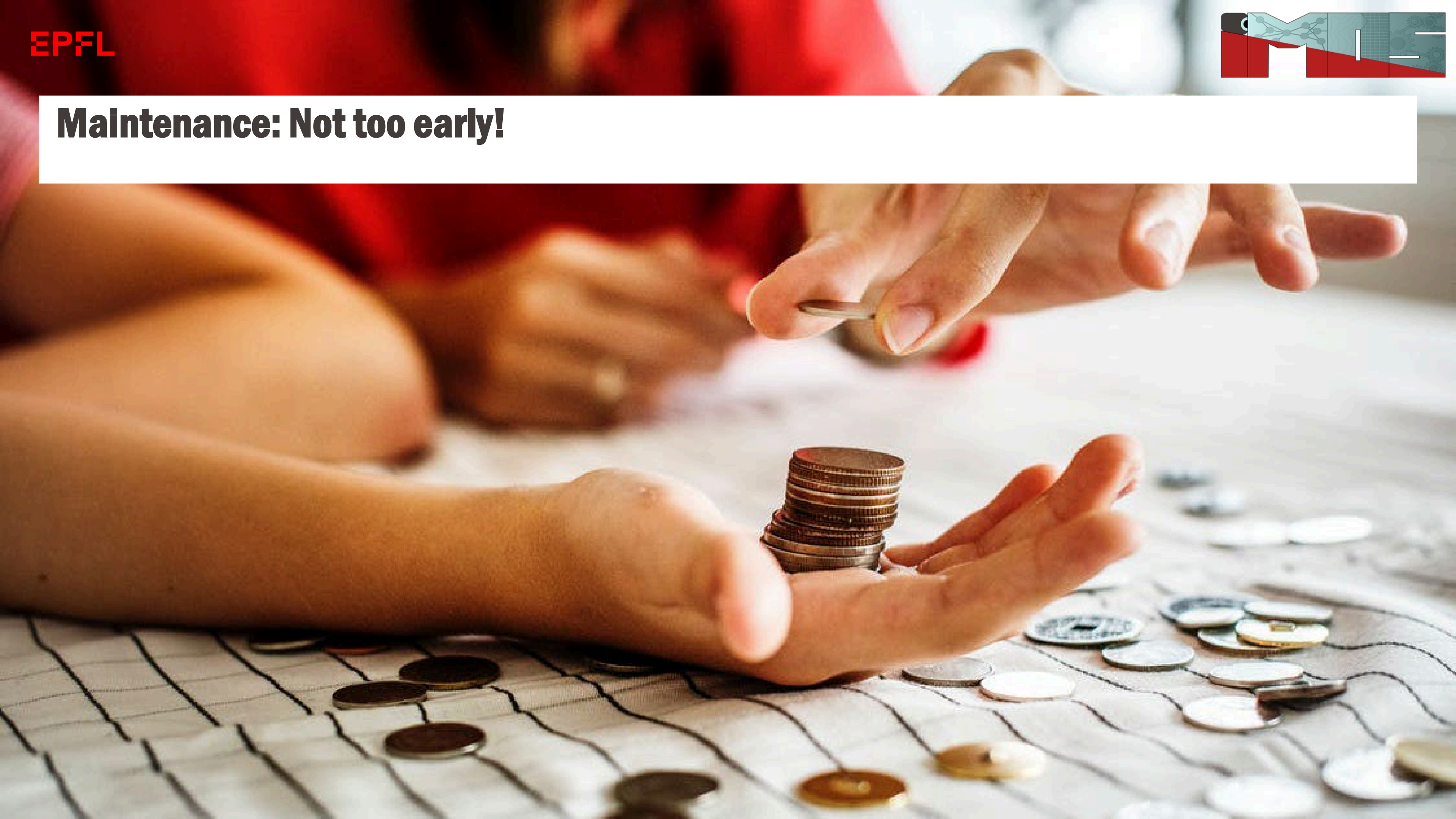
Source: EN 13306

EPFL

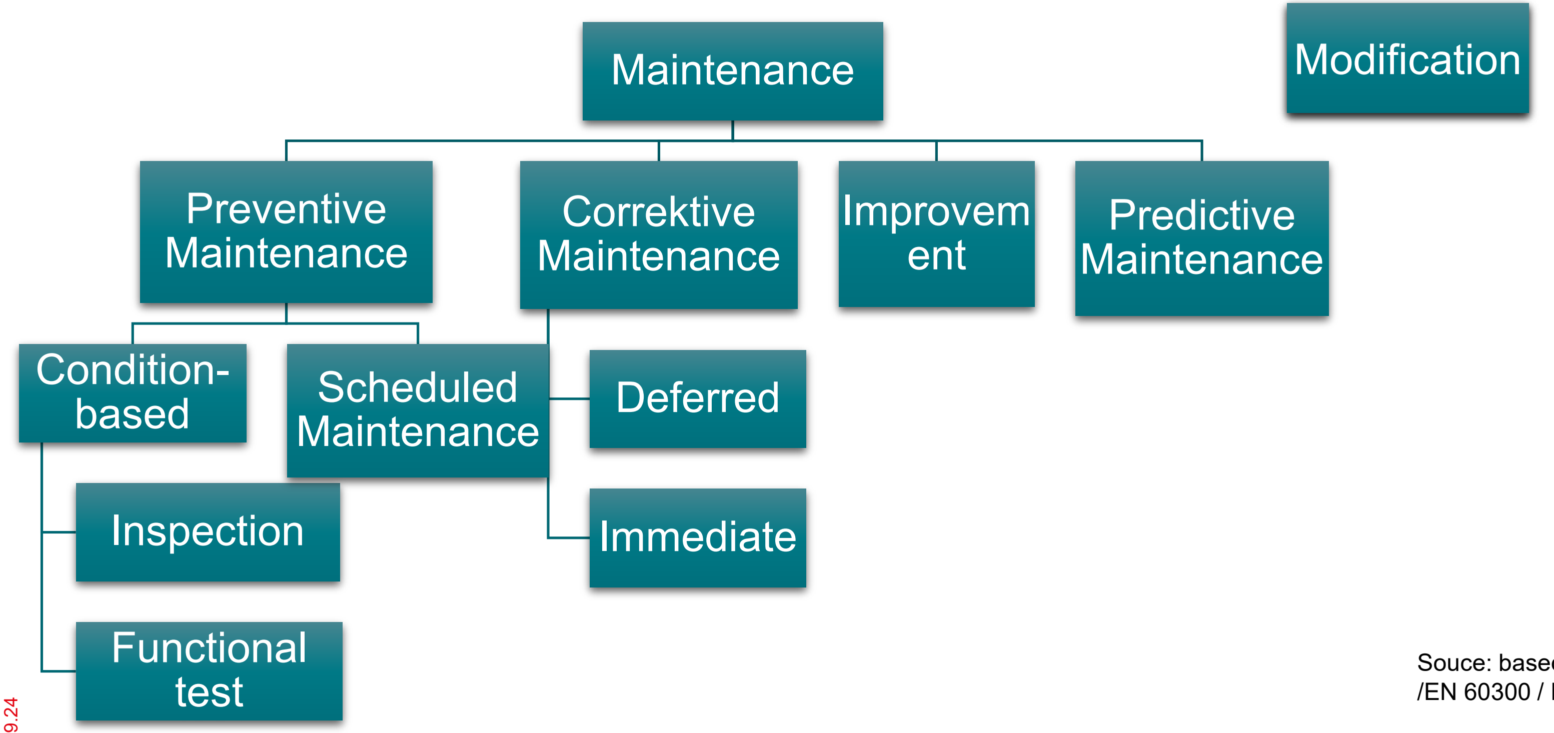
Maintenance: Not too late!



Maintenance: Not too early!

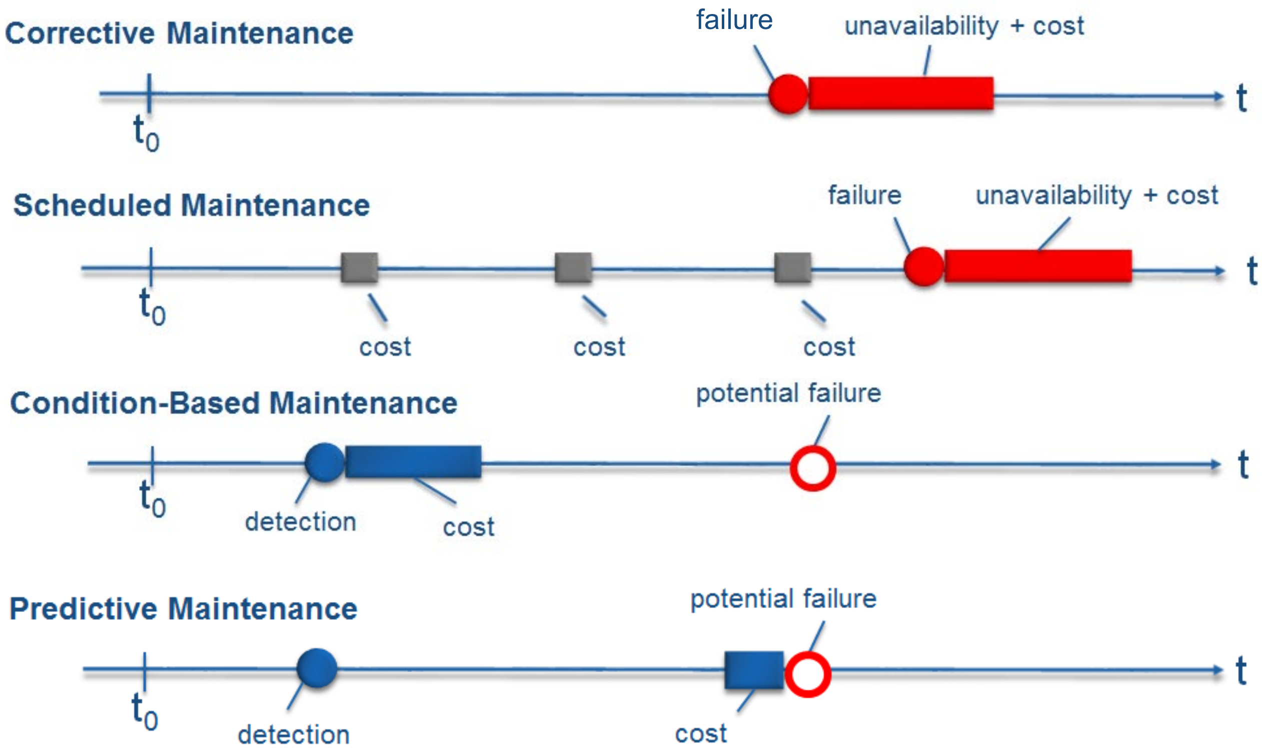


Different types of maintenance



Souce: based on EN 13306 /EN 60300 / DIN 31051

Predictive / Condition-Based Maintenance



What is artificial intelligence?

Artificial intelligence (AI) refers to the ability of a computer or machine to perform tasks that would normally require human intelligence, such as learning, problem solving, decision making, and language understanding.

What is machine learning?

Machine learning allows computers to act and make data-driven decisions without being directly programmed to carry out a specific task.

What is the difference between AI and machine learning?

Machine learning is considered as a part of AI

Other methods for achieving AI include

- rule-based systems,
- evolutionary computation
- expert systems…

Examples of what AI can do today

Solve homework

Explain complex topics in a simple way

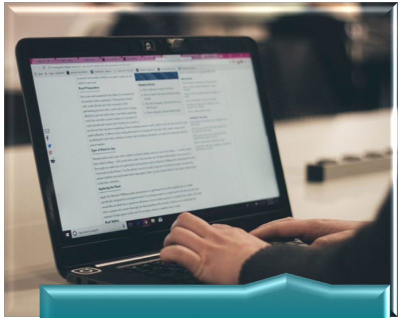
Debug code

Translate between languages

Trade stocks

Write music / film scripts

Generate ideas for party themes, decoration, presents…



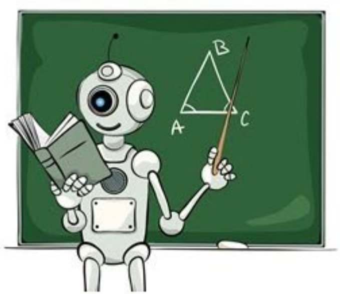
Write blog posts / articles

Machine Learning Research

Machine learning research is part of research on artificial intelligence, seeking to provide knowledge to computers through data, observations and interacting with the world. That acquired knowledge allows computers to correctly generalize to new settings.

Defintion by Yoshua Bengio, Université de Montréal

Types of machine learning

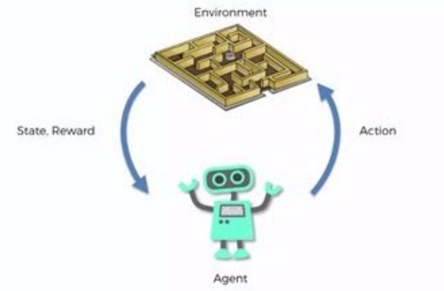


Supervised Learning

VS

Unsupervised Learning

VS



Reinforcement Learning

Generative AI

- **Definition:** Generative AI refers to artificial intelligence systems capable of creating new content such as images, text, music, and more.
- **Key Techniques:**
  - Generative Adversarial Networks (GANs): Uses two neural networks, a generator and a discriminator, that compete to produce realistic outputs.
  - Transformer-Based Models (e.g., GPT): Uses deep learning to generate coherent and contextually relevant text.
- **Applications:**
  - Art and Entertainment: Creating artwork, music, and creative writing.
  - Data Augmentation: Generating synthetic data for training other AI models.
  - Content Creation: Developing marketing copy, news articles, and social media posts.
- **Benefits:**
  - Innovation: Enables the creation of novel and diverse content.
  - Efficiency: Automates content creation, saving time and resources.
- **Challenges:**
  - Ethical Concerns: Issues related to copyright, plagiarism, and misinformation.
  - Quality Control: Ensuring the generated content is accurate and high-quality.

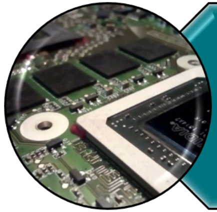
What are Large Language Models (LLMs)

- Large Language Models are a type of artificial intelligence designed to understand and generate human-like text. Think of them as advanced chatbots that can hold conversations, answer questions, write essays, and even create poetry. They are typically trained on vast amounts of text data.
- **How do they work?**
- **Training Data**:
  - LLMs are trained on massive datasets containing text from books, articles, websites, and other written sources. This helps them learn the structure, grammar, and nuances of human language.
- **Learning Patterns**:
  - During training, the model learns to predict the next word in a sentence. For example, if given the phrase "The cat sat on the...", it might predict "mat" as the next word. By repeating this process billions of times, the model learns to generate coherent and contextually appropriate text.
- **Parameters**:
  - The "large" in LLM refers to the number of parameters (adjustable components) the model has. For instance, GPT-3, one of the well-known LLMs by OpenAI, has 175 billion parameters. More parameters generally mean the model can understand and generate more complex and accurate text.
- **What can they do?**
  - **Conversation**: LLMs can chat with users in a natural and engaging way.
  - **Content Creation**: They can write articles, stories, and even code.
  - **Translation**: LLMs can translate text between different languages.
  - **Summarization**: They can summarize long documents into shorter, easy-to-understand versions.

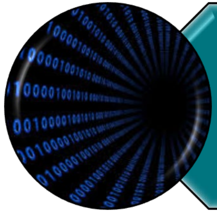
What are foundational models

- Foundational models are large-scale machine learning models that are pre-trained on vast amounts of data and can be fine-tuned for various downstream tasks. These models form the "foundation" upon which more specific applications are built, allowing for efficient transfer learning and rapid deployment across different use cases.
- **Large-Scale Pre-Training**:
  - **Data**: Foundational models are trained on extensive datasets, often comprising diverse and unstructured data such as text, images, and audio.
  - **Architecture**: These models typically use complex architectures like transformers, which can capture intricate patterns and dependencies in the data.
- **Transfer Learning**:
  - **Fine-Tuning**: Once a foundational model is pre-trained, it can be fine-tuned on smaller, task-specific datasets to perform well on particular applications.
  - **Versatility**: This approach allows the model to adapt to a wide range of tasks without requiring training from scratch for each new task.
- **Scalability**:
  - **Parameter Count**: Foundational models often have billions of parameters, making them highly capable but also resource-intensive.
  - **Compute Resources**: Training and deploying these models require significant computational power, often leveraging specialized hardware like GPUs and TPUs.
- **Different types of foundational models:**
  - Education
  - Sciences
  - Health
  - Sustainability / climate
  - Robotics

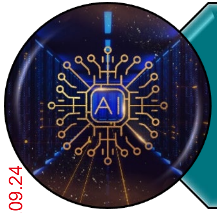
Key success factors for breakthroughs in Artificial Intelligence



Advancement of hardware

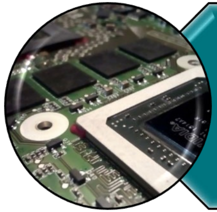


Emergence of “big” data



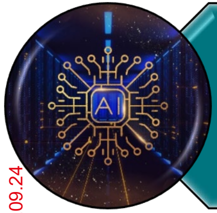
Advancements in machine learning and deep learning

Key success factors for breakthroughs in Artificial Intelligence



Advancement of hardware

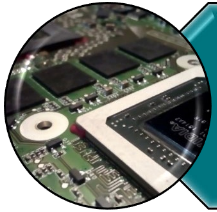
Emergence of “big” data



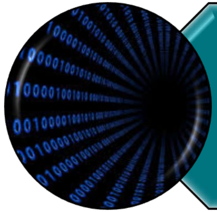
Advancements in machine learning and deep learning

Source: Joel Casutt, SBB

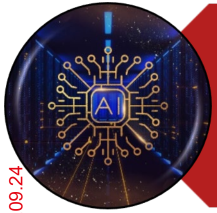
Key success factors for breakthroughs in Artificial Intelligence



Advancement of hardware

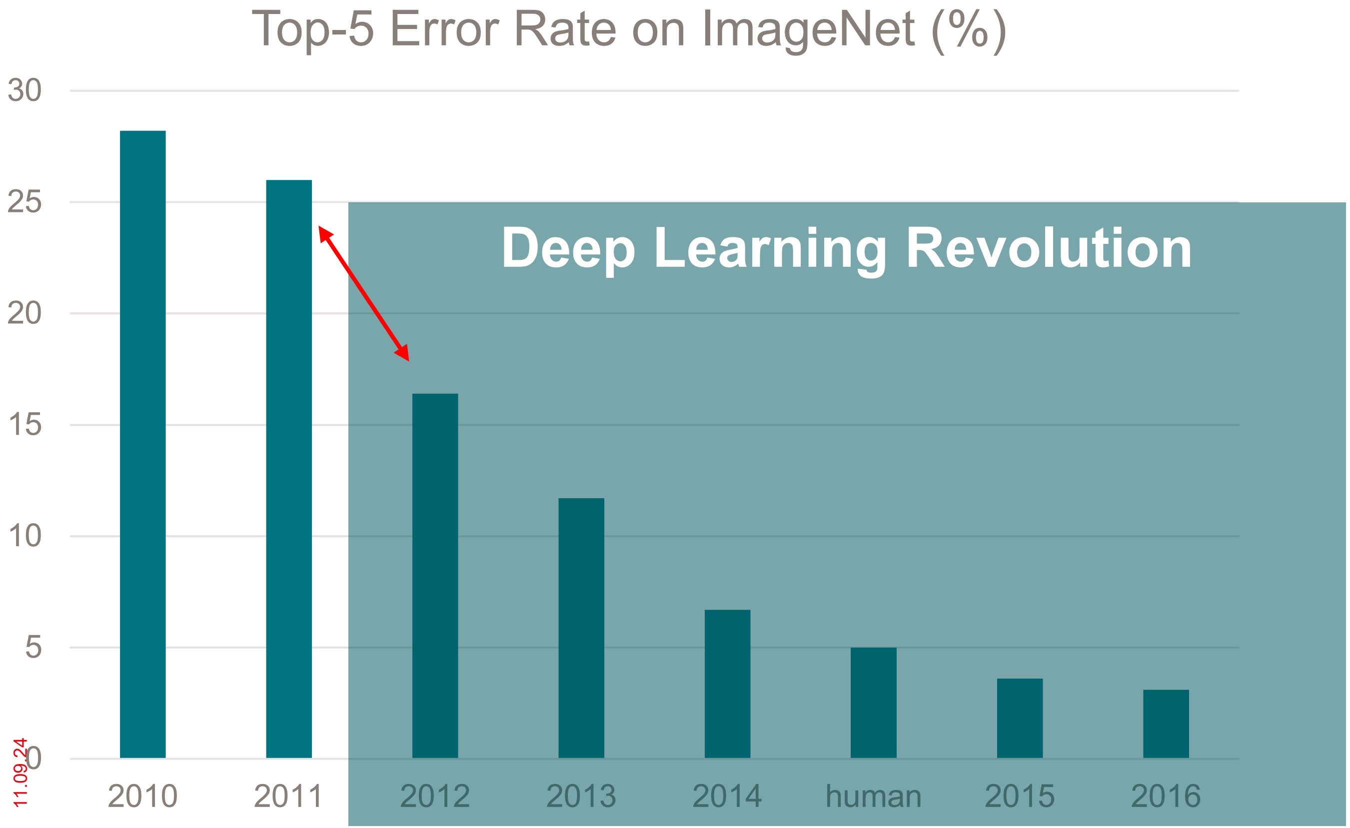


Emergence of “big” data

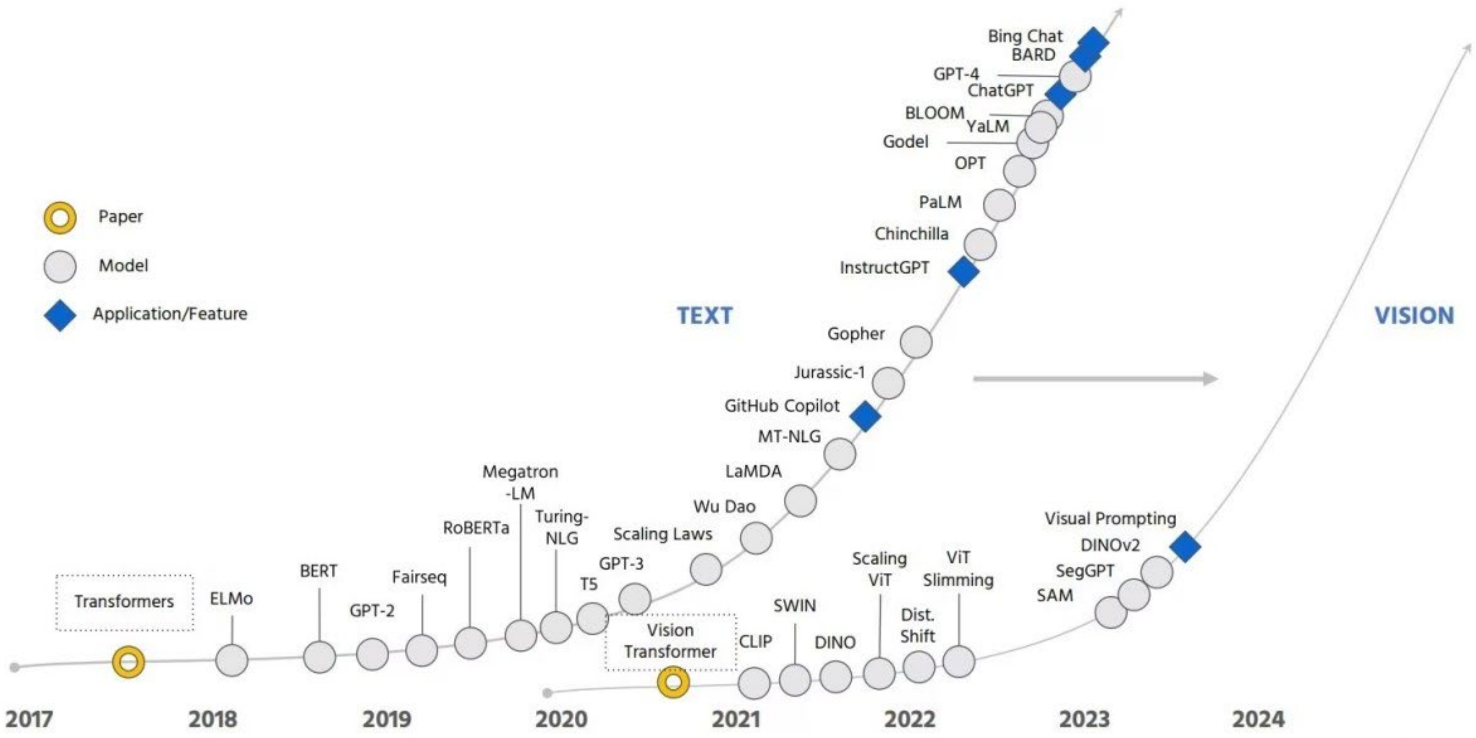


Advancements in machine learning and deep learning

Power of depth

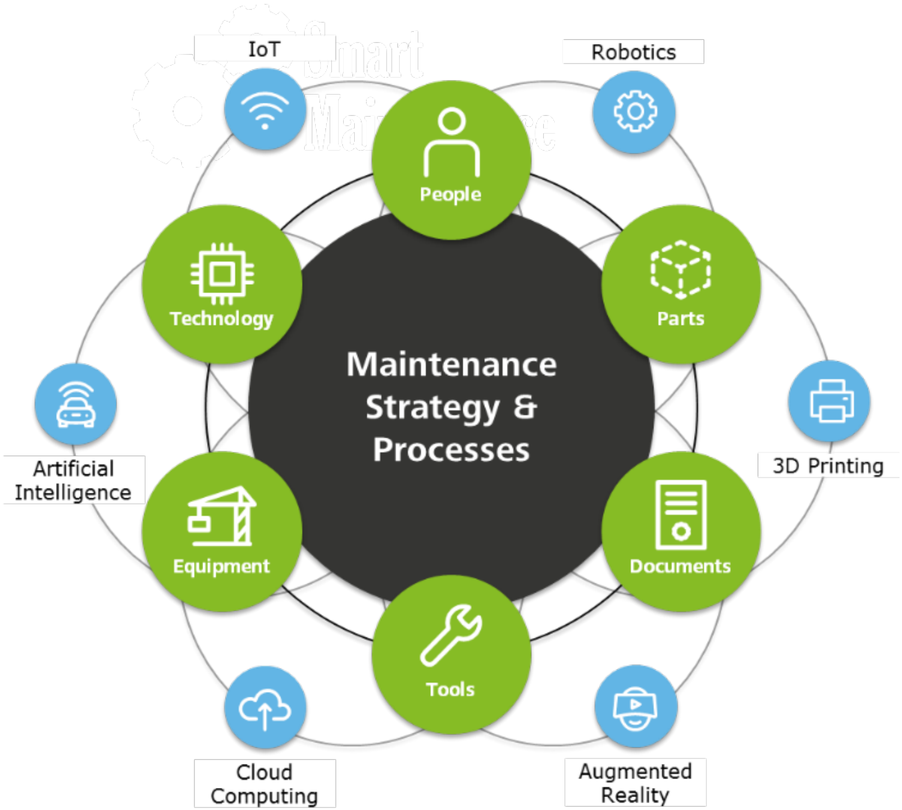


Transformer architectures



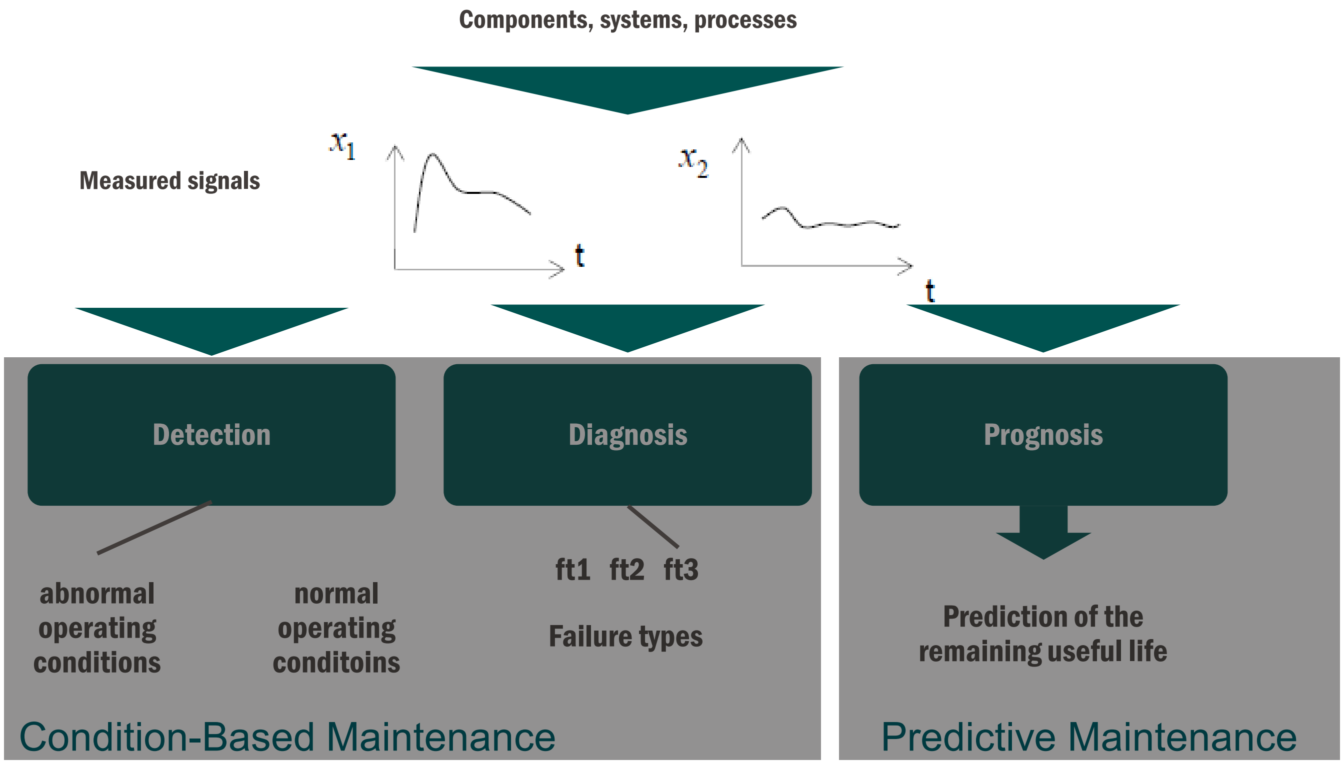
Andrew Ng

Elements of Intelligent Maintenance

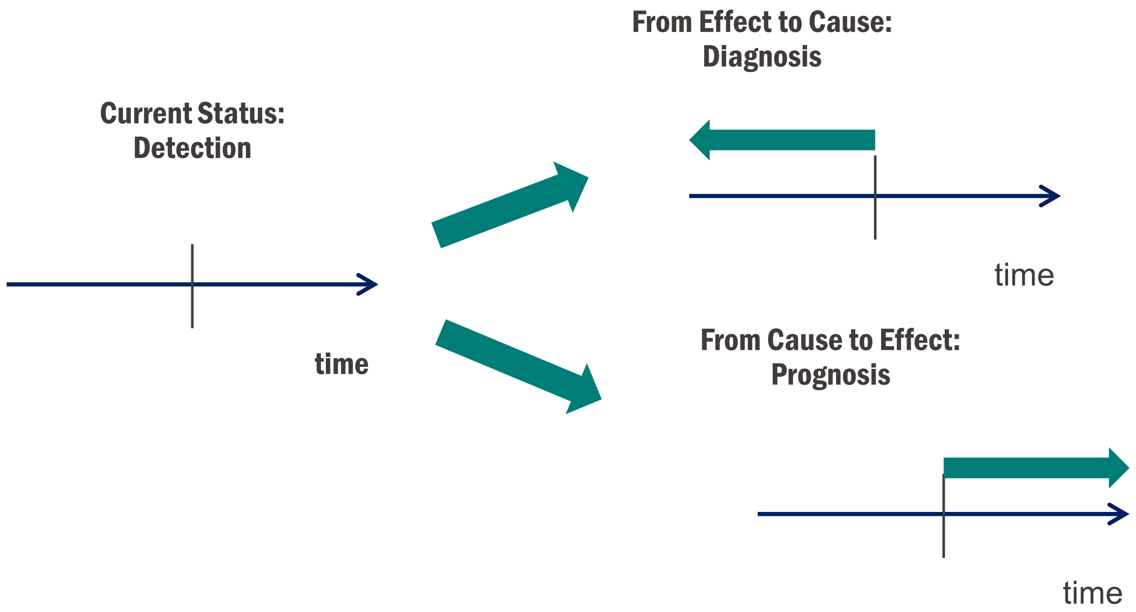


Source: Deloitte

Prognostics and health management (PHM)



Detection, Diagnosis, Prognosis



Prognosis

- Prognosis: Prediction of the Remaining Useful Life of a component
- Remaining Useful Life (RUL) – The amount of time a component can be expected to continue operating within its stated specifications.
- RUL is dependent on future operating conditions
  - Input commands
  - Environment
  - Loads
- Time of Failure (TOF): the time a component is expected to fail (no longer meet its design specifications)

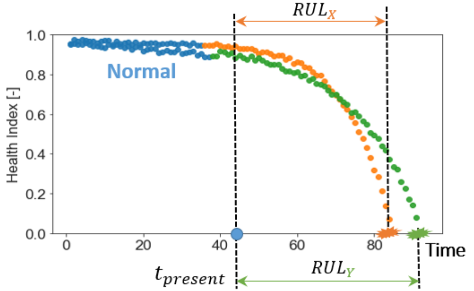
EPFL

Detect, Diagnose and Predict Faulty System Conditions → monitor the health evolution

Shelling

Non-roundness

Fault detection / diagnostics / prognostics

Detection Healthy or faulty operation?	
Diagnostics Are there different faults types? What is the root cause of the faults?	Type 0 → Normal Type 1 → Fault of X* Type 2 → Fault of Y* (* component, system, sensor, etc.)
Prognostics How long the system can continue operating i.e., remaining useful lifetime?	

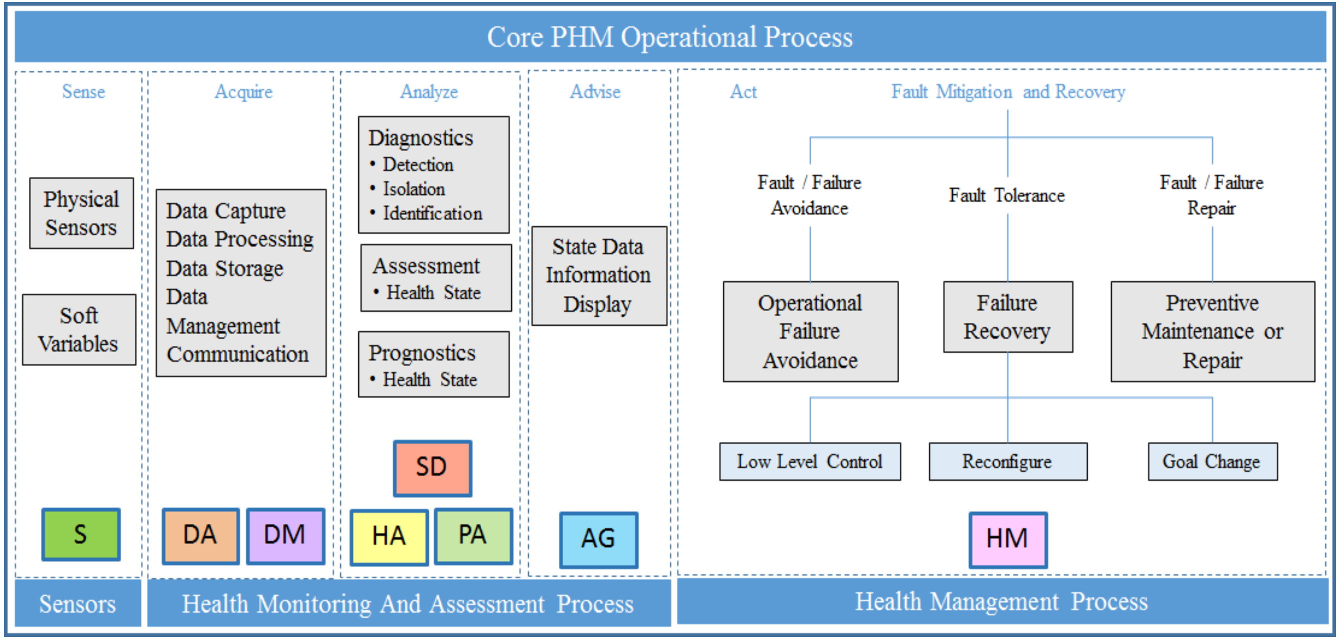
From data acquisition to health management: Prognostics & Health Management (PHM)

PHM Functional Model

Core PHM Operational Processes	Functional Block	Description
Act	Health Management (HM)	This function uses the information generated in the AG to institute actions to return the system to a "healthy state".
Advise	Advisory generation (AG)	This function provides actionable information to operational and maintenance personnel or external systems.
Analyze	Prognostic Assessment (PA)	This function provides future state of health, performance life remaining, or useful remaining life (usage) indicators.
Analyze	Health Assessment (HA)	This function provides information to determine the current state of health of the system.
Analyze	State Detection (SD)	This function evaluates equipment state conditions against normal operating profiles and generates normal or abnormal condition indicators.
Acquire	Data Manipulation (DM)	This function processes and transforms the sensor data and health state information collected by the DA.
Acquire	Data Acquisition (DA)	This function the sensor data and health state information from the system's internal monitors, the system data bus or data recorder.
Sense	Sensors (S)	This involves physical sensors and any "soft" system performance variables available within the system.

Source: P1856™/D31 Standard

PHM system operational view



Source: P1856™/D31 Standard

Predictive vs. Prescriptive Maintenance

Predict the remaining useful life

Anticipate the failure

Reduce the impact of the failure

Determine the optimal point in time for maintenance intervention

What can we do to prolong the remaining useful life?

How can we proactively adjust the operating conditions?

How can we control the process parameters?

Five levels of condition-based and predictive maintenance

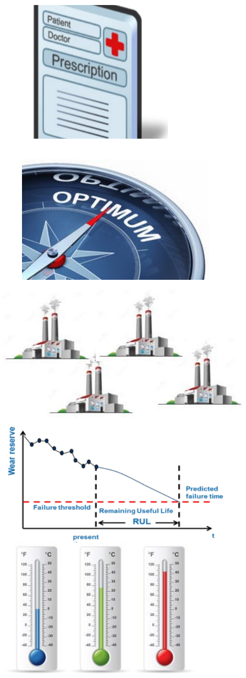
V. Prescrip-tive Level
- Prolong the component's lifetime
- Self configuration, optimization and adjustment

IV. Decision support Level
- Prioritize and optimize maintenance decisions
- Remote visualization

III. Fleet Level
- Fleets of machines
- Cyber-physical systems
- Peer-to-peer Monitoring
- Components under different operating conditions

II. PHM Level
- Fault detection, diagnostics and prediction of the remaining useful life
- Degradation and performance prediction

I. Condition Monitoring Level
- Effective sensor selection
- Connection + data aggregation
- Internet of things infrastructure



“The greatest challenge to any thinker is stating a problem in a way that will allow a solution”

Betrand Russell

Learning goal

- Define the learning problem in a way that allows its solution based on existing constrains such as lack of fault samples
- Design data-driven predictive maintenance applications for complex engineered systems from raw condition monitoring data
- Assess / Evaluate the performance of the applied algorithms
- Choose machine learning algorithms for fault detection, diagnostics and prognostics
- Interpret the results of the algorithms

Performance Assessment

- Performance will be assessed during the semester based on
  - 5 exercises, requiring the students to perform defined sub-tasks for designing a predictive maintenance system (60% of the final grade in total)
    - Feature engineering (12.5%)
    - Fault detection (12.5 %)
    - Prognostics (12.5 %)
    - Domain adaptation (12.5 %)
    - XAI (10%)
  - Final project: Report (including the implementation) and presentation of a real case study of designing a predictive maintenance system based on raw condition monitoring signals of a complex engineered system (30%) (→ presentation on 12.12.24 /16.12.24)
  - 2 quizzes on the content of the course → 10% of the grade

EPFL

(Preliminary) Course Schedule

Week	Date lecture	Date excercise	Lecture	excercise / graded Excercise	Quizzie / Project
1	9/9/2024	12/9/2024	Introduction to predictive maintenance / Basic Principles of PM, problem formulations / Pre-processing of raw condition monitoring data / Feature extraction;	Introduction to sciPy + Pytorch	
2	16/9/2024	19/9/2024	How to approach data science projects? Problem definition / Feature selection / dimensionality reduction methodology	Introduction to Excercise 1 / introduction to good practice data science	
3	23/9/2024	26/9/2024		Repetition signal processing, support slot excercise 1	
4	30/9/2024	3/10/2024	Fault diagnostics / Residual-based detection / Feature Learning	Submission Graded excercise 1,Introduction to graded excercise 2	
5	7/10/2024	10/10/2024	One-Class Classifiers /Health indicators / Self-supervised learning / performance evaluation	Debfriefing graded excercise 1, Support slot graded excercise 2	
6	14/10/2024	17/10/2024	Prognostics / Hyperparameter tuning	Submission graded excercise 2, Introduction to graded excercise 3	Quizzie1/ Intro to Final Project (Data Final Project)
7	21/10/2024	24/10/2024	Semester break	Semester break	
8	28/10/2024	31/10/2024	Domain adaptation approaches for PHM	Excursion ?	
9	4/11/2024	7/11/2024	Federated learning in PHM/ Explainable ML algorithms	Submission graded excercise 3, Introduction to graded excercise 4	
10	11/11/2024	14/11/2024	Physics-informed ML Models / GNN	Debfriefing graded excercise 3, Support slot graded excercise 4, Excercise Federated learning	Milestone Final Project
11	18/11/2024	21/11/2024	Semi-supervised learning for fault diagnostics and prognostics / Generative models	Submission Graded excercise 4, Introduction graded Excercise XAI	
12	25/11/2024	28/11/2024	Fleet approaches / Prescriptive maintenance concepts	Debfriefing graded excercise 4, Submission graded excercise 5; Excercise GNN	
13	2/12/2024	5/12/2024	Decision support	Support slot final project	Quizzie2
14	9/12/2024	12/12/2024	Benefits and costs of PM	Presentation final project	
14	16/12/2024	19/12/2024	Presentation final project		

EPFL

Exercise team

Dr. Florent Forest

Dr. Zhan Ma

Ismail Nejjar

Vinay Sharma

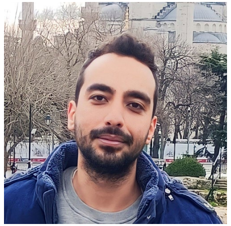
Raffael Theiler

Mengjie Zhao

Amaury Wei

Leandro von Kannichfeldt

Han Sun



Keivan Fagih Niresi

Sergei Garmaev

Chenghao Xu

Zepeng Zhang

Typical steps to follow in a machine learning (ML) project (1/3)

1. Problem Definition and Understanding
   - Clearly articulate the problem you aim to solve.
   - Identify the objectives and success criteria.
   - Ask the right questions: Identify the questions your data can answer to solve the business problem.
   - Understand the data: Learn the context, limitations, and opportunities within the available data.
2. Identify the Value:
   - Determine the potential value and impact of solving the problem (both from business and from the scientific perspective)
   - Assess how the solution will benefit stakeholders and align with business goals.
3. Collect Data:
   - Gather relevant data from various sources.
   - Make sure that your data is representative for the test data (application data) → be aware of the variability of the operating conditions (→ domain shift)
   - Ensure data quality and completeness.
   - Acquire labels for your data if possible (ensure the quality of labels)
4. Explore and Understand the Data:
   - Perform exploratory data analysis.
   - Visualize data to understand patterns and relationships.
   - Understand the data distribution (Explore the statistical properties, distributions, and trends in the data)
   - Identify data quality issues (Look for missing values, outliers, and inconsistencies that may affect analysis).
   - Generate hypotheses (Form hypotheses based on patterns observed during data exploration).

Typical steps to follow in a machine learning (ML) project (2/3)

5. Prepare Data:
   - Clean the data (handle missing values, outliers).
   - Feature engineering (create meaningful features, transform existing ones).
   - Split data into training, validation, and test sets.
   - Scaling and normalization (Prepare the data for modeling by applying appropriate scaling techniques)
   - Address class imbalances: If needed, use techniques like oversampling, undersampling, or synthetic data generation to balance the dataset.
6. Select and Train Models:
   - Select appropriate models (Choose models based on the problem type (regression, classification, clustering, etc.) and data characteristics)
   - Split the dataset (Use train-test splits (or cross-validation) to ensure your model's generalization to unseen data)
   - Baseline model (Start with a simple model as a baseline to compare more complex models).
   - Tune Hyperparameters (Optimize model hyperparameters for better performance).
   - Iterate and improve (Experiment with different models and tuning hyperparameters)
   - Avoid overfitting (Implement regularization techniques or apply cross-validation to avoid overfitting the training data).
7. Evaluate Models:
   - Choose appropriate metrics (Select evaluation metrics that are relevant to the problem (e.g., accuracy, precision, recall, F1-score, RMSE))
   - Validate on unseen data (Always validate your model on a test set or through cross-validation to assess its performance)
   - Compare models: Compare different models based on performance metrics and business value.

Typical steps to follow in a machine learning (ML) project (3/3)

8. Model Interpretation
   - What do the results mean?
   - Explainability (Ensure that you understand how your model works, and use XAI techniques such as SHAP values or feature importance to explain the results).
   - Check assumptions (Ensure that the model's assumptions hold and are consistent with the problem and data context)
9. Test Model:
   - Evaluate the final model on the test dataset to gauge its real-world performance.
10. Deploy Model:
    - Integrate the model into a production environment.
11. Monitor and Maintain:
    - Continuously monitor model performance.
    - Update the model as needed based on new data and changing conditions.
    - Monitor how your model is actually used
12. Documentation and Communication
    - Document thoroughly (Keep detailed notes on every step, including data sources, preprocessing steps, model choices, evaluation, and deployment procedures)
    - Communicate results (Present findings to stakeholders in a clear, actionable manner. Tailor communication based on the audience (e.g., technical team vs. business executives).)

Most importantly:

- **KEEP IT AS SIMPLE AS POSSIBLE!**
- If a linear regression model solves your problem, don't opt for more complex models.
- Evaluate whether applying rules or thresholds is enough to make the decision.
- Keep the decision maker in mind, and focus on how to present model outputs clearly.

PHM Process

Data Acquisition (DA)

↓

Data Manipulation (DM)

↓

State Detection (SD)

↓

Health Assessment (HA)

↓

Prognostics Assessment (PA)

↓

Advisory Generation (AG)

Possible means of condition monitoring

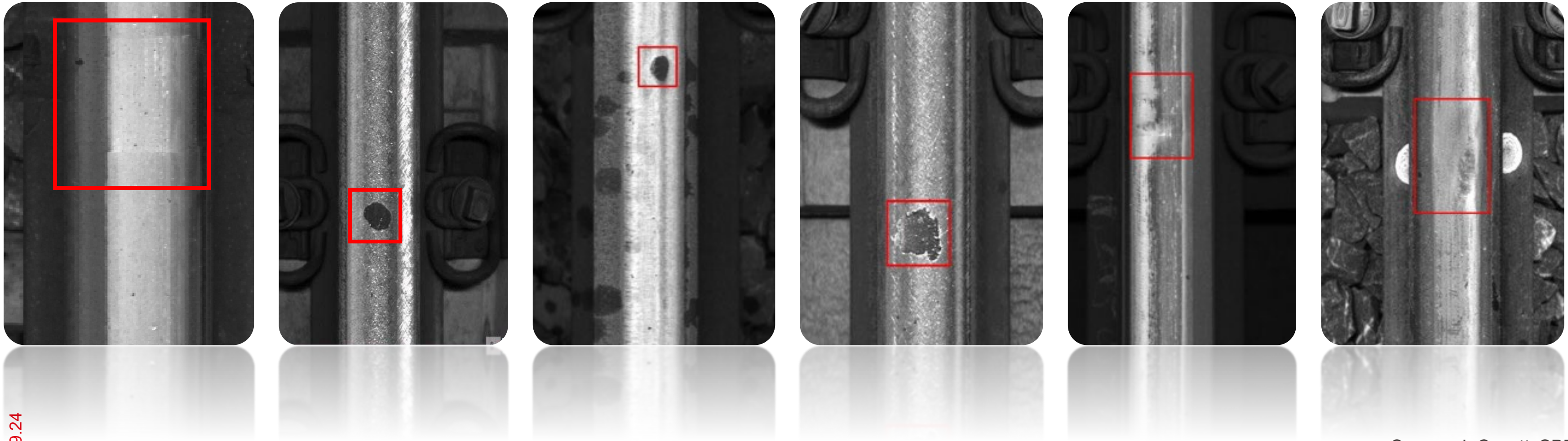
- Thermography
- Measurement of compressed air consumption
- Ultrasonic testing technology
- Oil quality monitoring
- Current consumption measurement
- Vibration diagnosis
- Acoustic emission monitoring
- Computer vision

Video inspection: example SBB (Swiss Federal Railways)

Source: J. Bisang, SBB

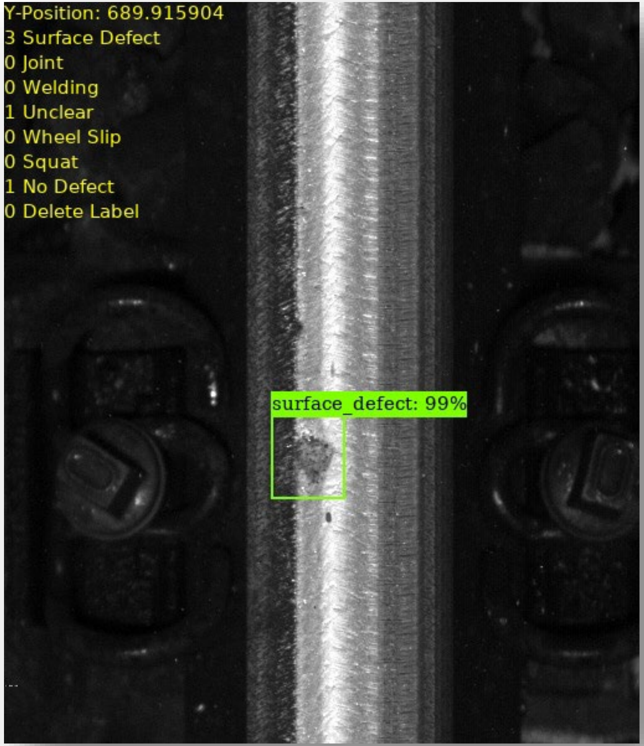
Defect detection of track: example SBB (Swiss Federal Railways)

Welding | Plastic Particle | Surface Defect | Chewing Gum | Squat | Wheel Slip



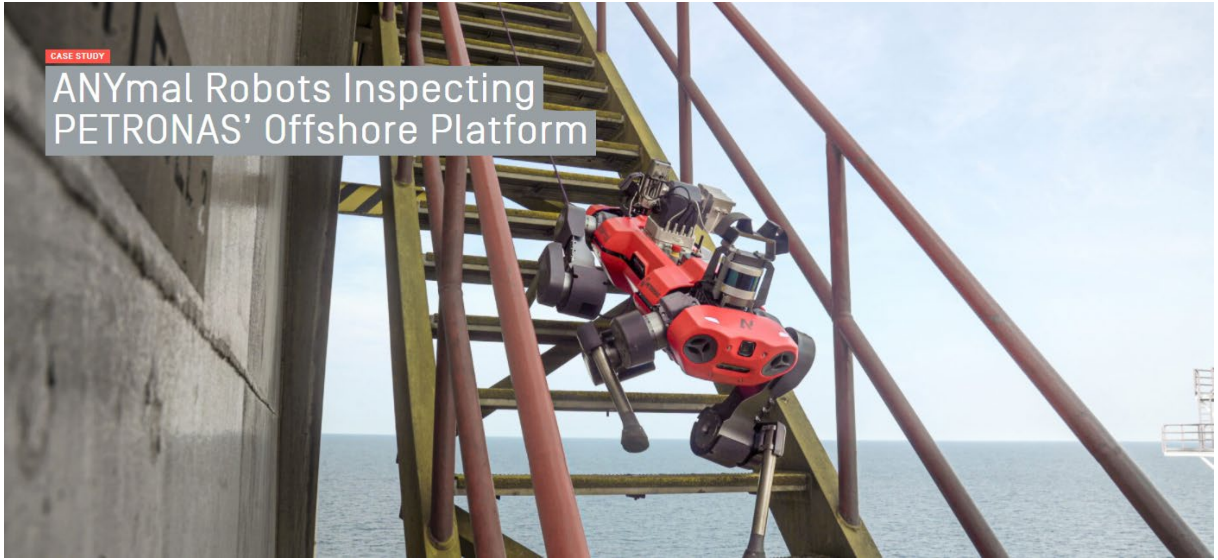
Source: J. Casutt, SBB

Defect detection of track: example SBB



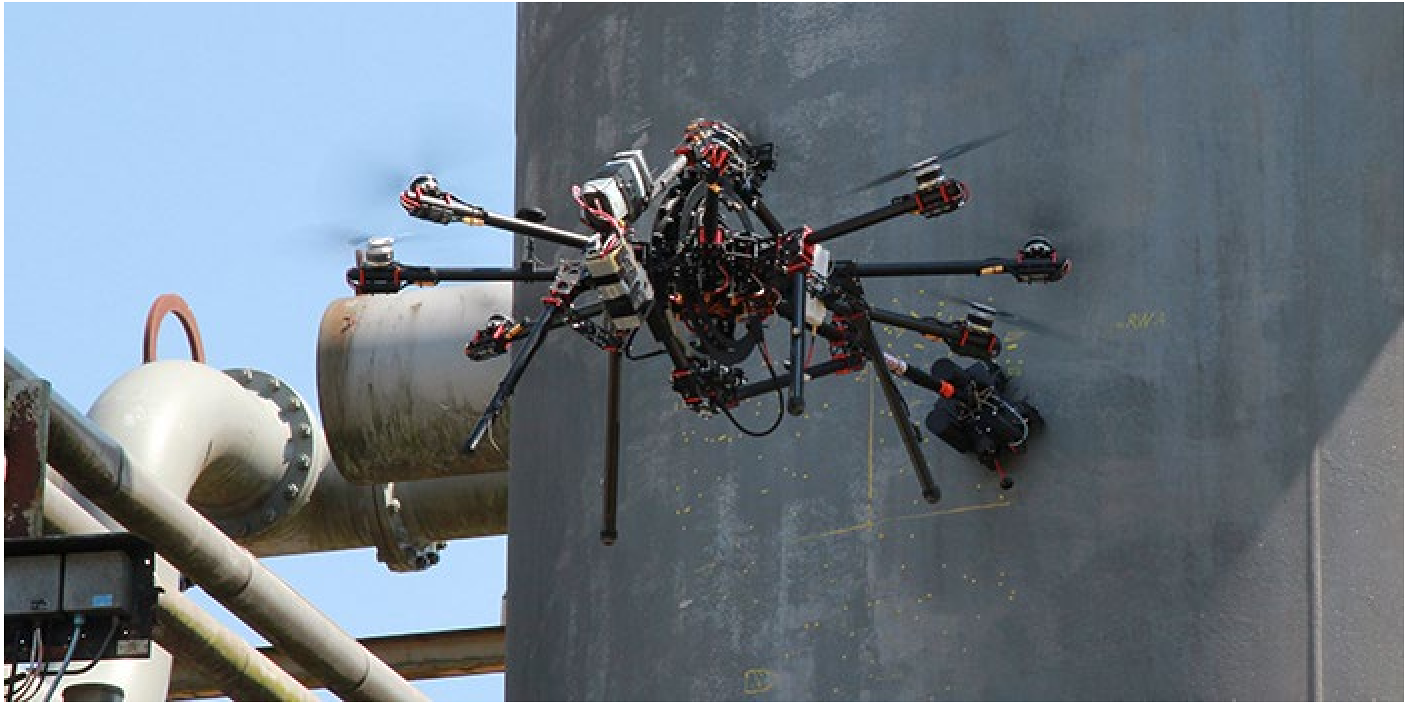
Source: J. Casutt, SBB

Inspection by robots



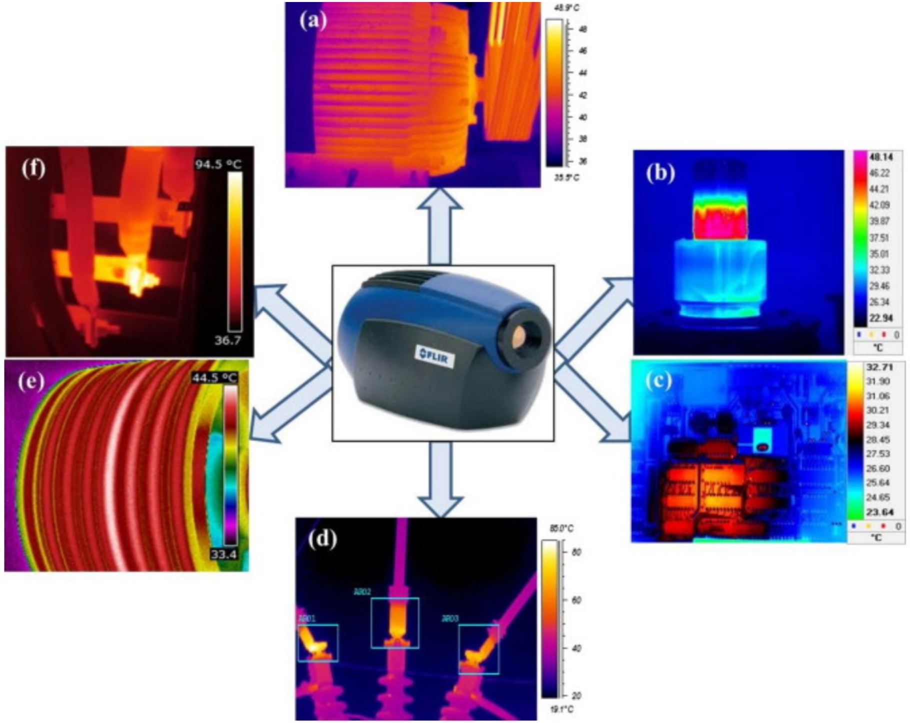
Source: Anybotics

Drone monitoring



Source: www.dtu.dk

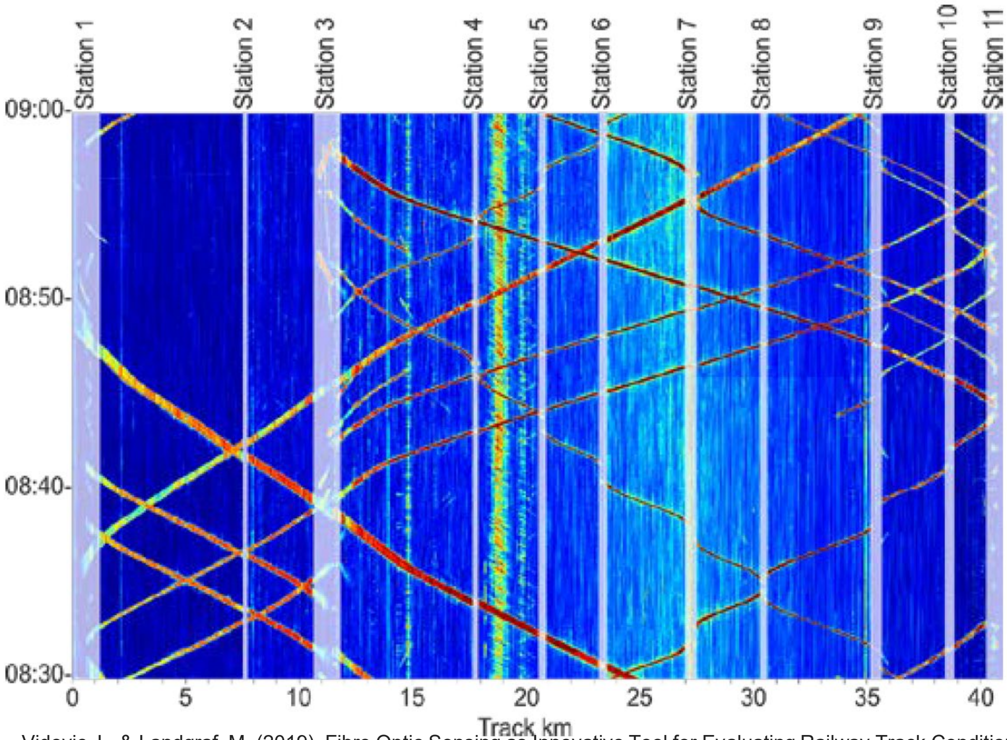
Infrared thermography for condition monitoring



Various condition monitoring applications of infrared thermography: (a) Monitoring of machineries where abnormal surface temperature distribution is an indication of a probable flaw. (b) Inspection of liquid levels in industrial components. (c) Inspection of printed circuit boards. Localized defects like short circuits or current leakages produce hot-spots which can be easily detected by infrared thermography. (d) Typical thermal images of a transformer circuit breaker where the faulty regions can be clearly seen as hot-spots. (e) Inspection of shaft belt where the thermal anomaly is due to over-tightening of a belt. (f) Condition monitoring of three phase electrical panel where local hot spots are developed due to load imbalance.

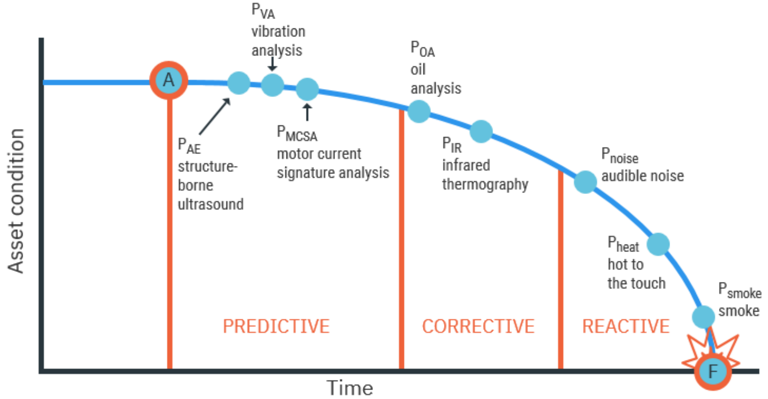
Source: Infrared thermography for condition monitoring – A review

Distributed Acoustic Sensing (fibre optic cable sensing) for detecting defects in switching mechanisms



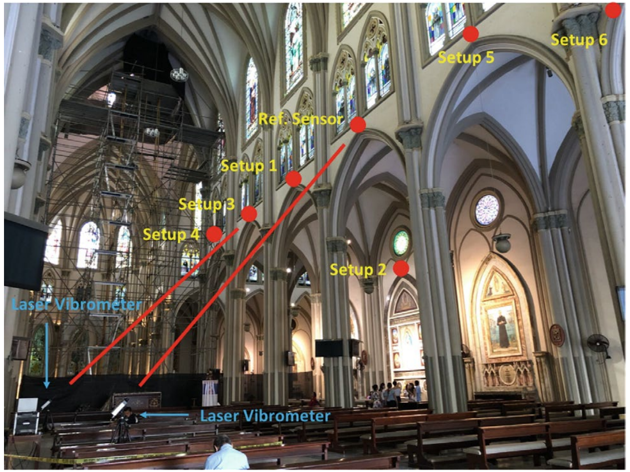
Vidovic, I., & Landgraf, M. (2019). Fibre Optic Sensing as Innovative Tool for Evaluating Railway Track Condition?. In *International Conference on Smart Infrastructure and Construction 2019 (ICSIC) Driving data-informed decision-making* (pp. 107-114). ICE Publishing.

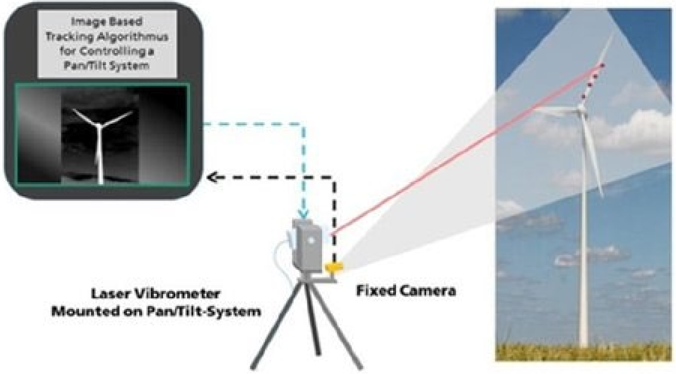
Acoustic emission monitoring

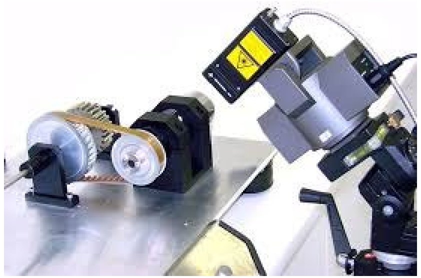


Source: Semiotic labs
Kistler

Laser vibrometer

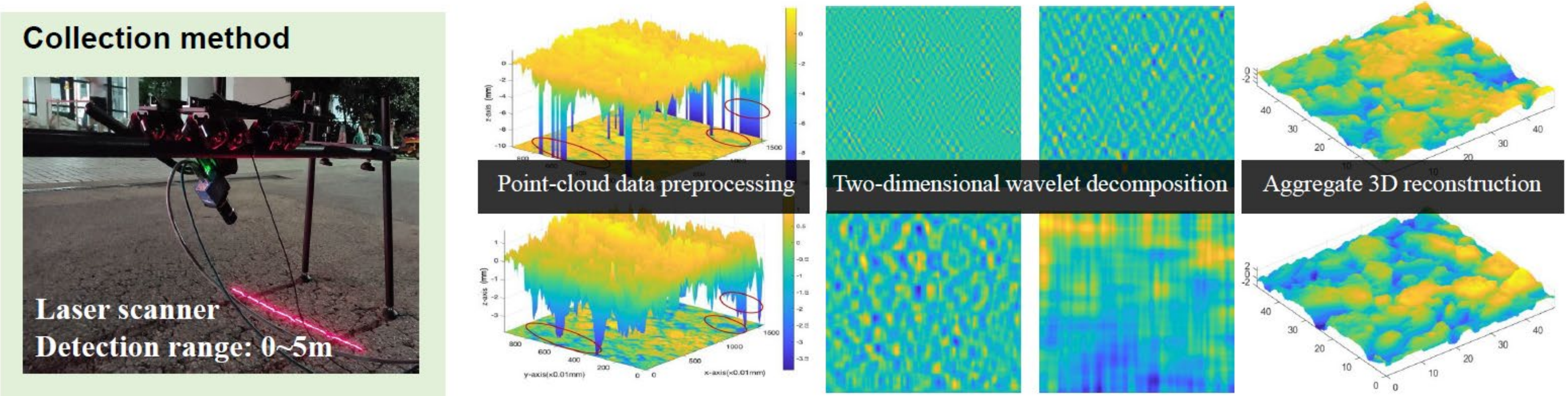






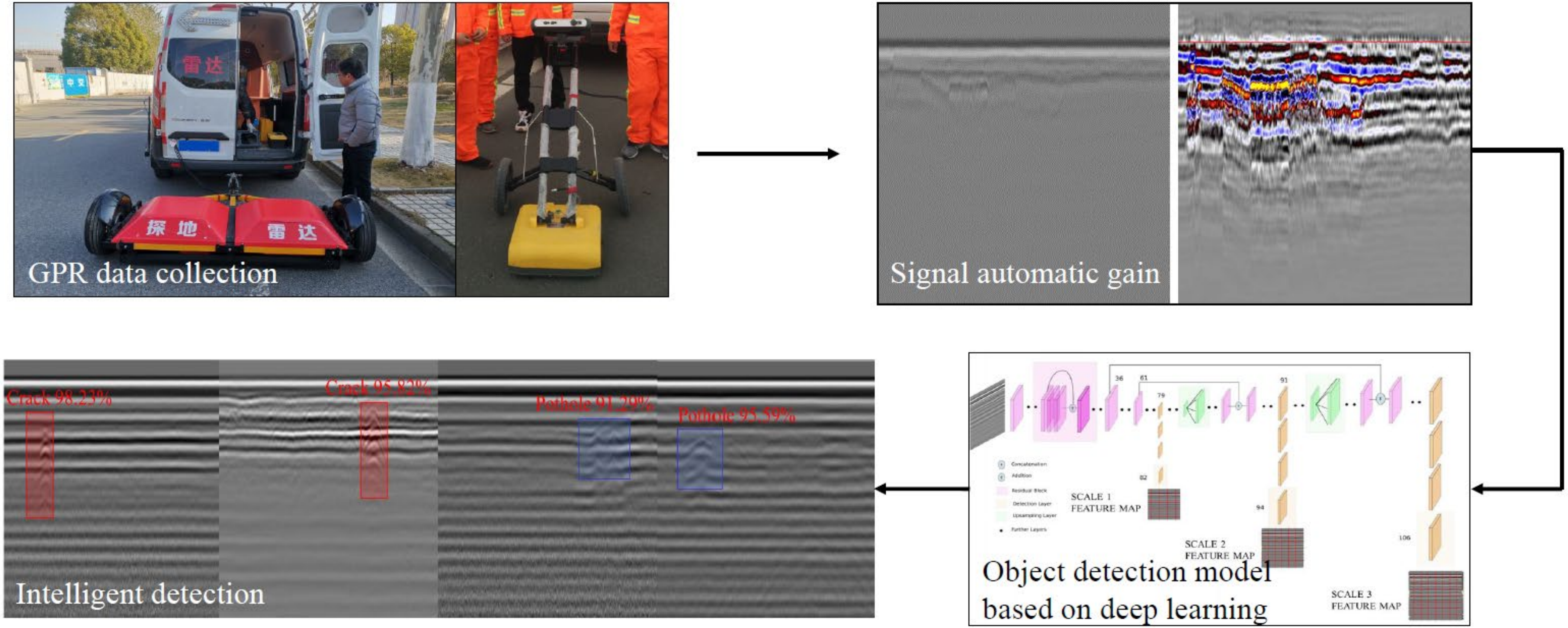
Source: SPIE digital library
Dynamics of Civil Structures, Volume 2
Polytec

Pavement laser measurements



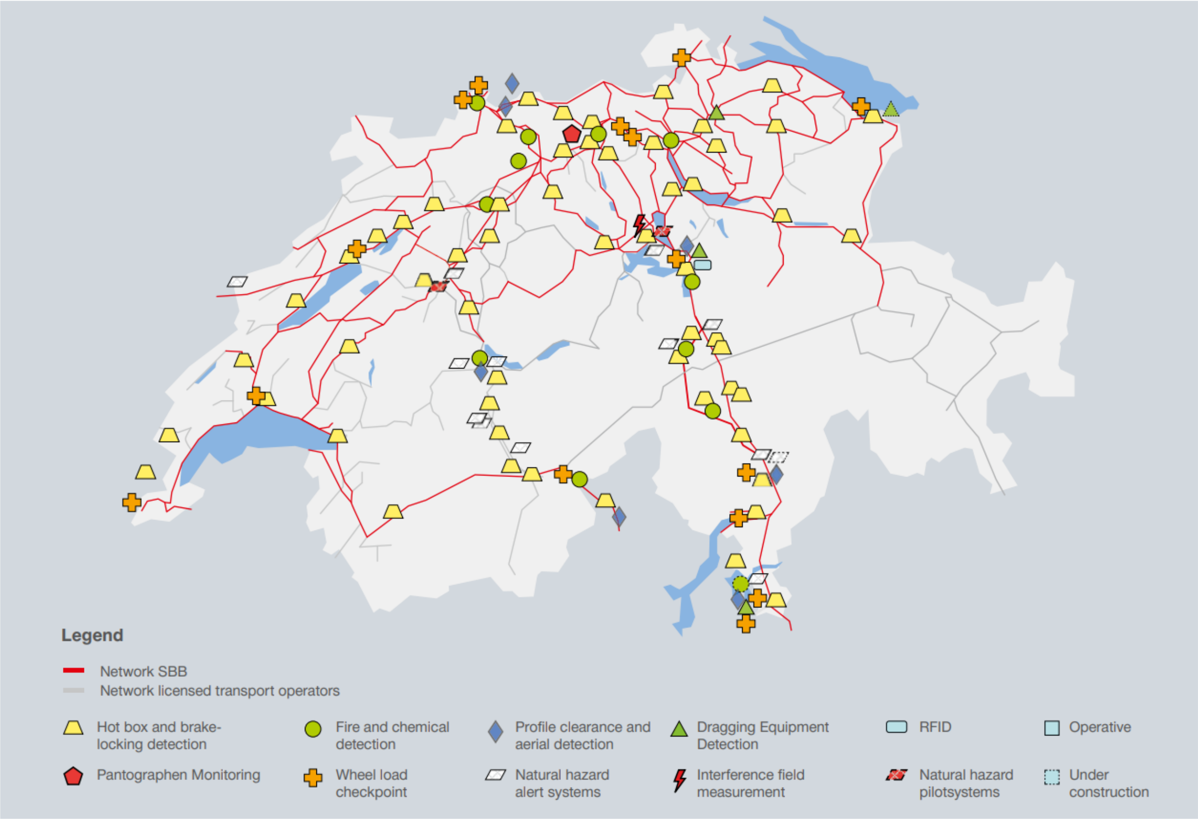
Source: Yishun Li

Detection of Subsurface Distress via Ground Penetrating Radar



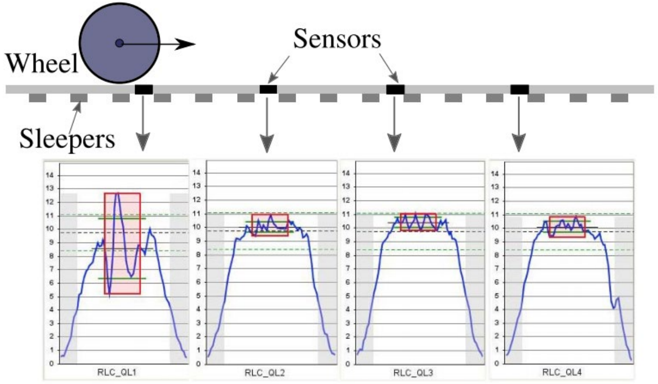
Source: Yishun Li

Wayside monitoring devices at SBB



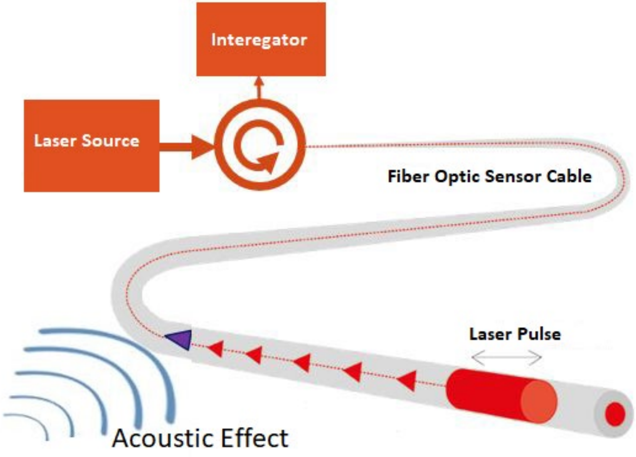
Source: SBB

Wheel load checkpoints +RFID



Accelerometers

(Distributed) Acoustic sensors



Source: GRAS

Maintenance reports / manuals / event recordings

Natural Language Processing (NLP) in maintenance

- **NLP** can process maintenance logs, manuals, and other unstructured text data.
- **Predictive Maintenance Insights**:
  - Analyzing unstructured text from maintenance logs and work orders to predict potential equipment failures.
  - Detecting patterns in historical records to forecast breakdowns.
- **Automated Reporting & Documentation**:
  - Automatically generating maintenance reports by summarizing complex work orders or logs using NLP.
  - NLP-based transcription for voice notes or technician inputs.
- **Fault Detection from Logs**:
  - Scanning equipment error messages and sensor logs to identify and classify common faults.
  - NLP can interpret error codes, anomaly reports, and operational data in natural language.
- **Knowledge Management**:
  - Extracting, organizing, and summarizing information from vast amounts of maintenance manuals, technical documentation, and historical repair records.
  - Providing quick searchability of solutions through conversational AI or chatbot assistance.
- **Chatbots and Virtual Assistants**:
  - NLP-powered chatbots to assist maintenance technicians with real-time troubleshooting, guiding them through steps or providing manual references.

Source: H. Canaday

PHM Process

Data Acquisition (DA)

↓

Data Manipulation (DM)

↓

State Detection (SD)

↓

Health Assessment (HA)

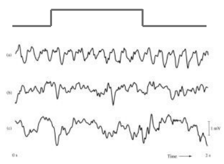
↓

Prognostics Assessment (PA)

↓

Advisory Generation (AG)

Data properties



Time dependency
- Time variant
- Time invariant (meta data such as asset ID, system configuration)

Sampling (time discretization)
- Transaction/event (push)
- Sensor (pull)
  - Evenly sampled
  - Unevenly sampled

Physics nature
- Temperature
- Pressure
- Current
- Voltage
- Speed
- Acceleration
- ...

Sample dimension (not counting time)
- Scalar
- Vector
- Matrix
- ...

Dynamics (relative to sampling)
- Stationary
- Cyclic (non periodic)
- Waveform (periodic)
- Stochastic (non cyclic)

Types of data
- Binary
- Nominal
- Ordinal
- Discrete
- Continuous (real number)

Source: Wang, 2012

Data Sampling

- Transaction/event (data are “pushed” by data originator)
  - Data records occur only at the specified event / transaction / time stamp
  - Data between the time stamps / events are undefined.
- Sensor (data are “pulled” from data originator)
  - Data samples are acquired only at the specified time stamp
  - Data between the time stamps are just not observed.
  - Sampling rate
    - Evenly sampled – controlled (e.g. 100 Hz)
    - Unevenly sampled - triggered

Source: Wang, 2012

Categorical vs. Numerical Data

- Data
  - Categorical Data
    - Nominal
    - Ordinal
  - Numerical Data
    - Discrete
    - Continuous

Examples:

- **Nominal:** Operating mode, Event code, asset ID
- **Ordinal:** Performance level; severity level; friction level (Low-Medium-High → ranked levels)
- **Discrete:** Number of occurring diagnostic events, number of occurring faults / interruptions
- **Continuous:** Temperature, pressure, acceleration (most sensors)

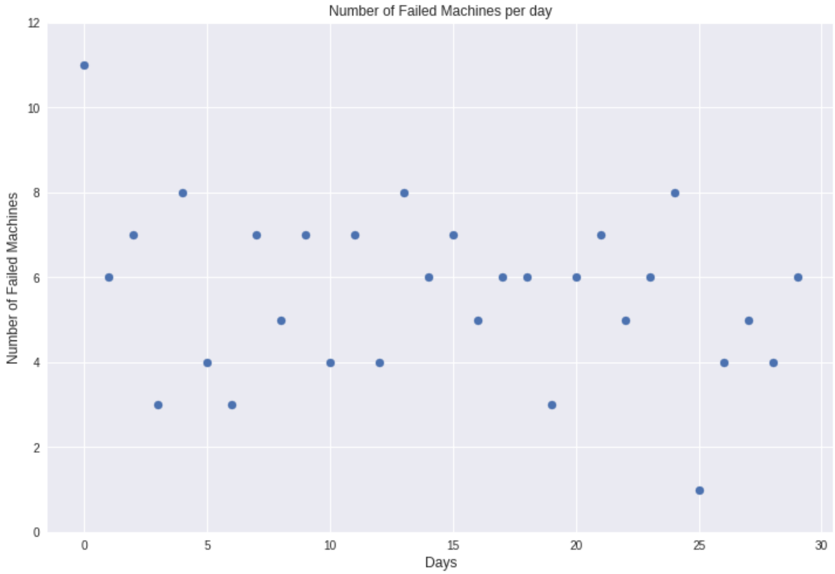
Binary data

- A type of categorical data in which there are only two categories
- Binary data can either be nominal or ordinal

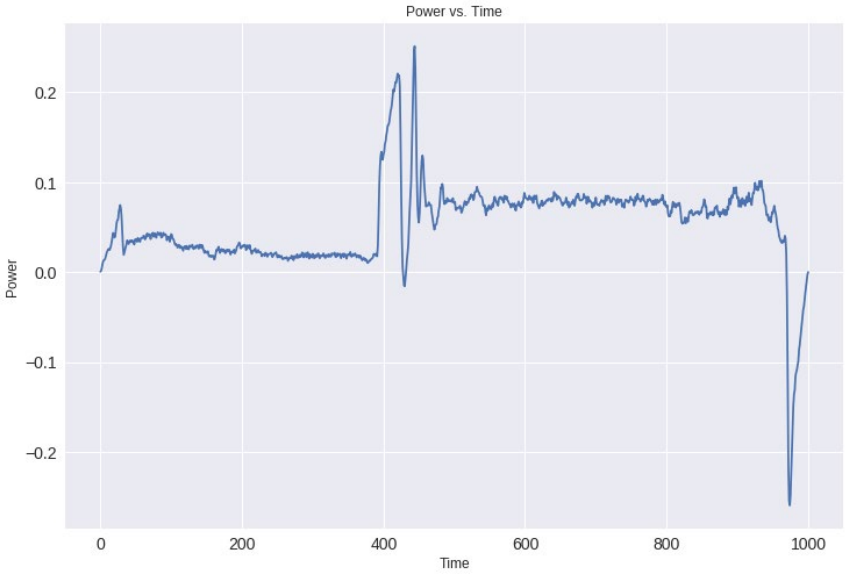
- Examples: event status, on/off sensor

Numerical Data: Discrete vs. Continuous

Discrete Data



Continuous Data

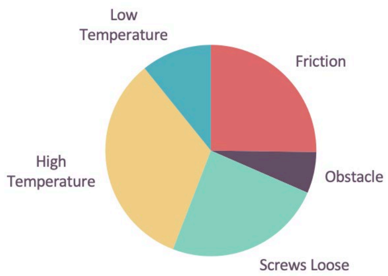


Categorical Data: Representations

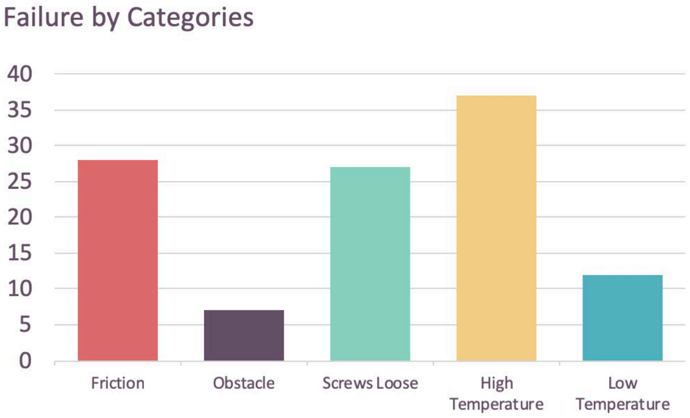
Frequency Tables

Cause of Error	*Number of Occurence*	*Percentage*
Friction	28	25.2%
Obstacle	7	6.3%
Screws Loose	27	24.3%
High Temperature	37	33.3%
Low Temperature	12	10.8%

Pie Charts



Bar Charts



Handling of categorical data

- Replacing values
  - Freely assign numbers to the categories according to the use case / expert knowledge
- Encoding labels
  - convert each categorical value in a column to a number between 0 and n_categories-1
- One-hot encoding
  - convert each category value into a new column and assign a 1 or 0 (True/False) value to the column
- Binary encoding
  - first the categories are encoded as ordinal, then those integers are converted into binary code, then the digits from that binary string are split into separate columns
- …

Example of one-hot encoding

Color
Red
Red
Yellow
Green
Yellow

Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1

Binary Encoding

Temperature
Hot
Cold
Very Hot
Warm
Hot
Warm
Warm
Hot
Hot
Cold

Order
1
2
3
4
1
4
4
1
1
2

Binary
001
010
011
100
001
100
100
001
001
010

Temperature_0	Temperature_1	Temperature_2
0	0	1
0	1	0
0	1	1
1	0	0
0	0	1
1	0	0
1	0	0
0	0	1
0	0	1
0	1	0

Handling of categorical data

- Some categorical indicators can be used to split the problem in sub-problems (e.g. indicator of the operating conditions for base and part load → developing two models for the two types of operating conditions)

Signal dynamics (relative to sampling)

Stationary (constant + white noise)

- Power, speed, temperature in steady state of, gas turbines, etc.

Stochastic (non-cyclic)

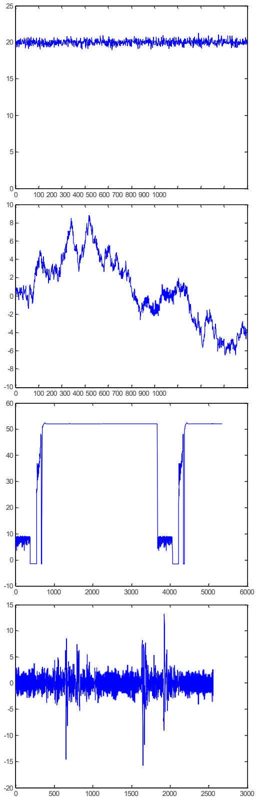
- Power, torque, speed

Cyclic (consider each period individually)

- Power, speed, pressure in manufacturing process, gas turbine startup, take-off of airplanes

Waveform (consider multiple periods together)

- Vibration sensors, acoustic sensors



Source: Wang, 2012

Why data pre-processing?

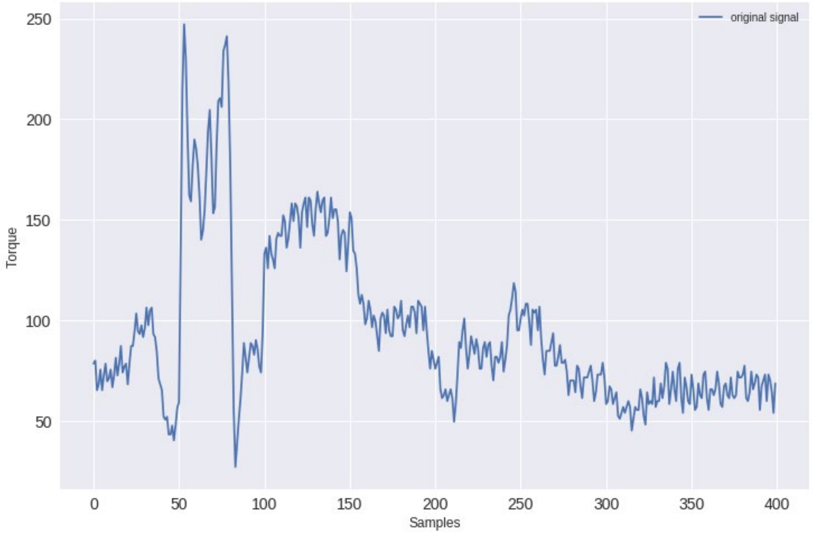
- Data in the real world is “dirty”
  - incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
  - noisy: containing errors or outliers
  - inconsistent: containing discrepancies in codes or names

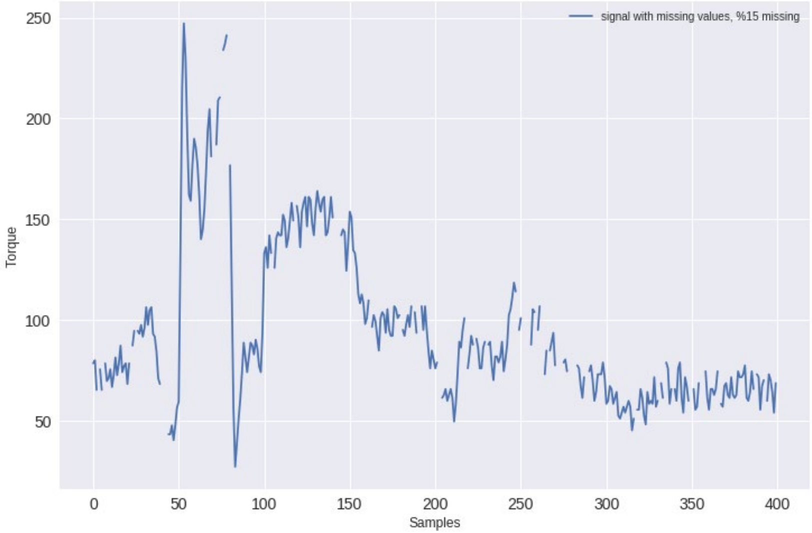
Main tasks in data pre-processing

- Data profiling
  - examining, analyzing and reviewing data
  - collect statistics about its quality.
- Data cleaning
  - Fill in missing values, smooth noisy data,
  - identify or remove outliers, and resolve inconsistencies
- Data integration (if needed)
  - Integration of multiple databases, data cubes, or files
- Data transformation
  - Normalization and aggregation
  - Structuring unstructured data
- Data reduction
  - Obtains reduced representation in volume but produces the same or similar analytical results
- Data discretization (if required)
- Data enrichment
  - Feature engineering
- Data validation
  - Assessing the dataset for quality assurance

Missing Data Problem

- Times series observed with 15% missing data





Missing, noisy, inconsistent data

- Missing data
  - Data imputation approaches (next slide)
- Noisy data
  - Binning
  - Filtering
  - Clustering
  - Remove manually
  - Apply denoising algorithms
- Inconsistent data
  - External references
  - Knowledge engineering tools

Data imputation

- Complete case analysis: Delete any record that has missing values from the data set.
- Nearest neighbors: to impute variable x average the value x of the k closest data points with no missing values.
- Average method: Average the value of x for the non-missing values.
- Hot deck: pick a “similar” record at random and use its value of x.
- Predictive: Fit a model to the data with variable x as the target and use it to predict the value (e.g. kernel regression)
- Single imputation: Draw a value at random from the conditional distribution of x given the other variables
- Multiple imputation: Repeatedly draw values at random from the conditional distribution of x given the other variables (e.g. as above), creating new data sets. Make the predictions with these now complete datasets and average the predictions.

Caution with the different imputation approaches

- Complete case analysis: can result in a bias
- Nearest neighbors: The definition of the “close points” and the value of k required
- Average method: Easy to implement but crude
- Hot deck: A definition of “similar” is required
- Predictive: Better suitable but understates the uncertainty in the imputation process.
- Single imputation: Better suitable, respects the uncertainty. However, just a single value is sampled.
- Multiple imputation: generally regarded as the best method (a sample is better than a single observation)

Data normalization / Standardization

- Feature scaling

(→ also referred to as min-max Nomalization)

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

- Standard score

(particularly suitable for normally distributed data)

(→ also referred to as Z-Score Standardization)

$$X' = \frac{X - \mu}{\sigma}$$

Alternatives (particularly for data with outliers)

- Quantile Transformation

- Power Transformation (non-linear transformation)

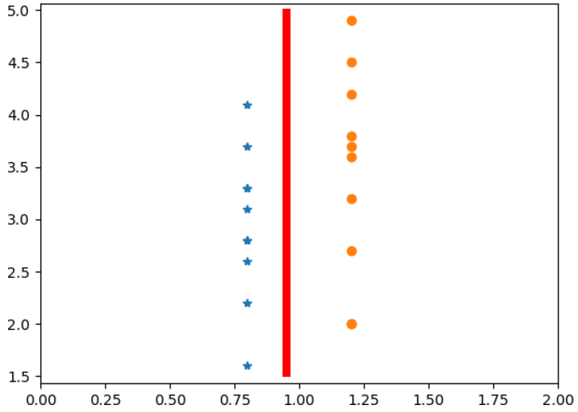
Data Discretization

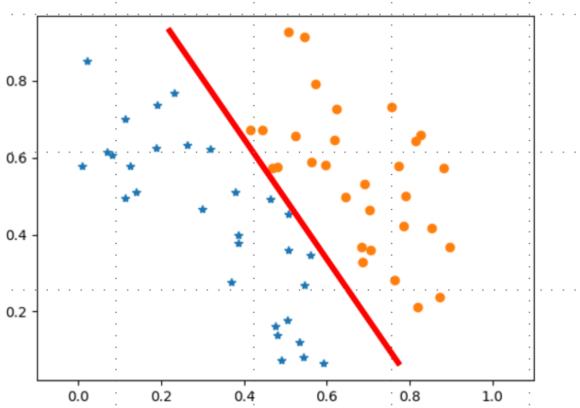
- Create bins (e.g. equal depth, equal size) → e.g. different categories of part-load conditions of a gas turbine
- Additional smoothing possible → replace the values in a bin by their mean

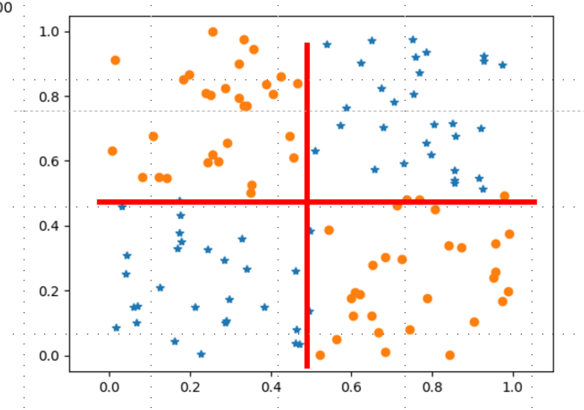
What are Features and Why do we need them?

- Transform raw signals into more informative signatures (or fingerprints) of a system
- Reduce size / complexity of the dataset
- Provide a physical description / representation
- Reduce resources necessary for further processing
- Achieve intended objectives
- Features are often to be the most crucial point for PHM applications

Univariate versus multivariate feature engineering







Example of feature engineering

Option 1: good feature engineering

Traditional Learning Algorithm

Input Data

↓

Feature Engineering

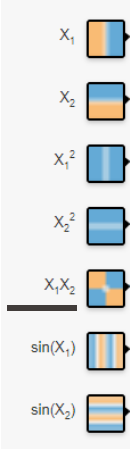
↓

$Y = g(X')$

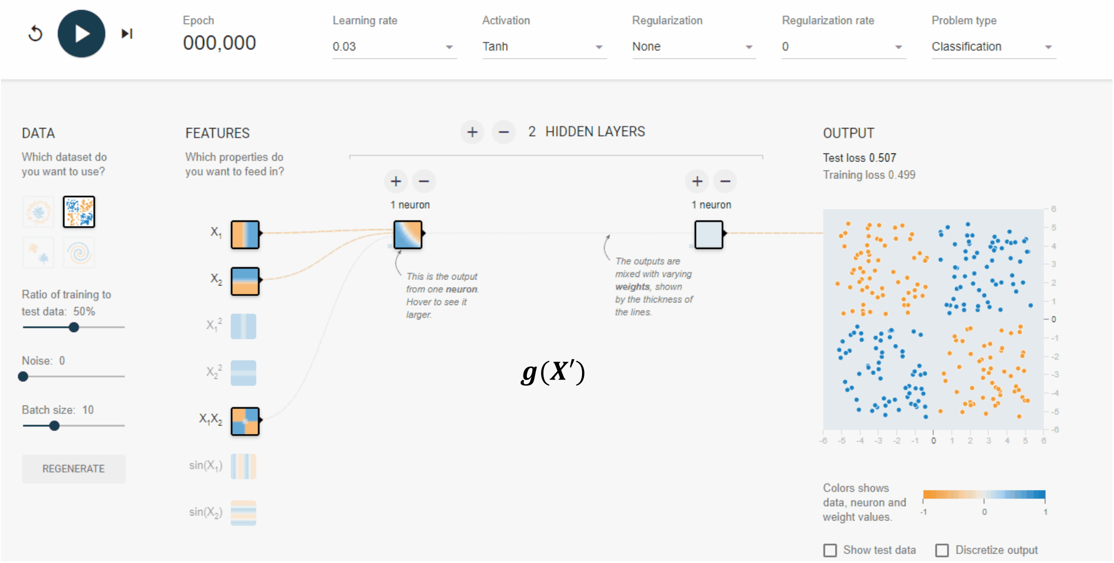
↓

Output model

Features



$\Rightarrow \quad X_1X_2$

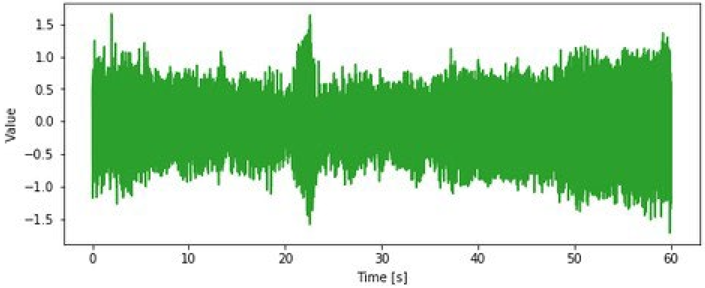


Example feature extraction in images

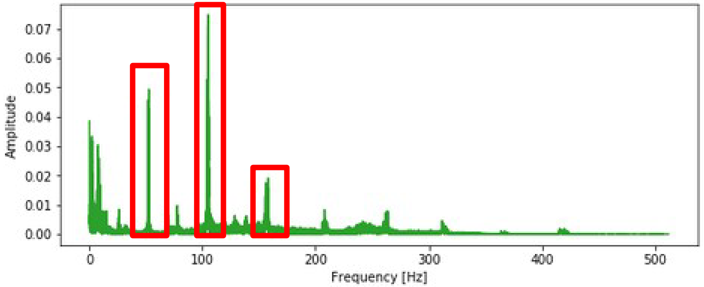
ML algorithm (e.g. classifier)

Prediction (label)

Example



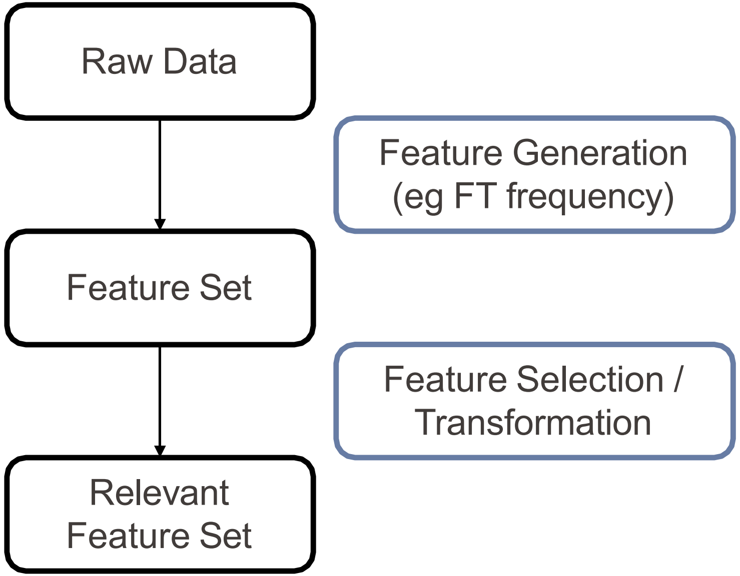
FT Transform



Features: Magnitude of 3 carefully chosen frequency bands

In the future, we can compare these features for any measurements

Feature Extraction Process:



- **Raw Data:** Can be of different type and nature and size
- **Generation:** exhaustive, ad-hoc,
  - prior knowledge (domain, physics), modify variable type (cat. to numeric)
- **Feature Set:** Various representation and
  - size (dimensionality change)
- **Dimensionality reduction:**
  - select best subset
  - transform in space of lower dimensions

What matters for the process

- Kind of data
→ available type of features

- Aim of the application and domain
→ (eg. monitoring, detecting, predicting)
→ (vibration, turbines, electrical,…)

- Kind of methods that will be used
→ To extract feature,
→ To use the features for the application

Different Technologies
- Statistical analysis
- Signal processing
- Image processing
- Time-series analysis
- Control theory
- Information theory

Different PHM applications
- Vibration analysis
- Turbine machines
- Electrical systems
- Electronic devices
- Batteries
- SHM

Different data types
- Continuous
- Categorical
- Binary
- ...

Time dependency
- Time independent (stationary)
- Time dependent (non-stationary)

Univariate vs. multivariate

Different data sampling rate

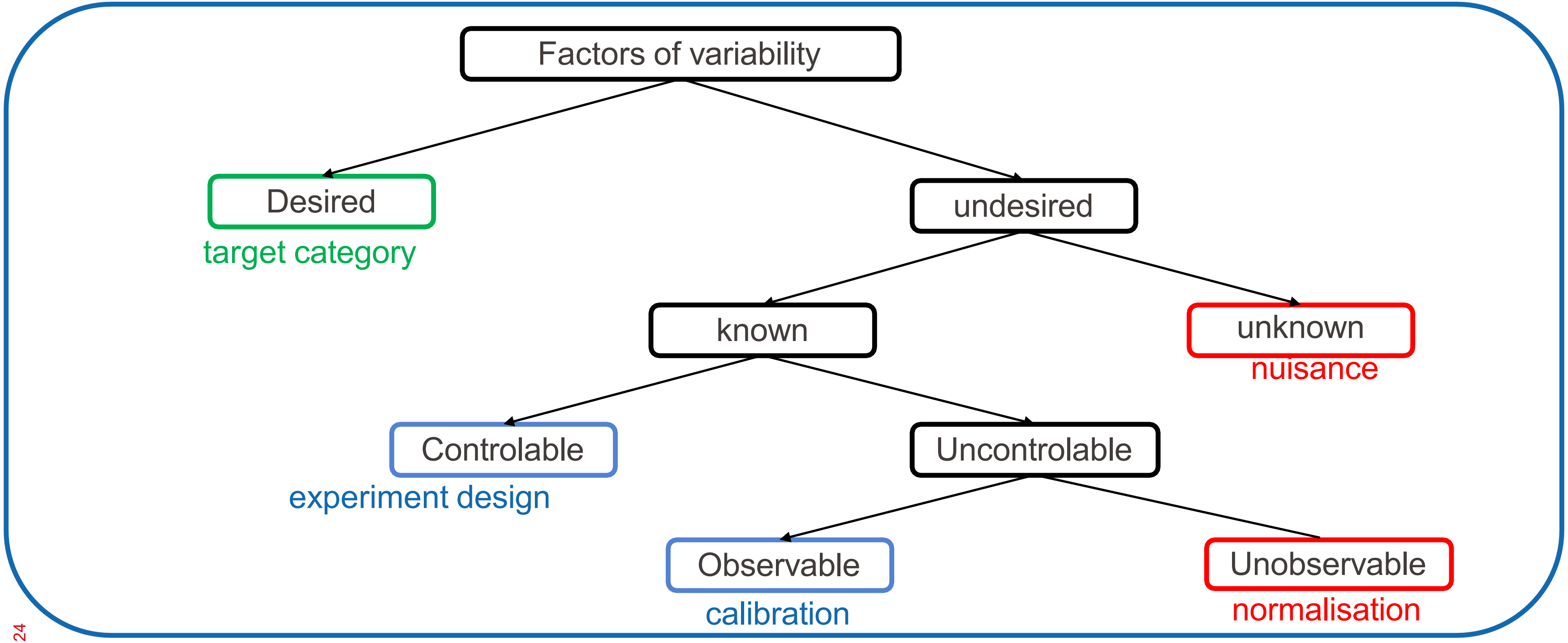
Aims and properties of features

- Explainability
- Parsimony
- Robustness (eg. to missing data)
- Uncertainty handling
- Link to knowledge and physics…

In fact, many features come from domain know-how

- Domain: Mechanical, structural, thermal, electrical,…
- Kind of system: vehicle, turbine, machinery, …
- Components gearbox, motor, pump, battery,…

Feature Variability



Feature extraction

- For fault detection:
  - Features should be dependent on the failure (and failure type)
- Feature tuned to operating conditions
  - feature relevance depends on the OC
- Feature designed based on knowledge:
  - Known relationships
  - known behavior of components

Some feature extraction approaches

- Descriptive statistical features:
  - For regularly sampled data: moments, correlation, RMS, …
  - For event based data: count, rate, duration, delay, …
- Descriptive models:
  - Distribution/histogram
  - Information based (mutual information)
  - Regression models, curve fitting
  - Classification, clustering (class label as a feature), sequence matching
- Mathematical Transformation:
  - derivative, cumulative sum, power, log, …. (eg. log if noise depends on amplitude, log normal)
  - Principal/Independent Component Analysis, etc…

Some examples of features

Power / Energy
- Root mean squared (RMS)
- Signal to noise ratio.
- Power Loss Efficiency (P_{out}/P_{in})

AC signatures
- Frequency
- Wavelength
- Crest Factor

Curve shape
- Curve length
- Curvature
- Eccentricity

$$\text{Crest Factor} = \frac{0.5 * (x_{max} - x_{min})}{RMS}$$

Magnitude
- Peak to peak
- Root mean squared (RMS)
- Peak to average ratio (PAR)

Statistical measures (moments)
- Mean
- Variance Skewness
- Kurtosis
- Hyperskewness
- Hyperflatness

Multi-variables
- Correlation
- Principal Components
- Distance to centroid (Silhouette, Mahalanobis)

Example: Feature extraction for vibration analysis

Stationary signals

Time domain

- Statistical-based
  - RMS
  - Variation
  - Skewness
  - Kurtosis
  - crest factor
- Model-based
  - AR model
  - HMM model
- Signal processing
  - TSA
  - Correlation
  - Convolution
  - Fractal analysis
  - Correlation dimension

Frequency domain

- Spectral analysis
- Envelope analysis
- Cepstrum analysis
- Higher order spectrum

Non-stationary signals

Time-frequency

- Short-time Fourier Transform (STFT)
- Wigner-Ville distribution (WVD)
- Empirical mode decomposition (EMD)
- Basis pursuit
- Spectral kurtosis
- Cyclostationary analysis

Wavelets

- Continuous wavelet transform (CWT)
- Discrete wavelet transform (DWT)
- Wavelet packet transform
- Morlet wavelet
- Hilbert-Huang transform

feature engineering is a research field

Yan, W. et al, 2008

Some feature extraction Approaches

domain know-how

- Physics based :
  - expected input-output relations, etc.
  - comparison to expected output (model)

- Special procedures for data processing:
  - operational regime segmentations, envelop analysis, etc…
  - Time synchronous averaging

Statistical Features: Moments

$$\mu_n = \int_{-\infty}^{\infty} (x-c)^n \, f(x)\,\mathrm{d}x = \int_{-\infty}^{\infty} (x-c)^n \,\mathrm{d}F(x)$$
$$= \mathrm{E}\left[(X-c)^n\right]$$

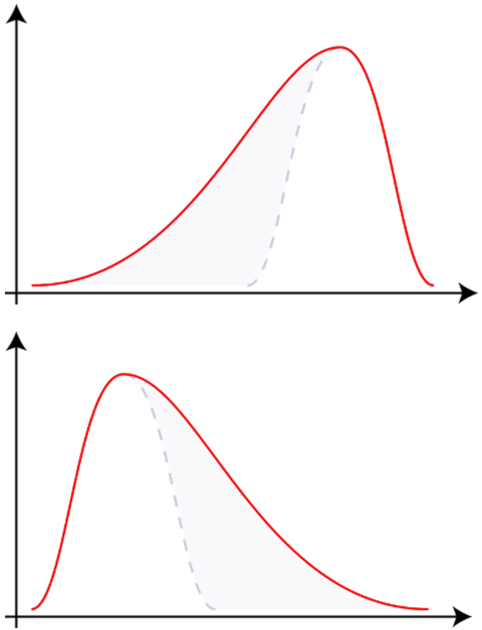
- c = 0 : raw moment
- c = average(X): central moment

- Normalized (central) moment:

$$\frac{\mu_n}{\sigma^n} = \frac{\mathrm{E}\left[(X-\mu)^n\right]}{\sigma^n}$$

n	**Raw Moment**	**Central M**	**Normalised M**
1	Mean	0	0
2		Variance	1
3			Skewness
4			kurtosis
5			Hyperskewness
6			Hypertailedness
7			

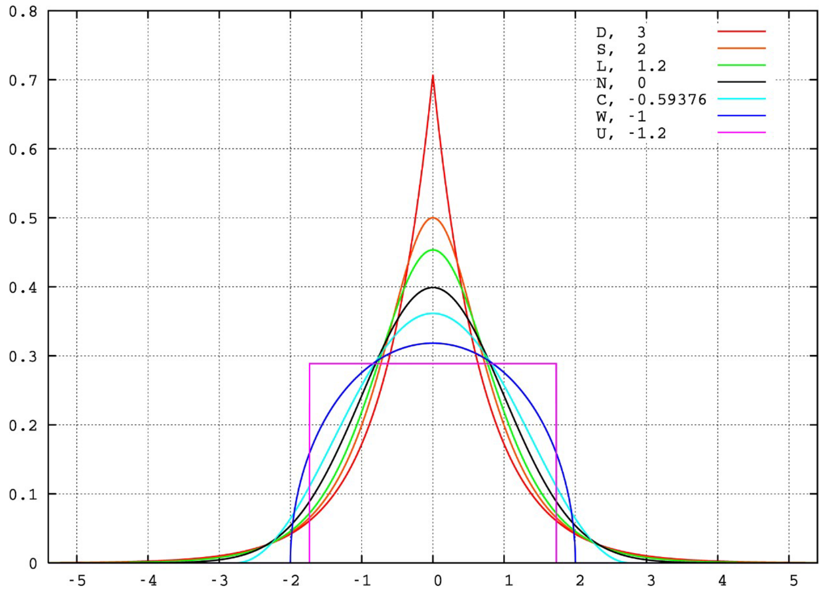
Skewness and Kurtosis



negative skew

positive skew

Kurtosis



Take away

- Procedure:
  - feature extraction + dimension reduction
- What to extract:
  - data property vs. application domain vs. algorithm requirements
- Feature extraction vs. signal processing
- Feature goodness:
  - Relevance and redundancy
- Feature selection:
  - wrapper approach vs. filter approach vs. embedding
- Feature consistency and sensitivity issues

From feature extraction to end-to-end / deep learning

ML algorithm (e.g. classifier)

Prediction (label)

End-to-end learning

Prediction (label)

Feature learning: Deep Learning

Option 2: feature learning

Deep Learning Algorithm

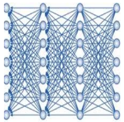
Input Data

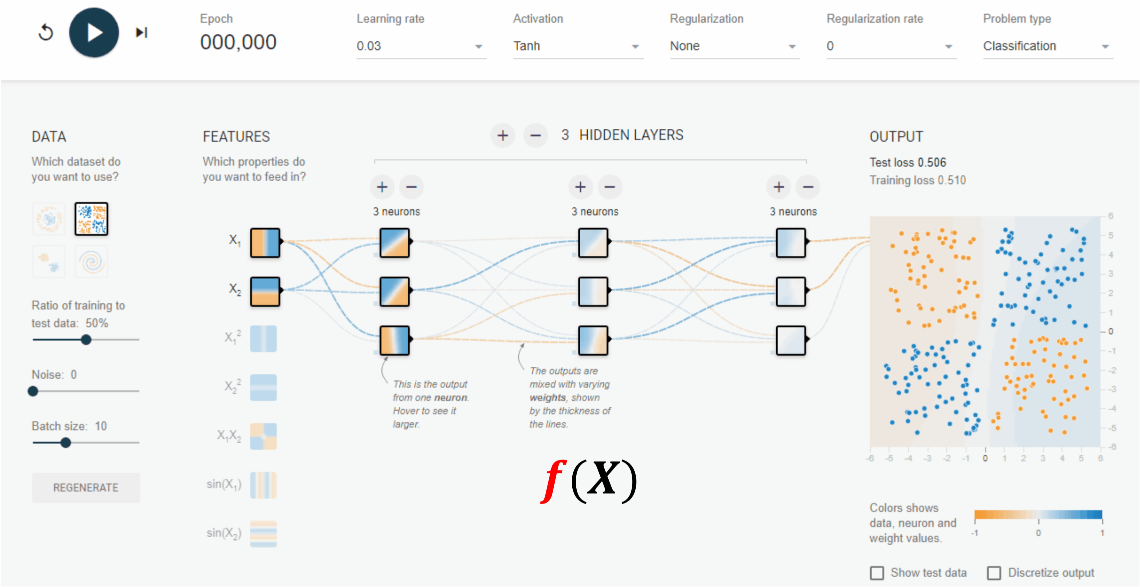
↓

$$\boldsymbol{Y} = \boldsymbol{f}(\boldsymbol{X})$$

↓

Output model





leading performance with abundant data

Feature engineering vs. Feature learning

Feature Engineering

Feature extraction

Feature dimensionality reduction
- Feature selection
- Feature low-dim. Projection

Feature Learning

Shallow feature learning
- Supervised
- Unsupervised

Deep feature learning
- Supervised
- Unsupervised

Knowedge based
Manuel, labor intensive
Domain/problem specific
Not sclable

Data driven
Automated
Generic
Scalable

Fault detection / diagnostics / prognostics

Detection

Healthy or faulty operation?

Diagnostics

Are there different faults types?

What is the root cause of the faults?

Prognostics

How long the system can continue operating i.e., remaining useful lifetime?