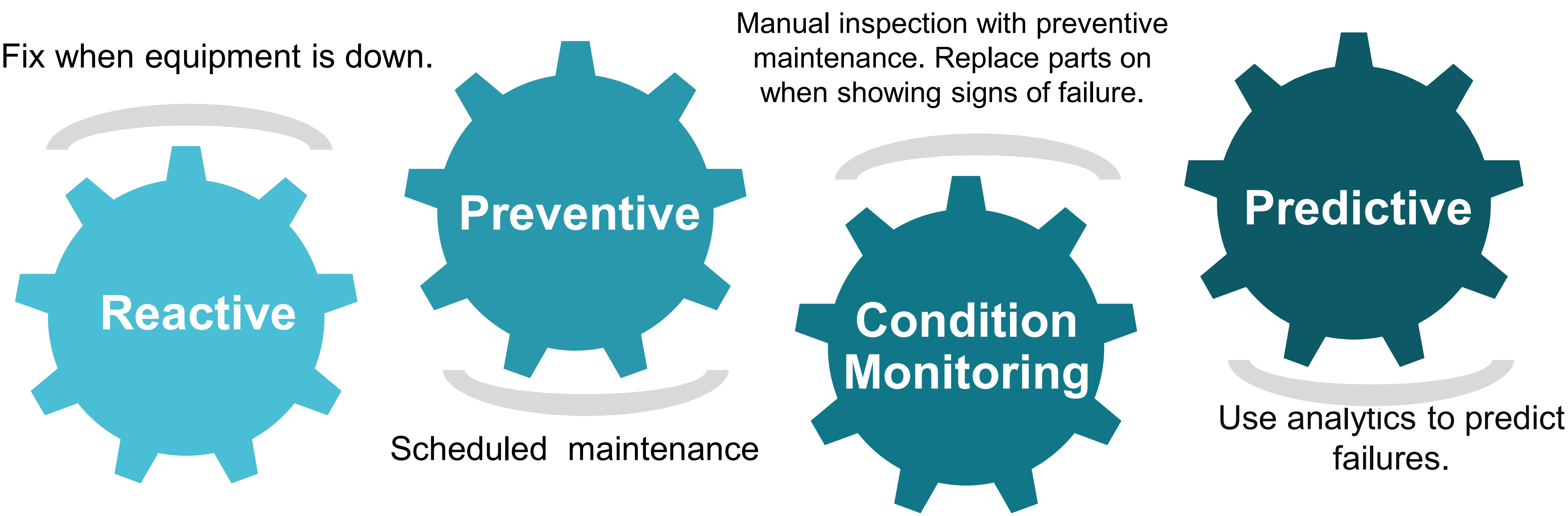
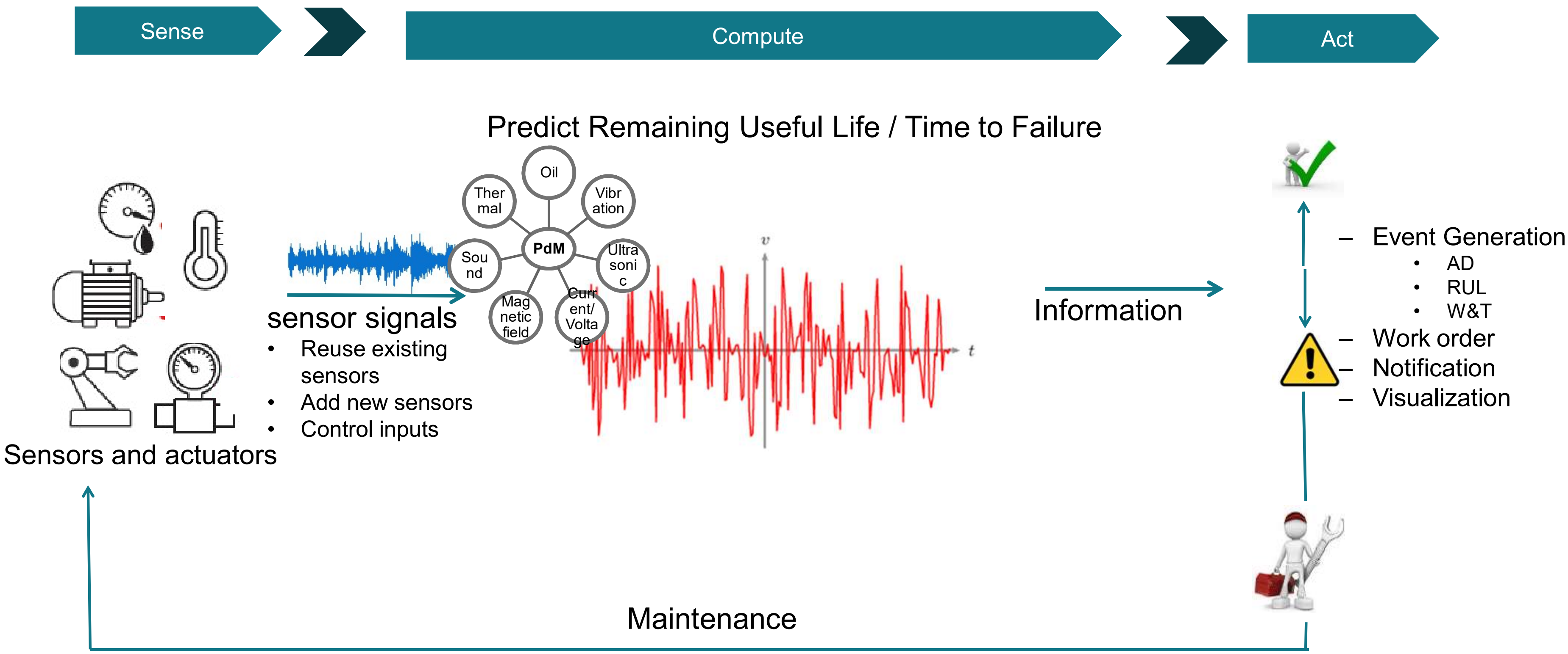
Using Deep Learning for Predictive Maintenance

Evolution of maintenance

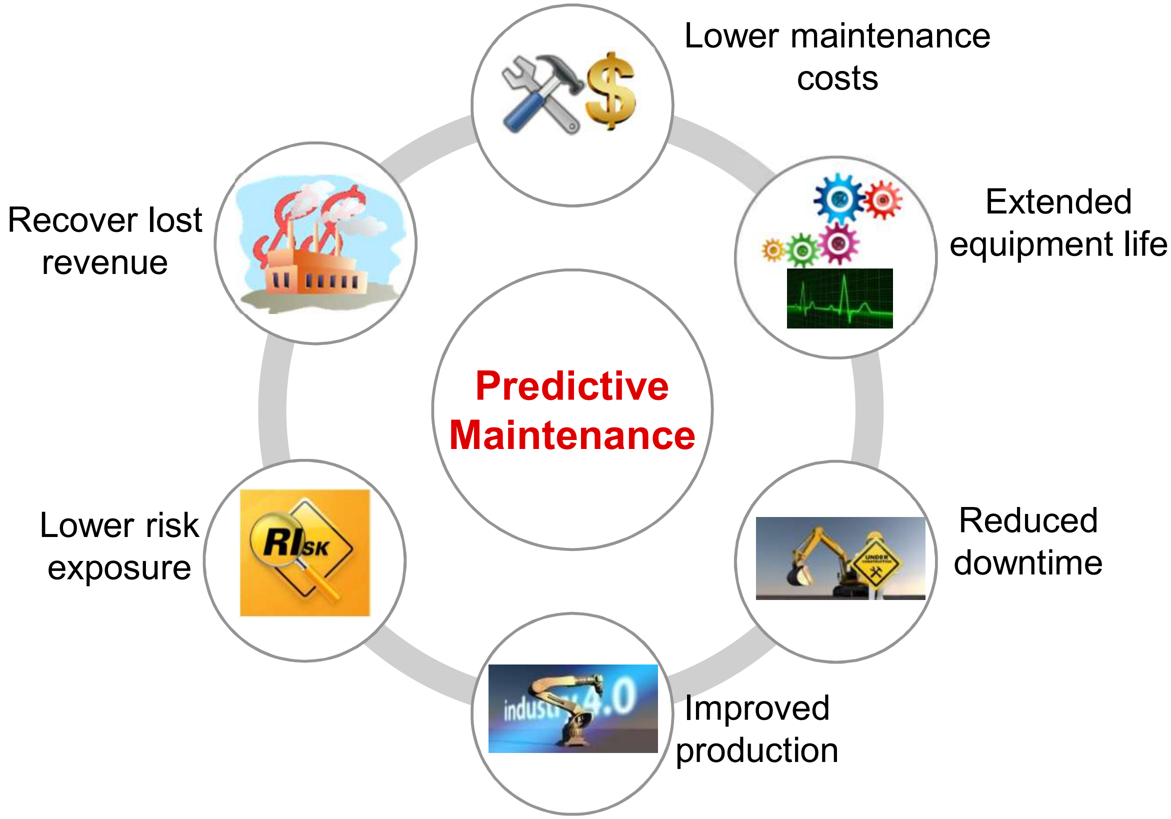


Industry 4.0 and predictive maintenance

Predictive maintenance employs advanced analytics on the machine data collected from end sensor nodes to draw meaning insight to predict machine failures.



Why predictive maintenance?



Examples from real-life scenarios

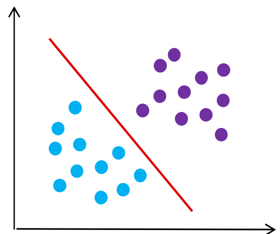
- Find defective bearings long before defects are visually seen.
- Find misalignment between two rotating pieces of equipment.
- Recognize when fans become unbalanced.
- Identify when bearings need lubrication.
- Report when an electrical connection needs to be tightened.
- Alert when oil is contaminated or in need of replacement.

Predictive maintenance problems

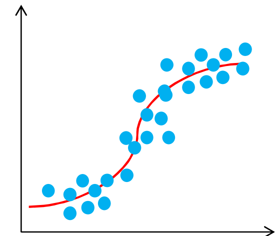
- Will this equipment fail in a given period of time (next 7 days, next 1 month, etc.): Yes or no?
- What is the Remaining Useful Life (RUL) or the Time to First Failure (TFF)?
- How to quantify wear and tear (of expandable components)?
  - Subset of RUL, focused on shorter-leaving subsystems
- Is there an anomaly in equipment behavior?
  - Further analysis may provide ***failure classification.***
- How best to optimize equipment settings?

Predictive maintenance approaches

- Problem definition: Classification or regression approach
  - Classification: Will it fail?
    Multi-class classification: Will it fail for reason X?
  - Regression: After how long will it fail?



Classification



Regression

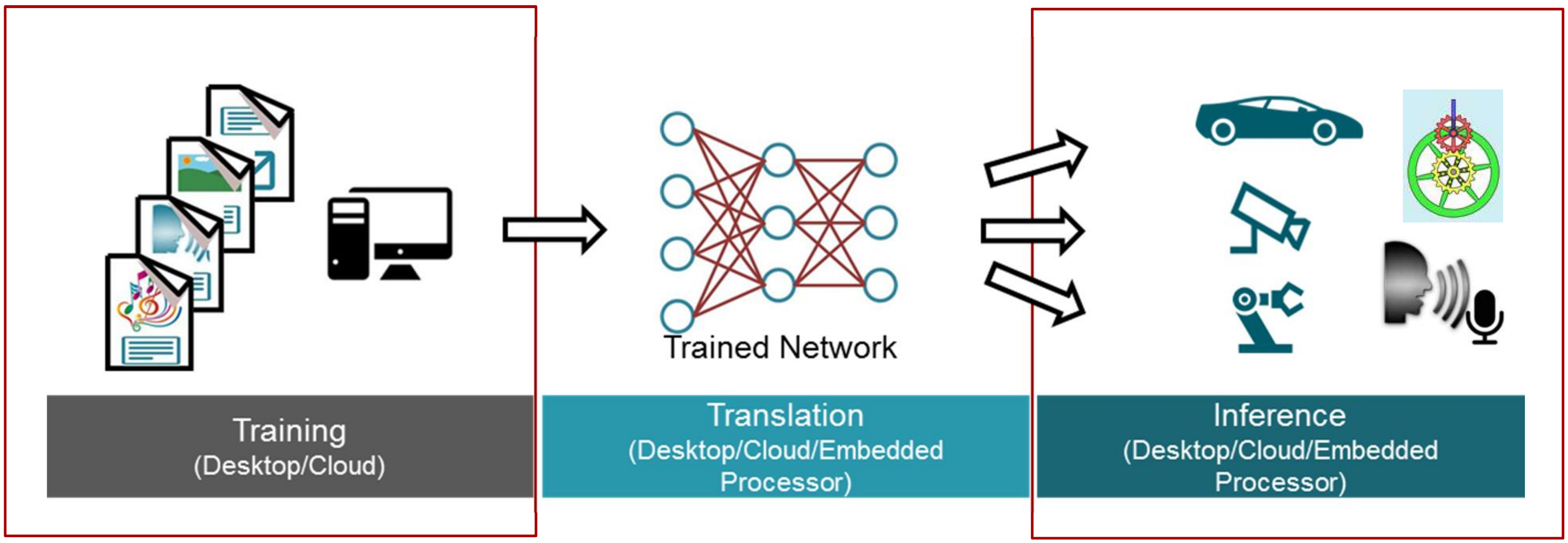
- Methods:
  - Traditional machine learning:
    - Decision trees: Random forests, gradient boosting trees, isolation forest
    - SVM (Support Vector Machines)
  - Deep learning approach:
    - CNN (Convolution Neural Network)
    - RNN (Recurrent Neural Network)/LGTM (Long Short Term Memory)/GRU (Gated Recurrent Unit)
  - Hybrid of deep learning and Physics-Based Modeling (PBM):
    - Use PBM to generate training data where lacking
    - Use PBM to reduce the problem space (feature engineering)
    - Use PBM to inform and validate DL models (e.g., to identify catastrophic failures, most notably in scenarios with low amounts of training data and a high degree of mission criticality)

When to use deep learning to predict failures

- Is accuracy of prediction more important than interpretability?
- Are there frequent changes in asset configurations / operational conditions?
- Are traditional approaches costly?
- Do you have access to a dataset that covers all kinds of events needed to discriminate?
  - Access to ample baseline data:
    - Dataset is relevant to “normal” asset behavior, the more the better
    - Big enough to ensure good statistics
      - For RUL/TTF scenarios, lots of baseline data eventually leading to failure
  - Access to failure history
  - Labeled, for supervised learning only
  - Up-to-date so that it covers any new events or behaviors
  - Access to maintenance and repair history

Deep learning algorithm development flow

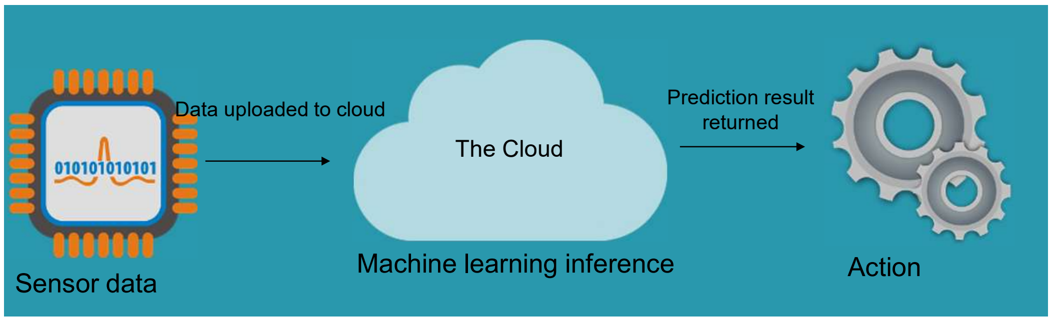
Two-step process: Training and inference



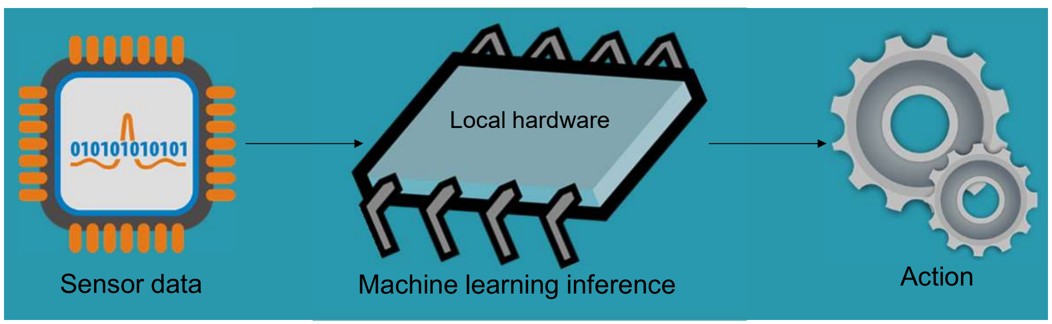
Comparing training approaches for deep learning-based predictive maintenance models

- Deep learning model training can be done offline, online or as a hybrid approach:
  - Offline training: Training is done on the static dataset.
  - Online training: Training is done as the data comes in.
- Offline training approach:
  - Complete dataset, fully determining system behavior is available
  - Similar deployment environment
  - Approximately static behavior
  - Same across all product instances (all product features perfectly aligned – no tuning required)
- Online training approach:
  - Datasets are not available and connectivity is either not enabled or narrow band:
    - System model identification is done at the edge, and prediction deviation used as outlier indicator.
    - Anomalies / faults are detected as outliers using unsupervised learning.
- Hybrid training approach:
  - Create initial model using offline training.
  - Adapt (transfer learning) using online training at the edge to take care of environment differences, and/or individual setup differences.

Deep learning inference: Cloud versus edge



Inference on the cloud



Inference at the edge

Cloud vs edge processing

- Data transmission cost
- Network bandwidth
- Network latency
- Network connectivity
- Security
- Reliability
- System power

For more information

- Sitara Processors Product Overview: http://www.ti.com/sitara
- Introduction to Deep Learning: https://training.ti.com/introduction-deep-learning
- WEBINAR: Why predictive maintenance is fundamental in Industry 4.0: https://training.ti.com/webinar-why-predictive-maintenance-fundamental-industry-40-0
- Texas Instruments Deep Learning (TIDL) Overview: https://training.ti.com/texas-instruments-deep-learning-tidl-overview
- Predictive maintenance of smart meters: https://training.ti.com/predictive-maintenance-smart-meters
- Predictive maintenance with robust IEPE vibration sensing over IO-Link interface: https://training.ti.com/predictive-maintenance-robust-iepe-vibration-sensing-over-io-link-interface
- For questions about this training, refer to the E2E Community Forums for Sitara Processors at http://e2e.ti.com