# Replay-Driven Continual Learning for the Industrial Internet of Things

CAIN 2023

Erik Johannes Husom
SINTEF Digital, Oslo, Norway

# Replay-Driven Continual Learning for the Industrial Internet of Things


Authors:

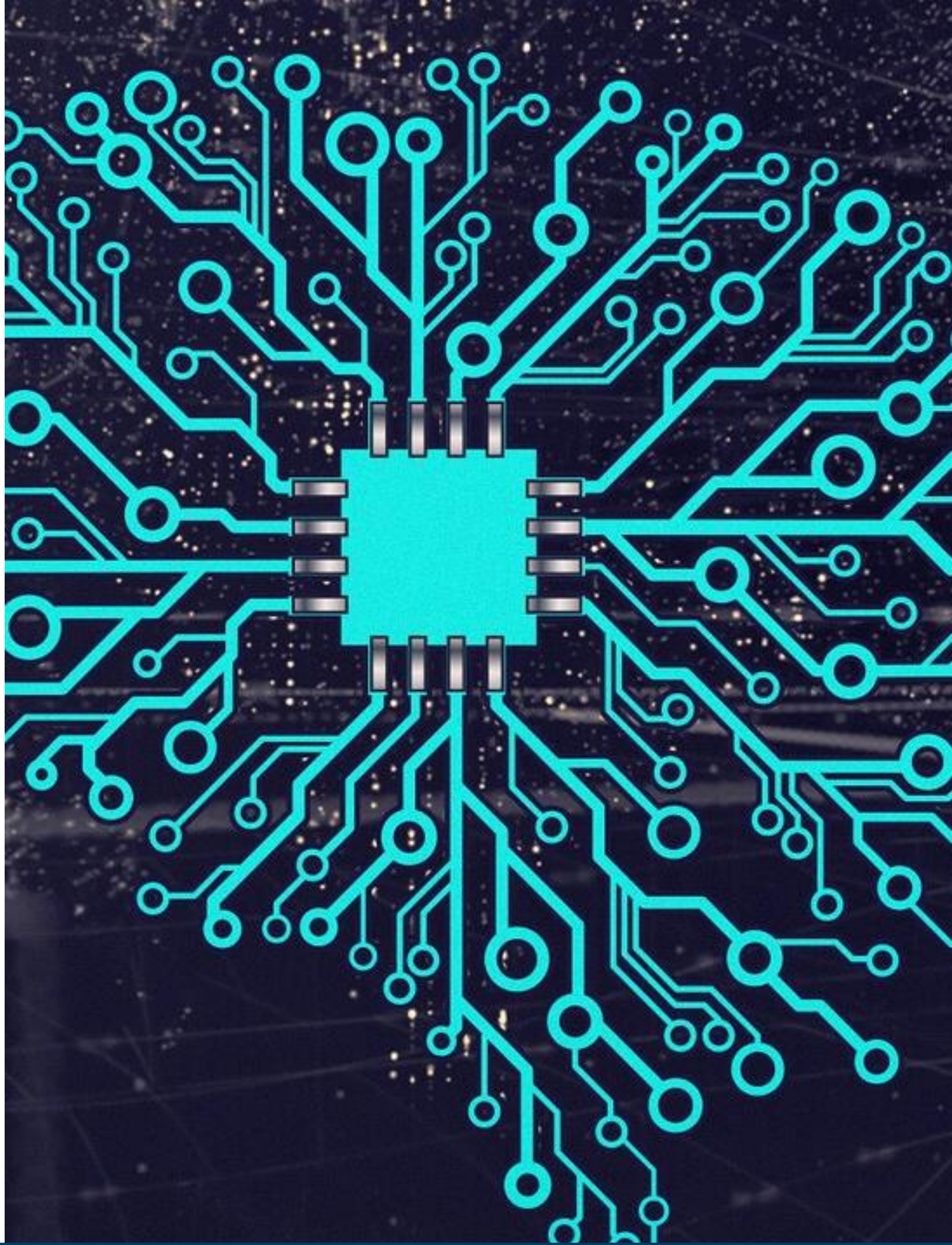
- Sagar Sen, SINTEF Digital, Oslo, Norway
- Simon Myklebust Nielsen, University of Oslo, Oslo, Norway
- Erik Johannes Husom, SINTEF Digital, Oslo, Norway
- Arda Goknil, SINTEF Digital, Oslo, Norway
- Simeon Tverdal, SINTEF Digital, Oslo, Norway
- Leonardo Sastoque Pinilla, University of the Basque Country, Zamudio, Spain

# Industrial Internet of Things (IIoT)

- Interconnection of sensors, devices, and industrial machines
- Streaming access to data from sensors for ML applications
- Challenges of evolving data in dynamic IIoT environments

# Continual learning – background and challenges

- Humans can continually learn without forgetting the past
- Artificial neural networks can catastrophically forget if trained only on new data
- Replay as a mechanism to consolidate short-term memory to long-term memory

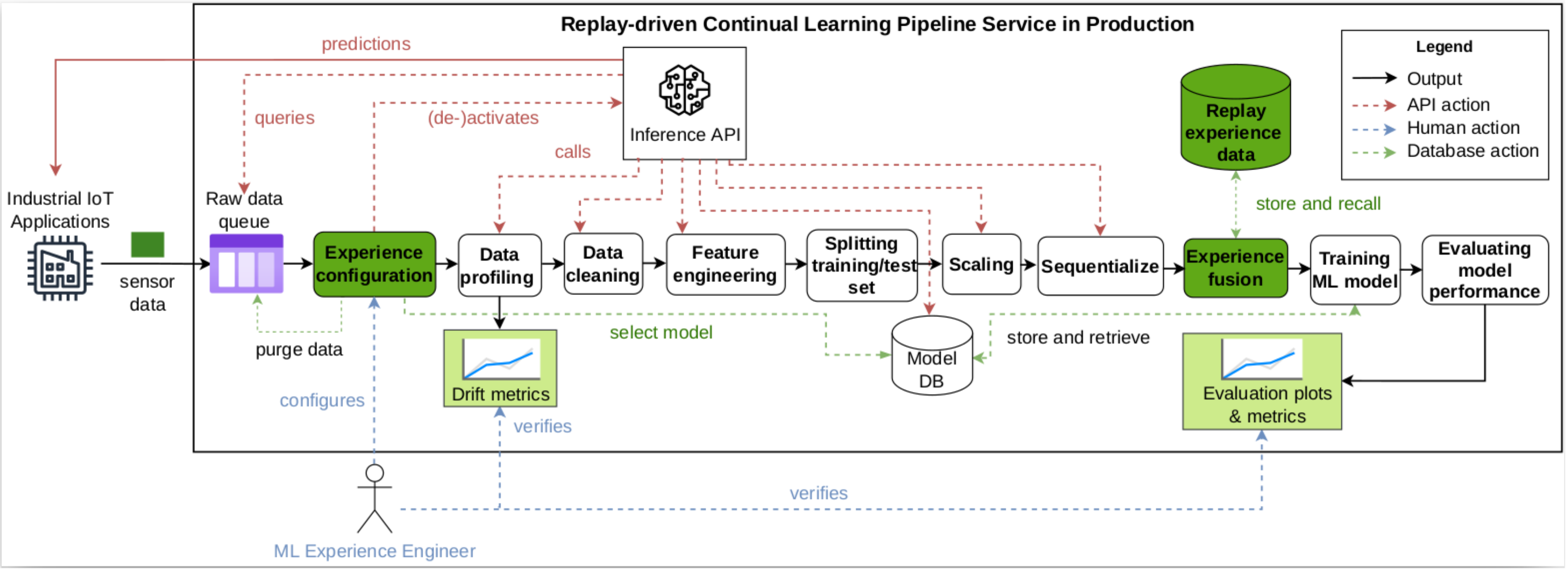


# Contributions: Replay-Driven Continual Learning Pipeline

- Novel Replay-Driven Continual Learning Pipeline
- New Tool Support (open-source ML pipeline)
- Evaluation on Industrial Case Studies

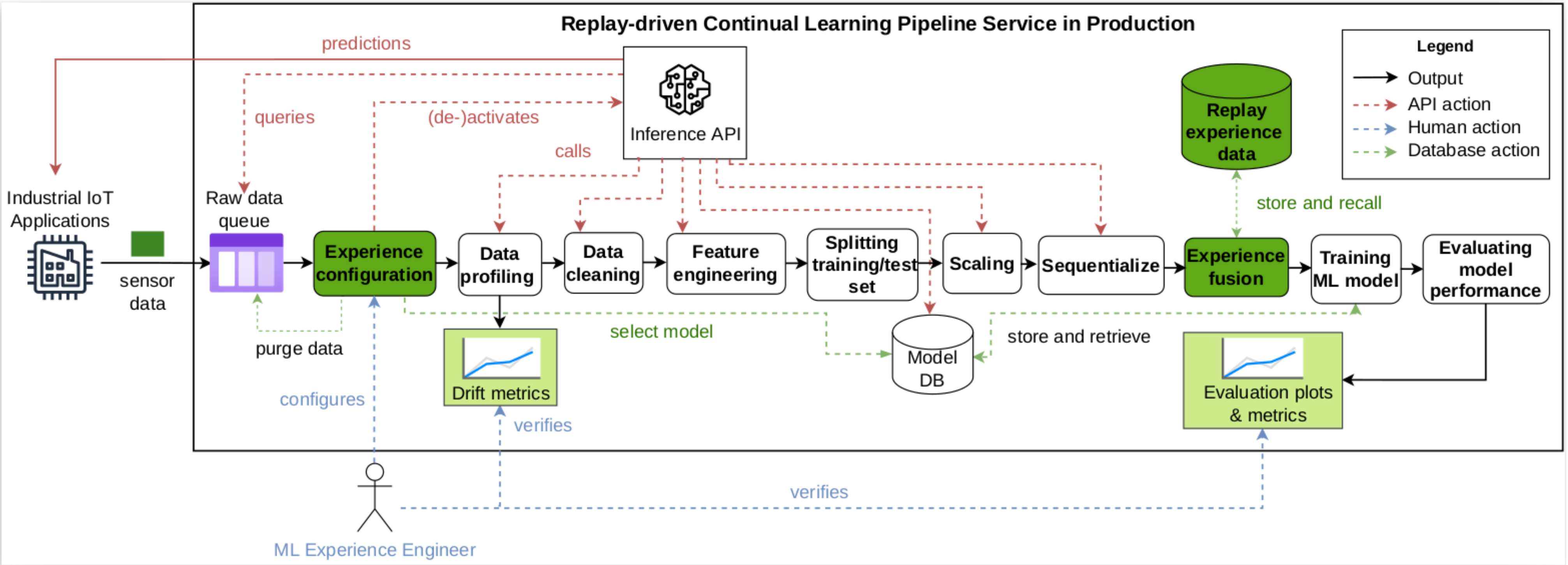
# Continual Learning Pipeline Overview

- Inspired by human learning experiences: learning, replaying, and inferring
- Six stages in the pipeline:
  1. Experience configuration
  2. Data pre-processing
  3. Experience fusion
  4. Training ML model
  5. Evaluating model performance
  6. Inference service (optional)



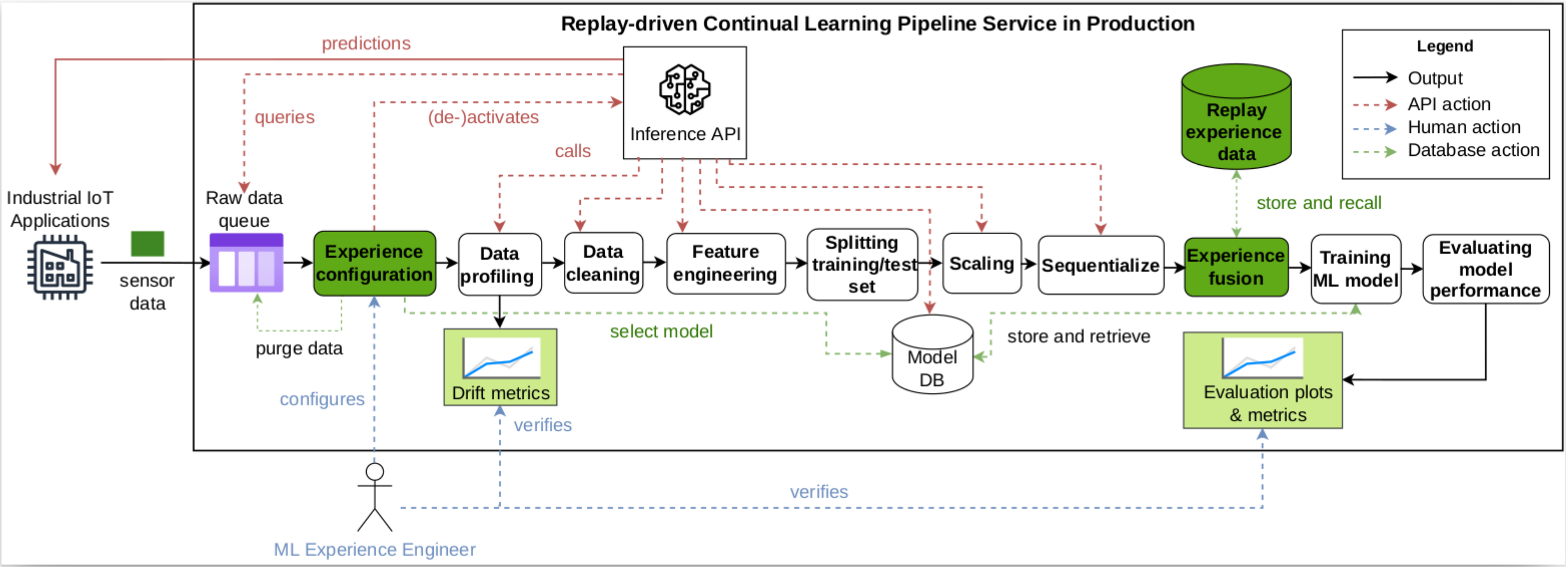
# Experience Configuration

- ML experience engineer configures experiences:
  - Baseline learning, inference, and replay-driven learning
  - Experience parameters: type, features, train/test split, replay data, ML model, and evaluation



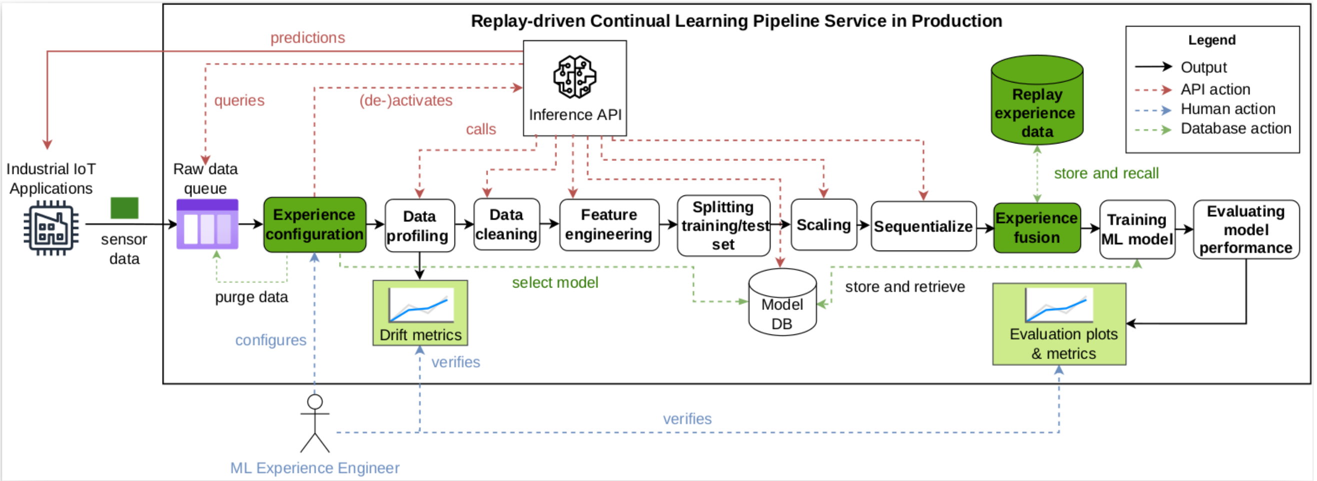
# Data Pre-processing

- Data pre-processing:
  - Raw data queue (MQTT, Apache Kafka)
  - Data profiling, cleaning, feature engineering, splitting, scaling, and sequentializing



# Experience Fusion, Model Training, and Evaluation

- Experience fusion: Combine new data with past experiences
  - Store and recall replay experience data to prevent catastrophic forgetting
- Training ML model:
  - DNNs/FCNNs, CNNs, and LSTM for training
  - Validation dataset to prevent overfitting
  - Model saved in Model DB
- Evaluating model performance:
  - Metrics: MSE, R2 score, MAPE, accuracy, and F1-score
  - Test dataset for unbiased evaluation



# Baseline Learning Experience

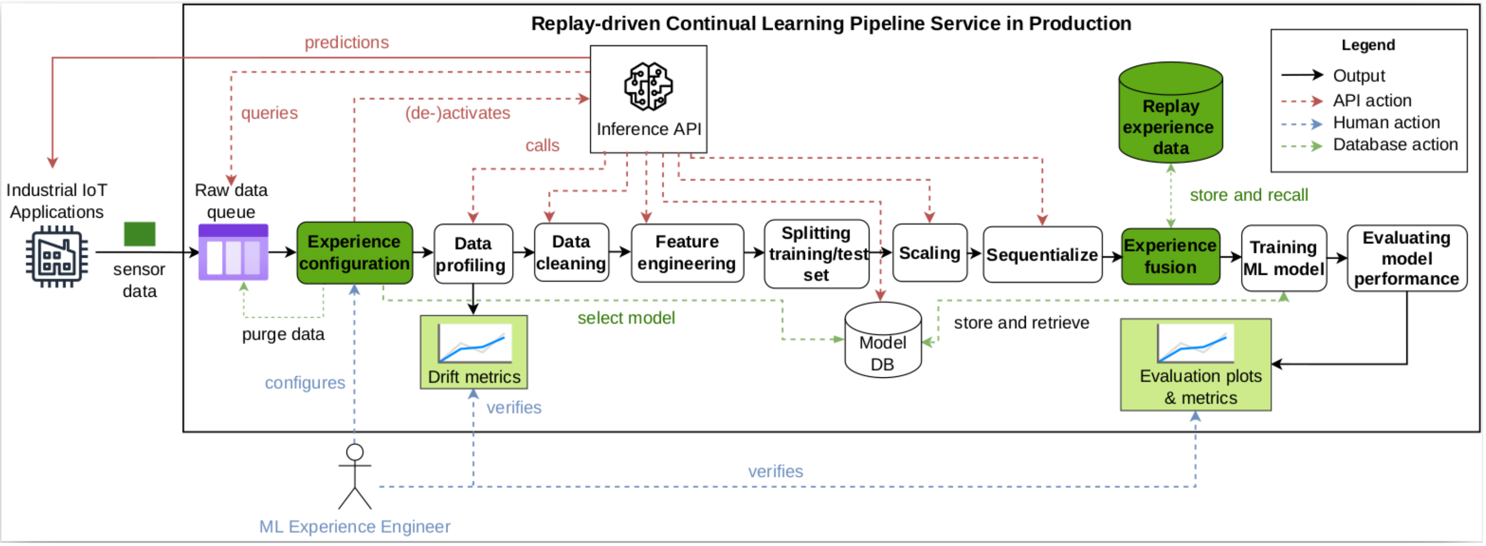
- First step in the pipeline
- Train ML model with static dataset
- Leverage correlations between input and target variables
- Fine-tune control parameters for acceptable performance
- Store trained model in Model DB

# Inference Experience

- Utilize existing ML model for predictions
- Runs model against a batch of new data
- Pre-processes data for inference
- Inference occurs when learning experiences are not happening
- Model remains stable under current circumstances

# Replay-Driven Learning Experience

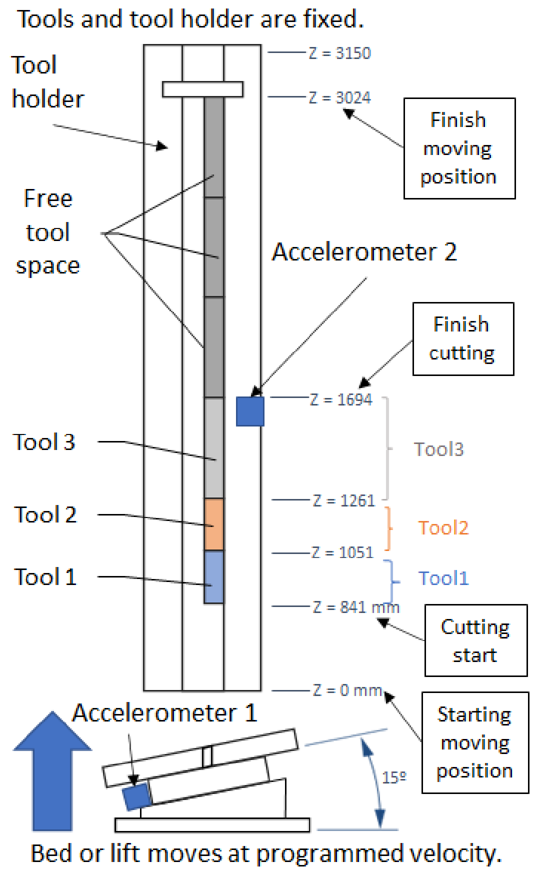
- Improve past ML models using new incoming data
- Monitor inference process for data drift and poor performance
- Combine new and past data for training
- Evaluate performance of new ML model
- Invoke inference experience with new model if satisfactory



# Evaluation: Subjects of the Evaluation

Three industrial case studies:

1. Bosch CNC Machining Dataset
   - Real-world industrial vibration data from milling CNC machines
   - Task: classify signals as normal or anomalous
2. Broaching of Airplane Turbine Discs
   - Data from broaching tools cutting slots on jet engine turbine discs
   - Task: predict tool wear in millimeters
3. Piston Rod Manufacturing Dataset
   - Data from the machining process of piston rods at CiP learning factory
   - Task: predict the remaining useful lifetime of the tool




Example setup from the broaching case study

# Evaluation: RQ1 - Baseline Model Performance

**RQ1: How does a baseline model based on static legacy data perform on new IIoT data for predictive inference?**

Performance measures: F1-score (classification) and $R^2$ score (regression)

The data from all use cases were split into five datasets in chronological order

**Bosch CNC Machining:**

- F1-score of 0.891 (1st dataset), significantly lower on other datasets (0.489 to 0.025)
- Performance deteriorates over time

**Broaching of Airplane Turbine Discs:**

- \- $R^2$ score lower than 0 on test data from the same dataset
- \- Performance degrades quickly in each subsequent dataset

**Piston Rod Manufacturing:**

- $R^2$ score of 0.539 (1st dataset), becomes negative on the 2nd dataset
- Performance quickly deteriorates

**Conclusion:** Results indicate the need for updating the models through continual learning

# Evaluation: RQ2 - Replay-driven Continual Learning Performance

RQ2: How does replay-driven continual learning perform on new and old IIoT data for predictive inference?

- Replay: Fusion of a certain percentage of past data with new IIoT data to train existing ML models
- Evaluated on Bosch CNC dataset, Broaching dataset, and Piston Rod dataset
- Tested replay percentages: 0%, 20%, 60%, and 100% (100% is equivalent to traditional ML)

# Evaluation: Performance of Replay-driven Continual Learning on Use Cases

SINTEF

**Bosch CNC Machining:**

- \- All degrees of replay work well in predicting anomalies in new data
- \- 0% replay: Model performance declines significantly with fewer anomalies
- \- 20%-100% replay: Prevents catastrophic forgetting and maintains model performance

**Broaching of Airplane Turbine Discs:**

- \- 0% replay: Steady deterioration in model performance
- \- 20%-100% replay: Improved performance on past data, but poor performance on new data
- \- Tool wear process understanding needed for better performance

**Piston Rod Manufacturing:**

- \- 0% replay: Poor performance on past data, better on new data
- \- 20%+ replay: Significant improvement on past data, high performance on new data (except for Piston Rod Set 5)
- \- Data imbalance affects performance

# Evaluation: Conclusion for RQ2

- 20% replay is adequate to maintain good performance on continual learning, minimizing catastrophic forgetting
- Both continual learning and traditional ML with 100% replay perform poorly on new unforeseen data that is likely out-of-distribution
- Careful selection of data for 20% replay is important to maintain the balance between performance on new and old data

# Evaluation: RQ3 - Engineering Continual Learning in Industrial Settings

RQ3: How can our approach be run in industrial production environments?

Lessons learned in two different industrial settings

Key Challenges & Solutions:

1. Data versioning and dependency management
   - Custom data and model management
2. Deployment infrastructure
   - Virtualized Docker container, shared model repository
3. Quality attributes
   - Data profiling, domain-specific assertions
4. Monitoring and logging
   - Purge policies, Bayesian neural networks for uncertainty estimation
5. Integration of models and components
   - Standalone continual learning pipeline, coupling learning and inference

# Discussion: Actionable Guidelines and Selection of Replay Data

Actionable Guidelines for Engineers:

- Configure pipeline to create a baseline model
- Monitor performance metrics on new and unforeseen data
- Assess training on new data and pros/cons of forgetting past knowledge

Selection of Replay Data:

- Realistic scenario: random selection of 20% of old training data
- Alternative data selection techniques: maintaining class balance and value range

# Conclusion and Future Work

- Presented a replay-driven continual learning pipeline
- Assessed pipeline with three IIoT case studies
- Performance maintenance by replaying 20% of past data
- Discussed AI engineering dimensions

Future Work:

- New continual learning metrics
- Continual balancing techniques
- Uncertainty estimation
- Green AI and Continual Learning

Technology for a better society