Accenture Applied Intelligence

Deep learning for predictive maintenance of rolling bearings

FUNDAMENTAL PRINCIPLES OF DATA SCIENCE MASTER'S THESIS

Laia Domingo Colomer

Advisor: Jordi Vitrià Marca

Accenture Advisor: Jesús Gabaldón

July 9th 2020

Universitat de Barcelona

UNIVERSITAT DE BARCELONA

CASE INTRODUCTION

INTRODUCTION

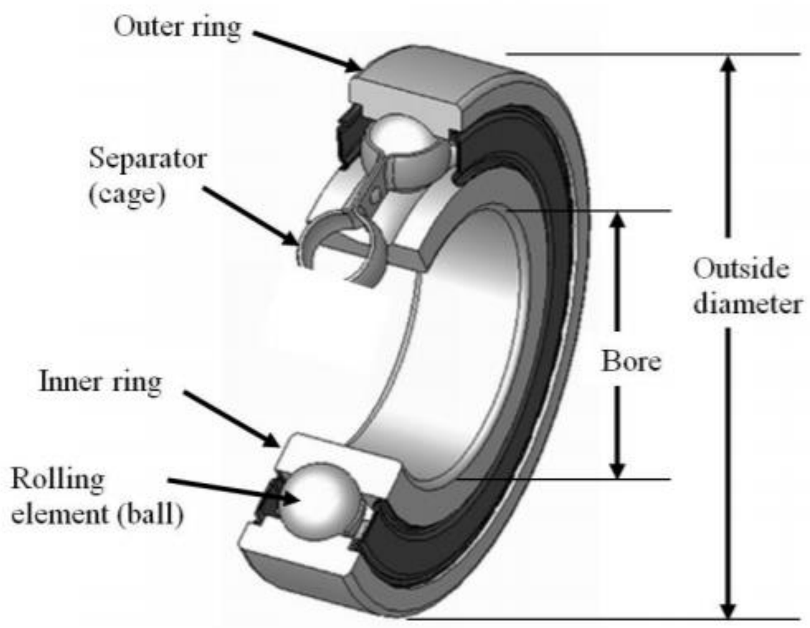
Rotating equipment (and specially pumps) in the facilities of a major oil & gas company were suffering from high failure rates.

DIGITAL MIGRATION JOURNEY

Failure risk detections manual process

to

Condition based monitoring using **sensors and an analytical model** to predict failure and its root cause for pumps



PROBLEM DESCRIPTION

PHYSICAL MODEL

Requirements:

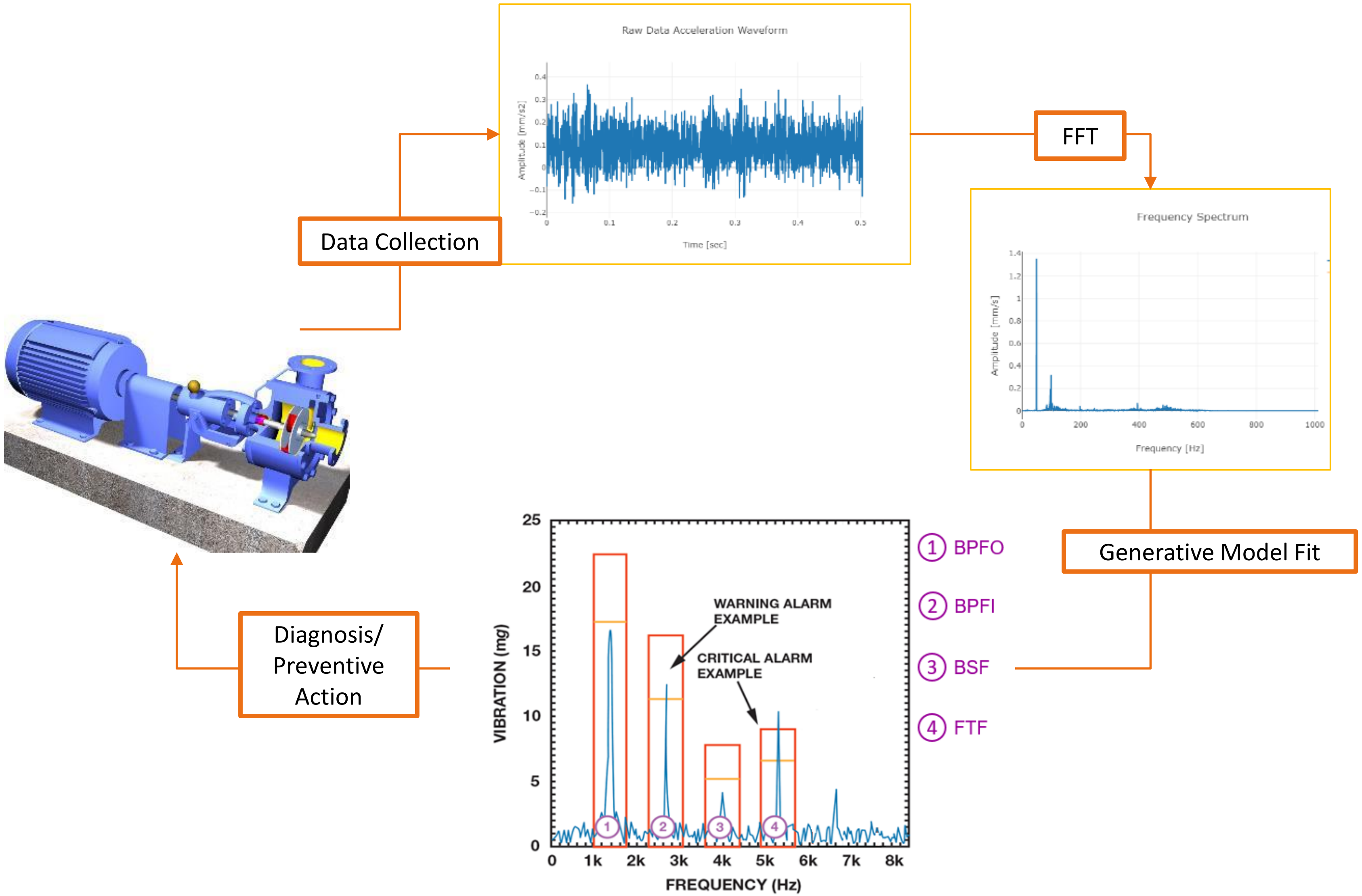
- Rotational speed
- Characteristic coefficients

Pros:

- Very accurate
- No need of training data

Cons:

- Expert knowledge required to develop the model
- **Rotational speed and/or bearings coeff not always available**



PROBLEM REQUIREMENTS

Goal

Design a model to classify the health state of a rolling element bearing.

PROBLEM REQUIREMENTS

Goal

Design a model to classify the health state of a rolling element bearing.

Requirements

- The model must thus be **robust to different rotating velocities**.
- The **bearing failure coefficients** are supposed to be **unknown.**
- The model must **detect failure modes at early stage.**
- The model must be able to work with **small training data sets**.

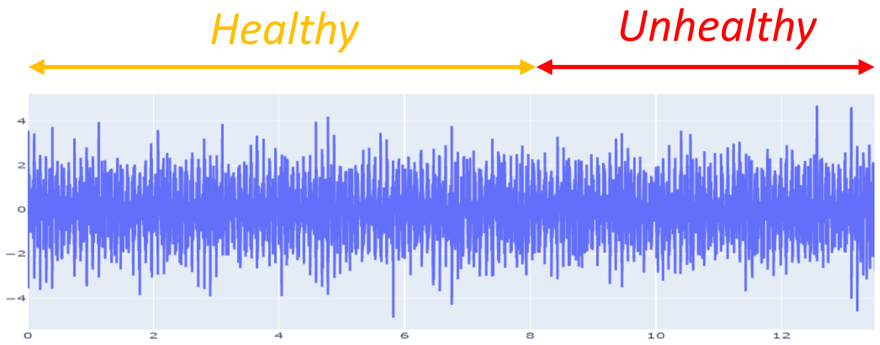
DATA STRUCTURE

We use the vibration data, called *waveforms,* as the input for our models.

We use 2 data sets **to simulate different operating conditions.**

IMS Data set

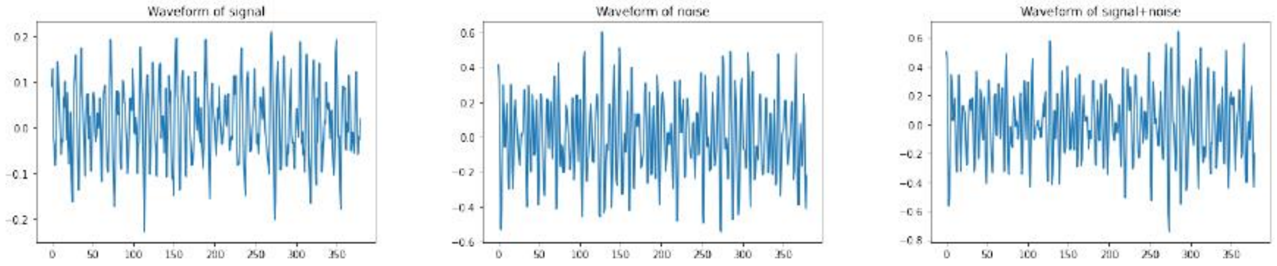
- Consists of three test-to-failure experiments.
- Contains the complete evolution of the failures.



	number of files	Duration	Damage occurred
Test 1	2156	49680 min	Bearing 3: Inner ring Bearing 4: Rolling element
Test 2	984	9840	Bearing 1: outer ring
Test 3	4448	44480	Bearing 3: outer ring

Paderborn Data set

- The failure is completely developed in all the experiment
- Contains data with different rotation speeds.
- We add noise to make the data more realistic



Nº	Rotational speed [rpm]	Torque [Nm]	Radial force [N]	Name
0	1500	0.7	1000	N15_M07_F10
1	900	0.7	1000	N09_M07_F10
2	1500	0.1	1000	N15_M01_F10
23	1500	0.7	400	N15_M07_F04

PROJECT STRUCTURE

The problem will be split into three subproblems:

- **Early detection of the failure**
  - ⇒ Detect the failure at its early development ⇒ 1D convolutional autoencoder

PROJECT STRUCTURE

The problem will be split into three subproblems:

- **Early detection of the failure**
  - Detect the failure at its early development → 1D convolutional autoencoder
- **Classification of the failure**
  - Identify the position of the failure → Siamese network/Triplet learning

PROJECT STRUCTURE

The problem will be split into three subproblems:

- **Early detection of the failure**
  - ⟹ Detect the failure at its early development ⟹ 1D convolutional autoencoder
- **Classification of the failure**
  - ⟹ Identify the position of the failure ⟹ Siamese network/Triplet learning
- **Continuous learning of the models**
  - ⟹ Re-train the models when new data arrives ⟹ Elastic weight consolidation

ANALYTICAL SOLUTION: FAILURE DETECTION

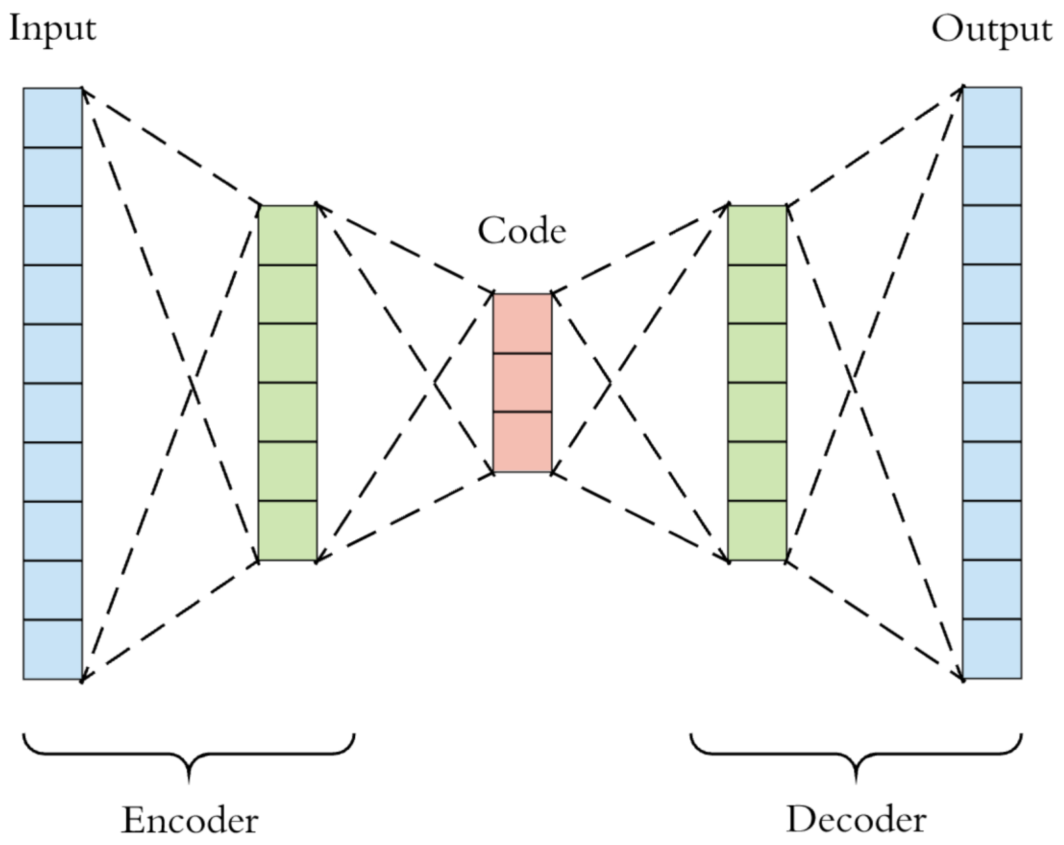
EARLY DETECTION OF THE FAILURE

1D Convolutional autoencoder (AE)

- The AE tries to reconstruct the input data:

$$\mathcal{L}(x, g(f(\tilde{x}))) = \mathrm{MAE} = |x - \hat{x}|$$

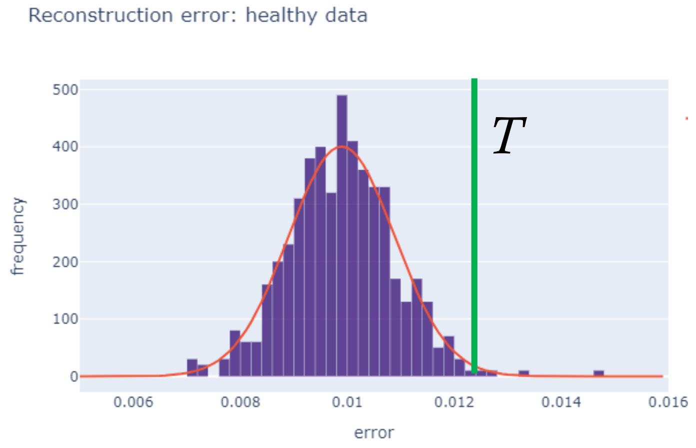
- AE for anomaly detection:
  - The model is train only with *healthy* data
  - The *reconstruction error (MAE)* is used to detect anomalous data (bearing failure).



$$\text{Encoder: } h := f(x)$$
$$\text{Decoder: } \hat{x} := g(h) = g(f(x))$$

EARLY DETECTION OF THE FAILURE

We use the reconstruction error probability distribution to set a threshold for the beginning of the failure.

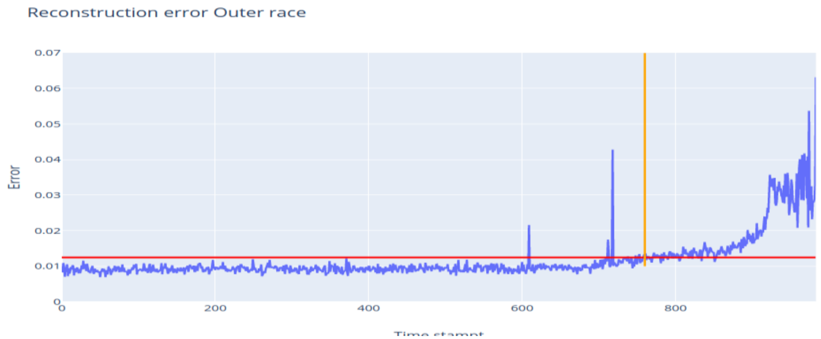


$$T = 0.0125$$

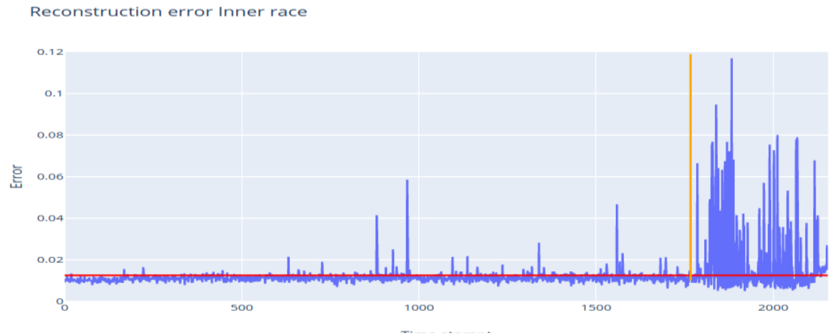
Beginning of the failure		
Inner ring	Outer ring	Rolling el.
1780	760	1700

Results

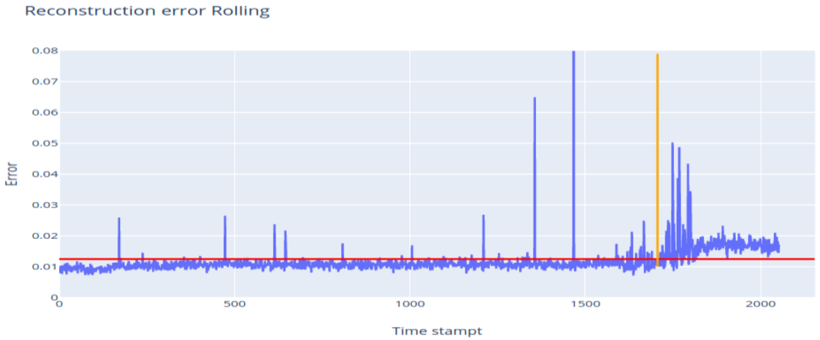
Outer ring



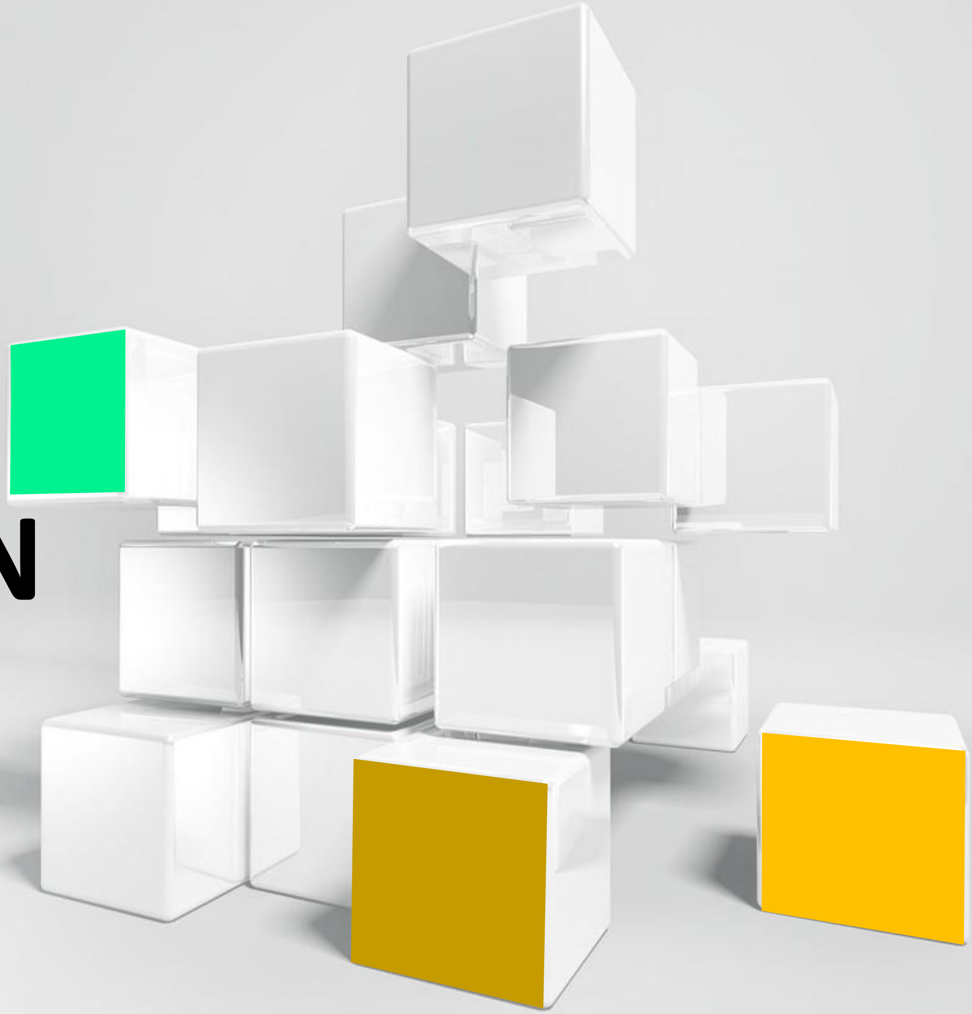
Inner ring



Rolling element



ANALYTICAL SOLUTION: FAILURE CLASSIFICATION



FAILURE MODE CLASSIFICATION

Goal:

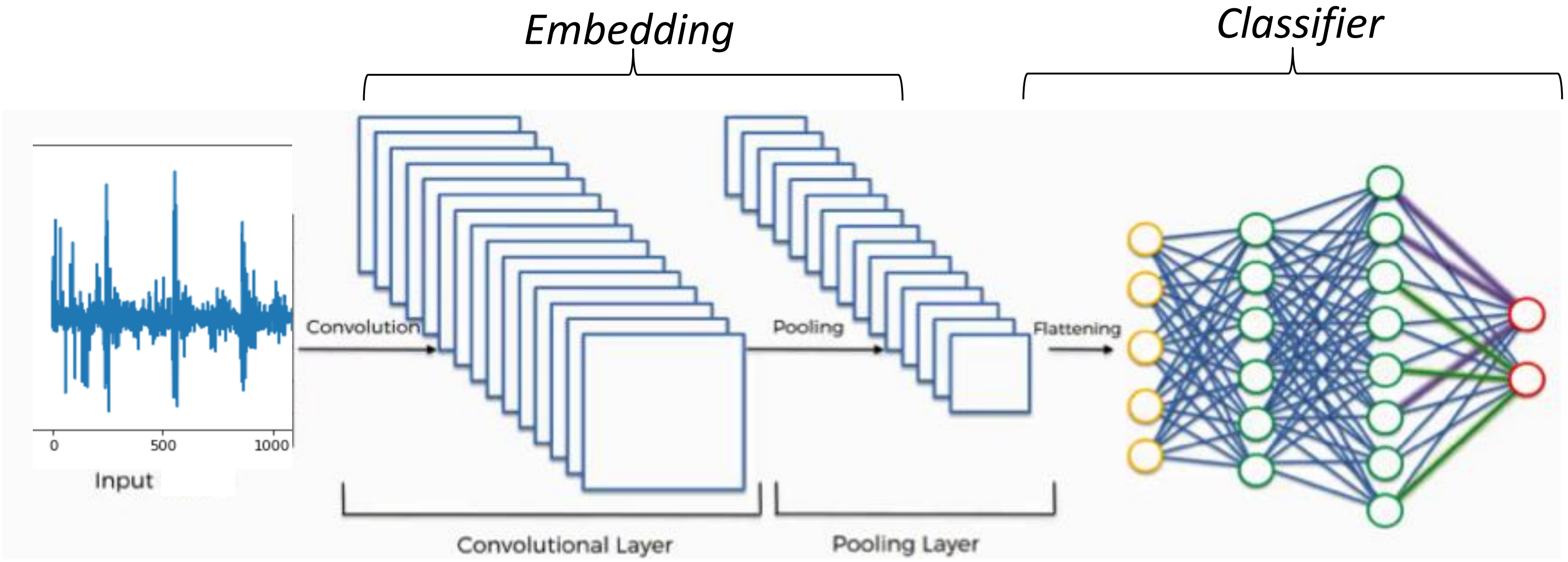
Design a Deep learning classification model which identifies the position of the bearing failure.

The model must be *robust to different operating conditions* and must work with *small data sets*.

Design:

The model consists of a *Convolutional Neural Network* divided into two parts:

- **Embedding:** extracts valuable features from the training data ⇨ Siamese network/Triplet learning
- **Classifier:** returns the type of fault of the bearing



SIAMESE NETWORK

Goal:

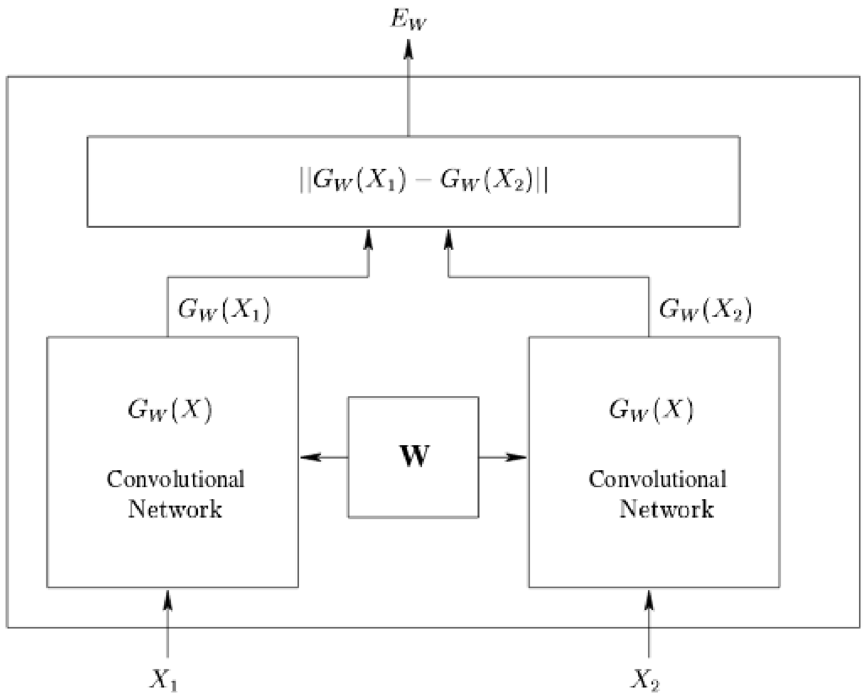
Construct an embedding of the input data such that:

- Two examples with the **same label** have their embeddings **close together** in the embedding space.
- Two examples with **different labels** have their embeddings **far away**.

The network is trained using the **contrastive loss:**

$$\mathcal{L}(W) = \sum_{i=1}^{N} \mathcal{L}(W, (Y, X_1, X_2)^i),$$

$$\mathcal{L}(W, (Y, X_1, X_2)^i) = \underbrace{(1-Y)\cdot\frac{1}{2}E_W^2}_{\text{Loss for same-class pairs}} + \underbrace{Y\cdot\frac{1}{2}\{\max(0, 1-E_W)\}^2}_{\text{Loss for different-class pairs}}$$



TRIPLET LEARNING

Goal:

Construct an embedding of the input data such that:

- Two examples with the **same label** have their embeddings **close together** in the embedding space.
- Two examples with **different labels** have their embeddings **far away**.

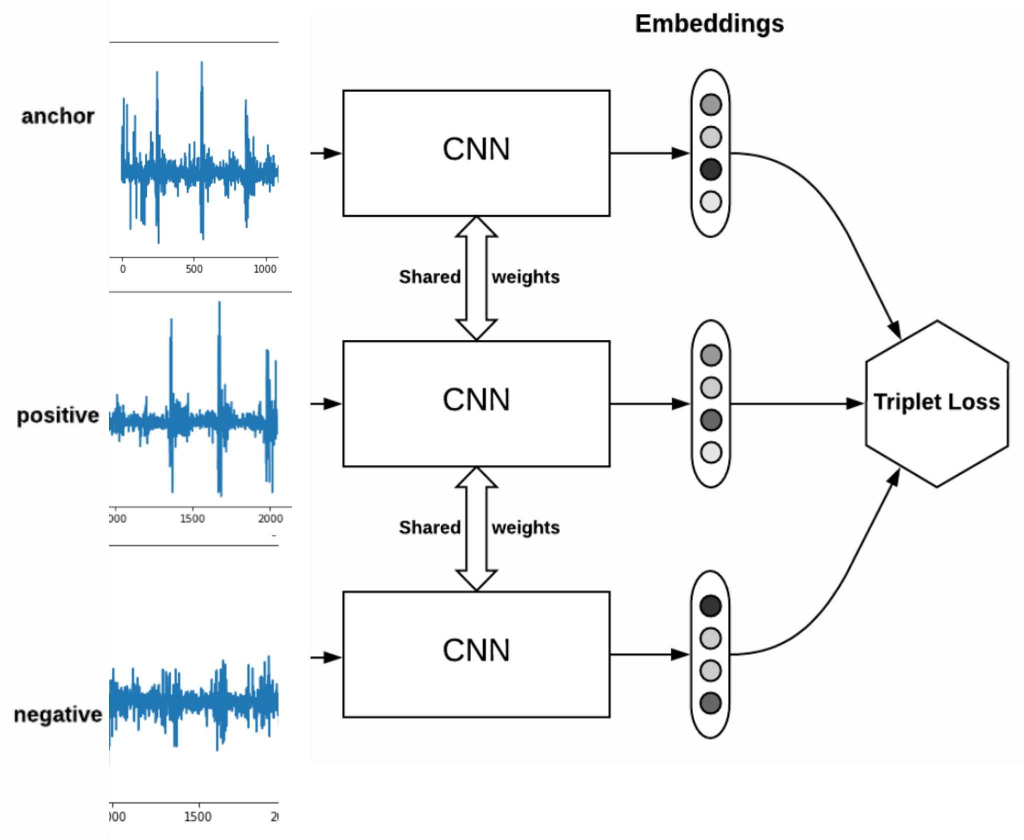
The loss will be defined over **triplets** of embeddings:

- an **anchor**
- a **positive** of the same class as the anchor
- a **negative** of a different class

For some distance on the embedding space d, the loss of a triplet (a,p,n) is:

$$\mathcal{L} = max(d(a,p) - d(a,n) + margin, 0)$$

Given two positive examples of the same class and one negative example, the negative should be farther away than the positive by some margin



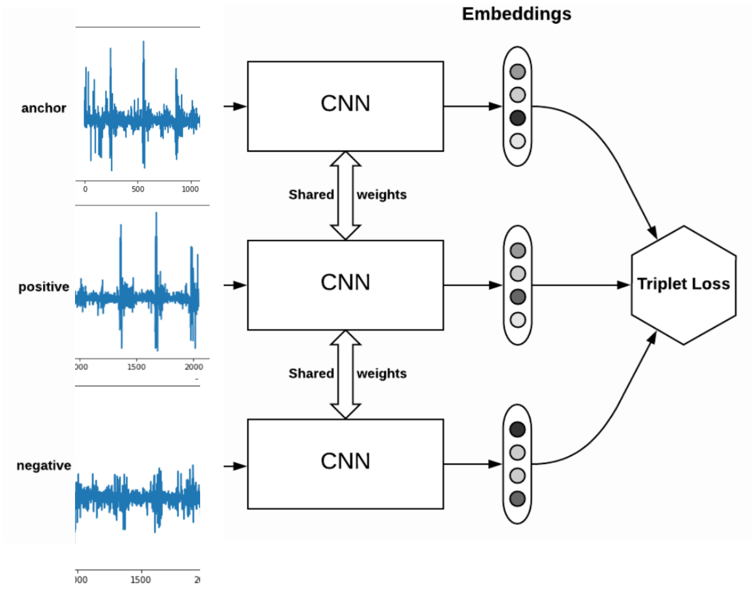
TRIPLET LEARNING

There are three types of triplets:

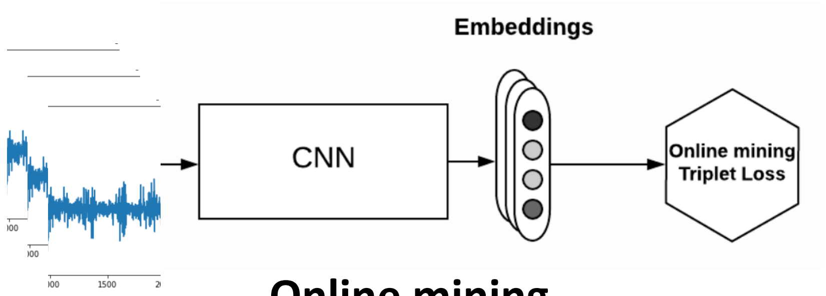
- **Easy triplets:** $d(a,p) + margin < d(a,n)$
- **Hard triplets:** $d(a,n) < d(a,p)$
- **Semi-hard triplets:** $d(a,p) < d(a,n) < d(a,p) + margin$

The choice of which triplets to use on training will greatly impact the model's performance. There are 2 strategies:

- **Offline mining:** Select the triplets at the beginning of every epoch. It is not very effective because some of the triplets will be easy triplets naturally, and will not help learning the embedding.
- **Online mining:** Select the triplets **for each batch** of the training process. Given a batch of B, there are at most B^3 triplets. Select the valid ones (anchor, positive, negative). Then train on the hard and semi hard triplets **(batch all)** or just the triplets containing the hardest positive and hardest negative **(batch hard).**



Offline mining

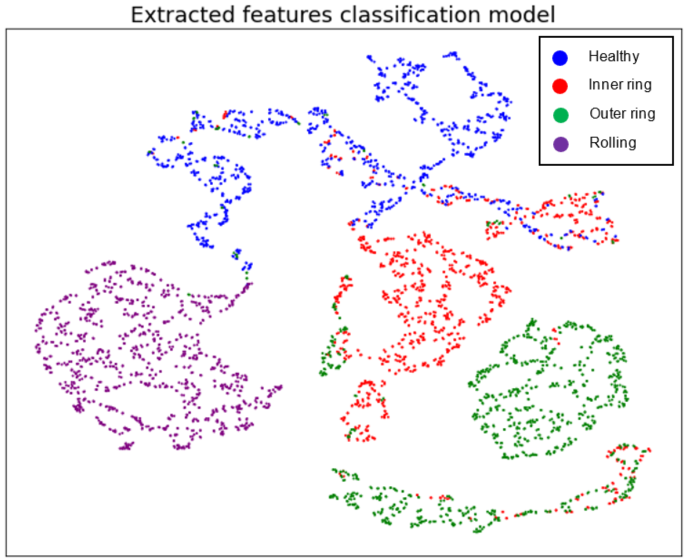


Online mining

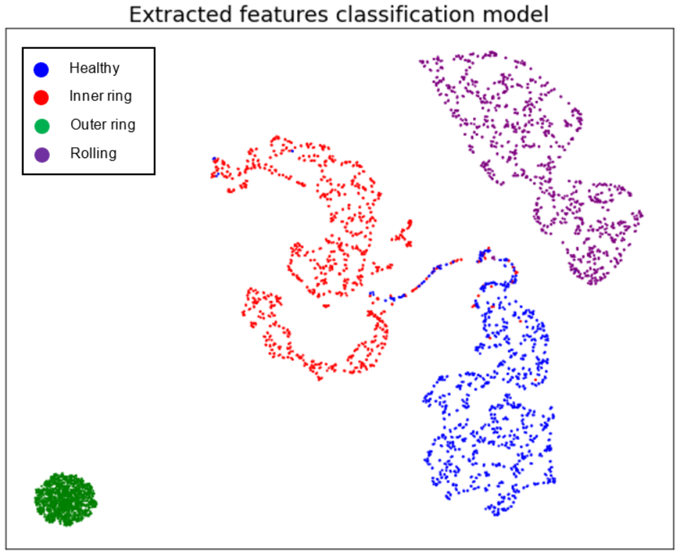
RESULTS

Visualization of features extracted by the models:

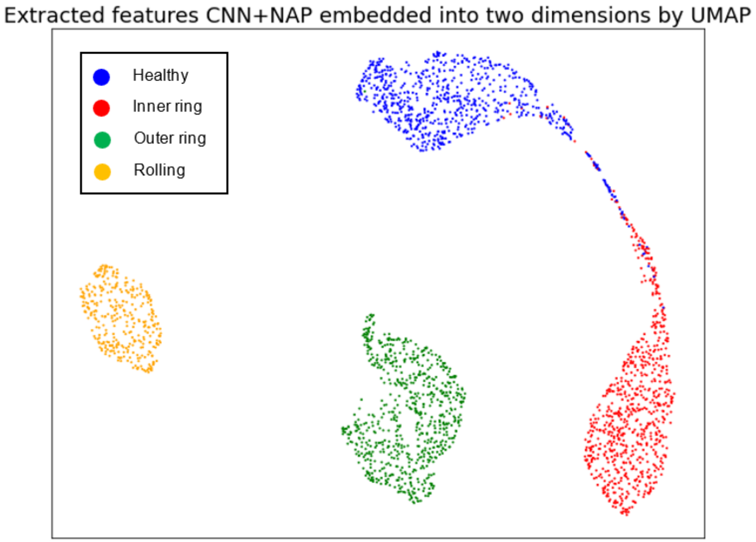
CNN



Siamese network



Triplet learning



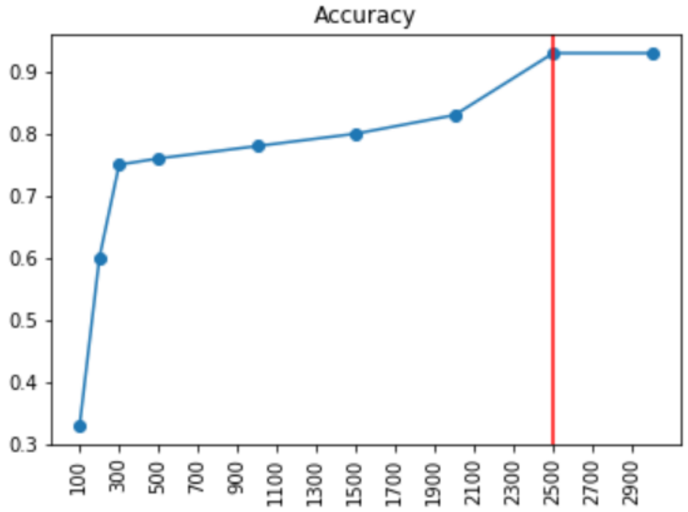
	CNN	Siamese network	Triplet learning
Accuracy	0.90	0.97	0.98

RESULTS

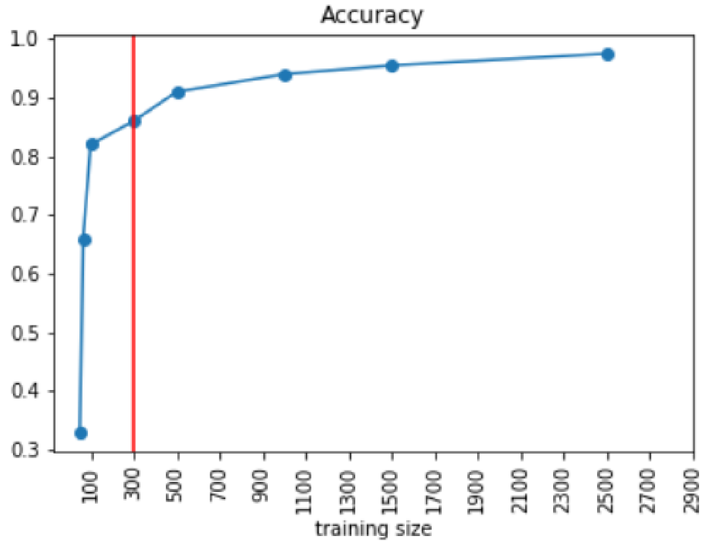
Comparison of both models:

- Both methods give similar performance.
- Triplet learning is more appropriate for small data sets.

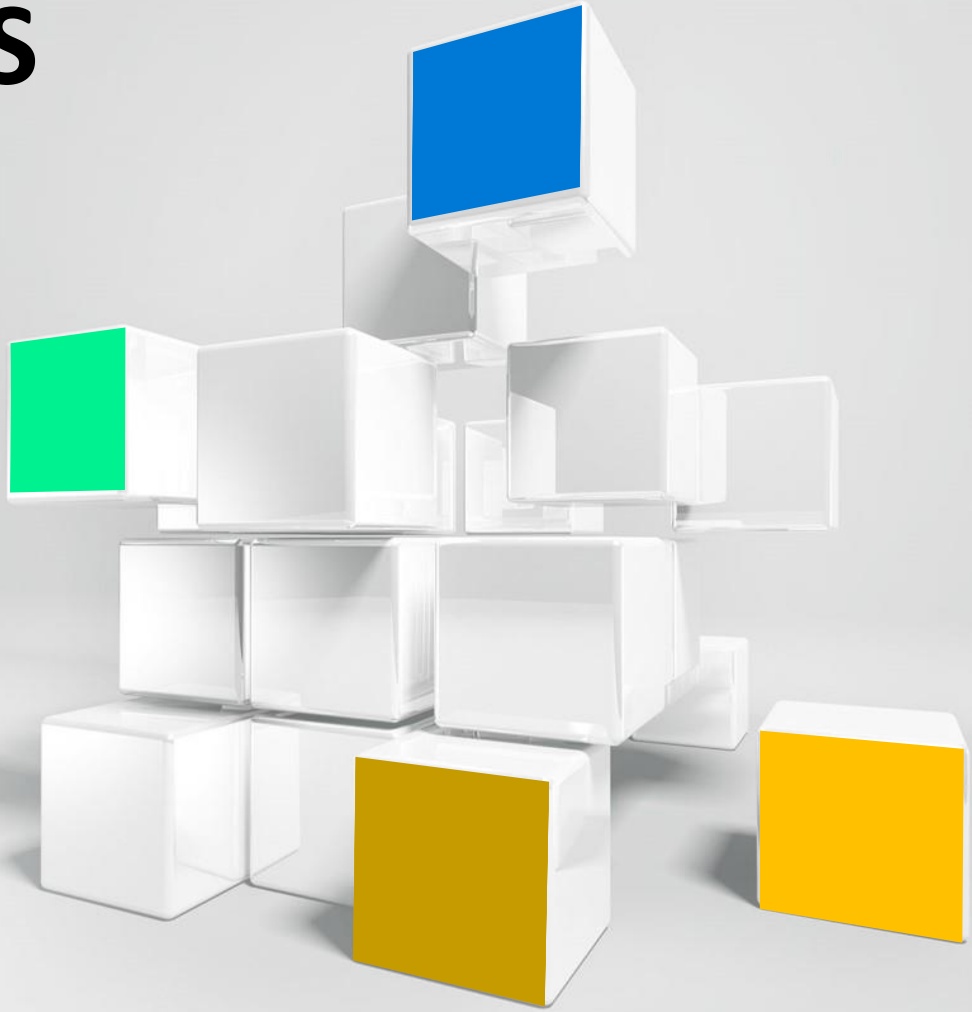
Siamese network



Triplet learning



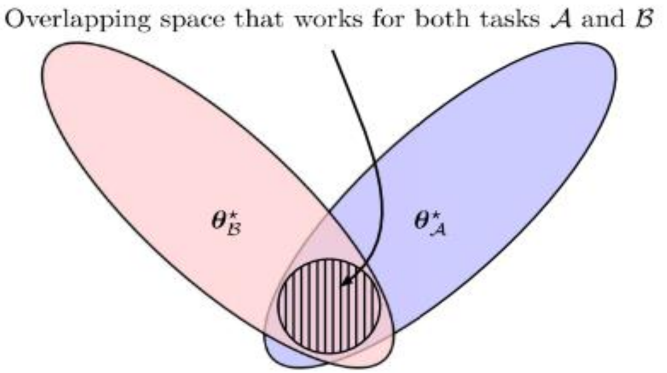
CONTINUOUS LEARNING



ELASTIC WEIGHT CONSOLIDATION

$$\mathcal{L}_{B|A}(\theta) = \mathcal{L}_B(\theta) + \lambda(\theta - \theta_A^*)^T \mathbb{1} (\theta - \theta_A^*)$$

Penalize changing weights that are important for the old data.



Learn new weights that are suitable for both data

Weights for old data

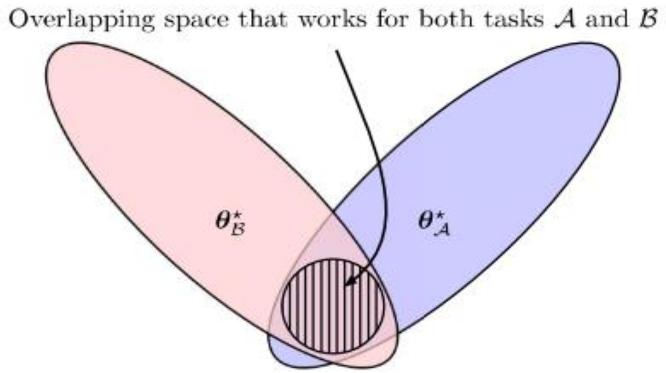
Weights for old + new data

ELASTIC WEIGHT CONSOLIDATION

$$\mathcal{L}_{B|A}(\theta) = \mathcal{L}_B(\theta) + \lambda(\theta - \theta_A^*)^T \cancel{\mathbb{1}} (\theta - \theta_A^*)$$

$$\mathcal{L}_{B|A}(\theta) = \mathcal{L}_B(\theta) + \lambda(\theta - \theta_A^*)^T F_A (\theta - \theta_A^*)$$

Penalize changing weights that are important for the old data.



Learn new weights that are suitable for both data

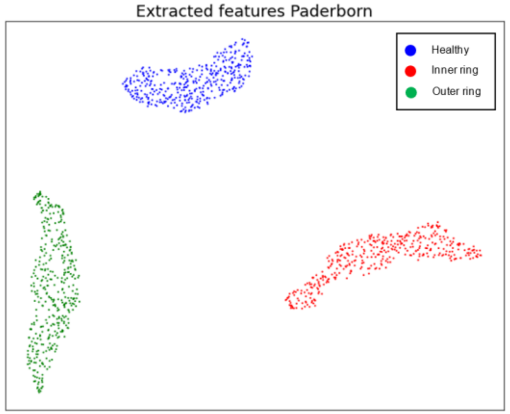
Weights for old data

Weights for old + new data

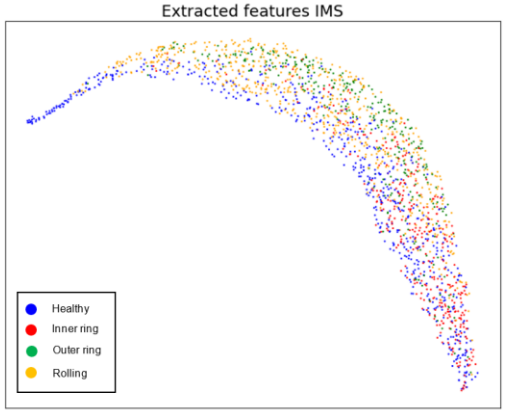
RESULTS

- **Paderborn** data set: *Old* data set
- **IMS** data set: *New* data set

Before EWC



Paderborn



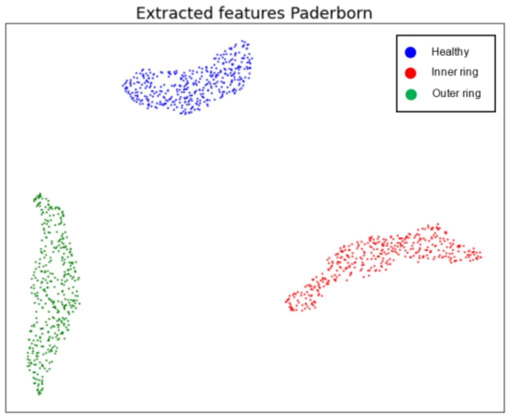
IMS

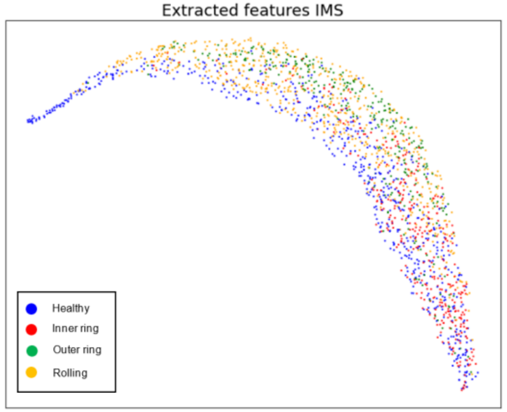
After training task A	
	Accuracy
Task A (Paderborn)	1.0
Task B (IMS)	0.19

RESULTS

- **Paderborn** data set: *Old* data set
- **IMS** data set: *New* data set

Before EWC



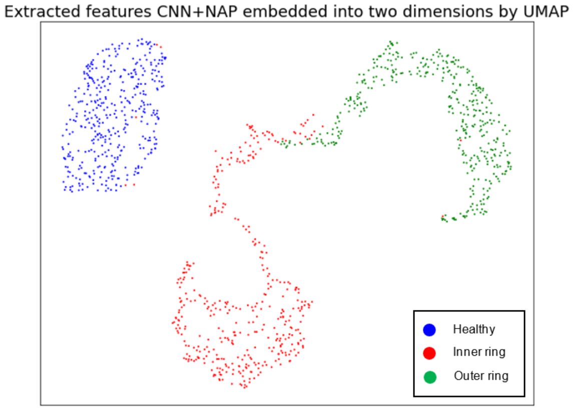


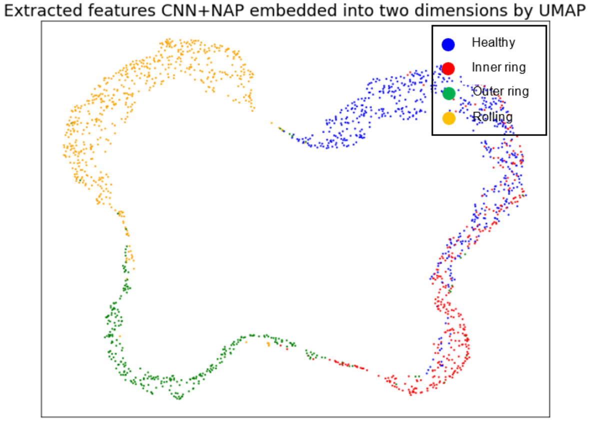
After training task A	
	Accuracy
Task A (Paderborn)	1.0
Task B (IMS)	0.19

Paderborn

After EWC

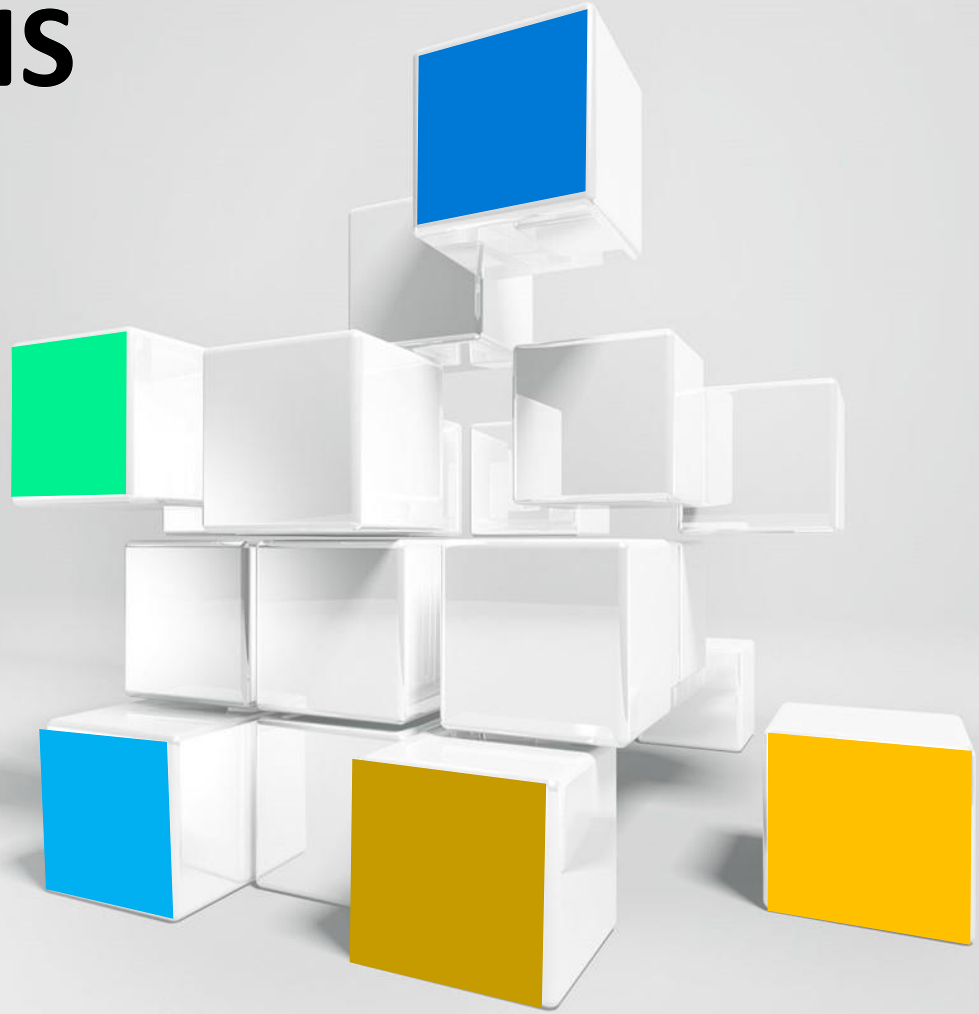
IMS





After training task B	
	Accuracy
Task A (Paderborn)	0.87
Task B (IMS)	0.90

CONCLUSIONS



CONCLUSIONS

- Monitoring the health of rolling bearing is of grat importance in the industry sector.
- Vibration analysis can be very useful in diagnosing the bearing damage when sufficient information of the bearing is available.
- The autoencoder model is useful to detect the beginning of the development of the bearing failure.
- Both siamese network and triplet learnin strategies are useful to classify failures regardless of the operation condition.
- However, triplet learning strategies are more robust to small data sets.
- Elastic weight consolidation is useful to train the model with multiple data sets, when all the data is not available at the same time.

Presentation