Meta-Learning for Cross-Domain Bearing Fault Detection

(Quang Hieu TANG, Khac Tuan HUYNH, Malika KHAROUF)[1]

[1]University of Technology of Troyes, Troyes, France

The 3^{rd} Annual SAGIP Congress

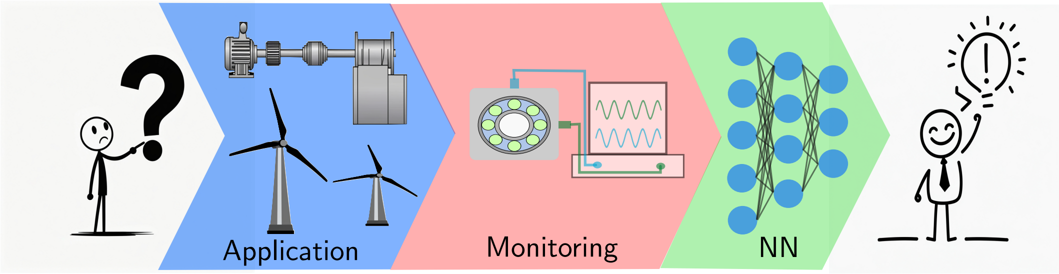
Société d'Automatique, de Génie Industriel & de Productique

utt UNIVERSITÉ DE TECHNOLOGIE TROYES MEMBER OF EUT+ EUROPEAN UNIVERSITY OF TECHNOLOGY

La Région Grand Est

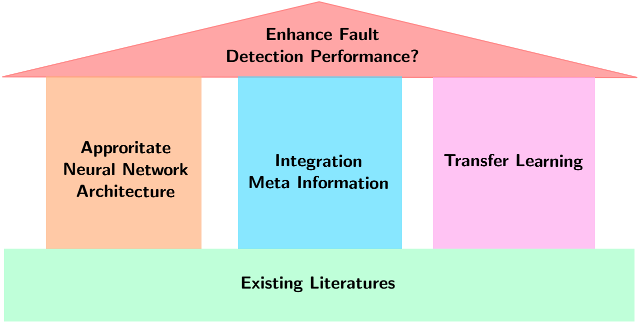
TABLE OF CONTENT

Motivation for the research



- **40%** motor fail due to bearing.
- Condition-based monitoring: Model-based → data-driven approach.
- Machine Learning and Deep Learning ↑ .

Key Research Question



- How to design proper neural network architecture
- Meta Information : Rotation and load
- Transfer learning: source → target.

Contributions

1- Preprocess Step

- Time to Time-Frequency Features
- Meta data normalization

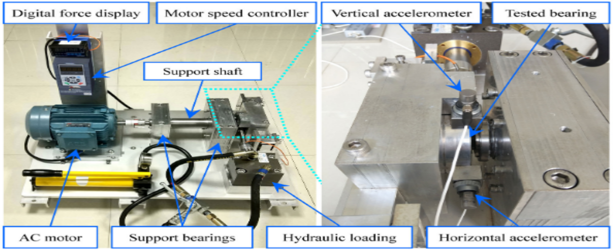
2- Architecture

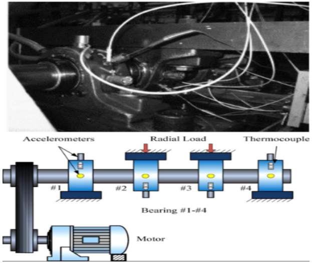
- Dual-Branch Architecture
- Inverted Residual Block

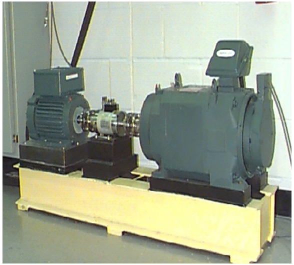
3- Assessment

- Uncertainty quantification

XJTU-SY, IMS, and CWRU dataset settings







XJTU-SY, IMS, and CWRU domain data description

XJTU-SY:

- 15 run-to-fail vibration datasets.
- Domain data includes bearing load and rotation speed.

$$\{V_{\text{rot}}\ (\text{rpm}),\ F_{\text{load}}\ (\text{N})\} = \{(2100, 12\times10^3),\ (2250, 11\times10^3),\ (2400, 10\times10^3)\}$$

IMS:

- Bearing 3 and 4 of set 1, bearing 1 of set 2, bearing 3 of set 3.
- Domain data includes bearing load and rotation speed.

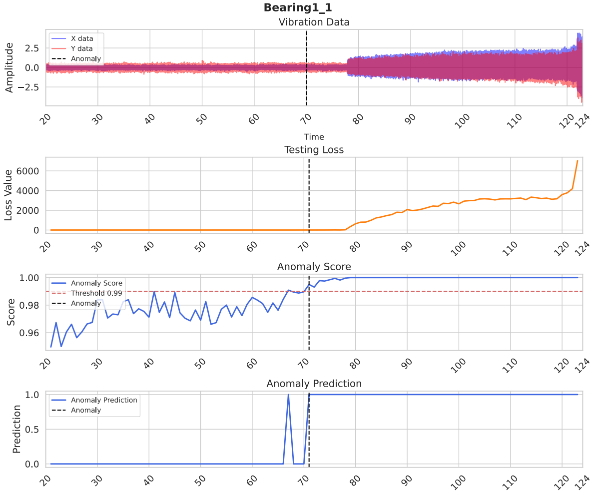
$$\{V_{\text{rot}}\ (\text{rpm}),\ F_{\text{load}}\ (\text{N})\} = \{(2000, 26\times10^3)\}$$

CWRU:

- Normal and drive end datasets.
- Domain data includes bearing load and rotation speed.

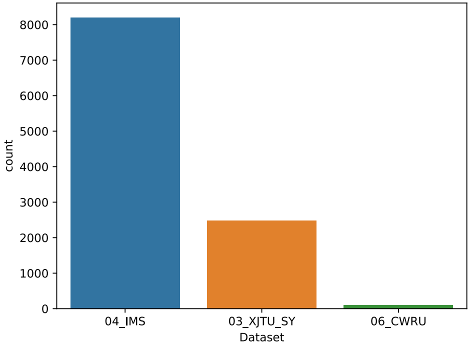
$$\{V_{\text{rot}}\ (\text{rpm}),\ F_{\text{load}}\ (\text{N})\} = \{(1797, 0),\ (1772, 480),\ (1750, 958),\ (1730, 1437)\}$$

Preprocess run-to-fail dataset

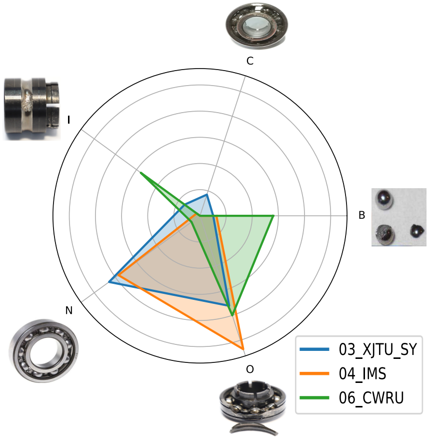


1. Unsupervised learning with AutoEncoder.
2. Calculate reconstruction error → anomaly score.
3. Threshold the anomaly score → split the dataset into normal and abnormal segments.

Data and error type distribution

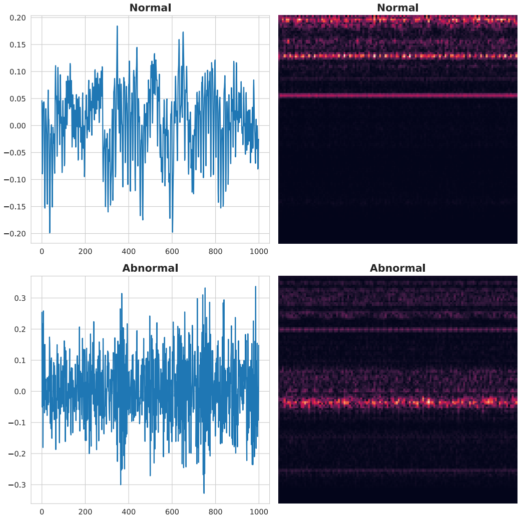


(a) Dataset distribution



(b) Error type distribution

Preprocess vibration data from time domain to time-frequency domain

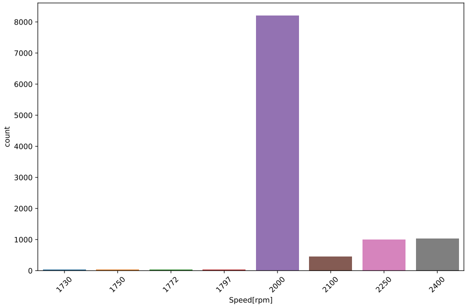


Formular:
STFT$[m, k] = \sum_{n=-\infty}^{\infty} x[n] \cdot w[n-m] \cdot e^{-j2\pi kn/N}$

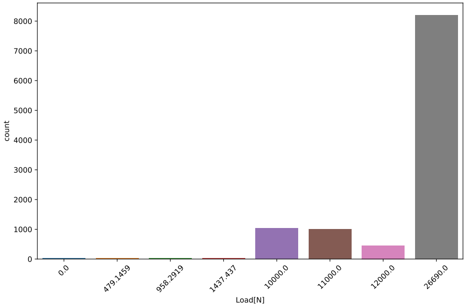
Time-Frequency Localization:
1D domain → 2D domain → **what**, **when** frequency occur.

Non-Stationary Signal Analysis:
Short-lived fault feature or modulated harmonics.

Preprocess min-max scale domain data



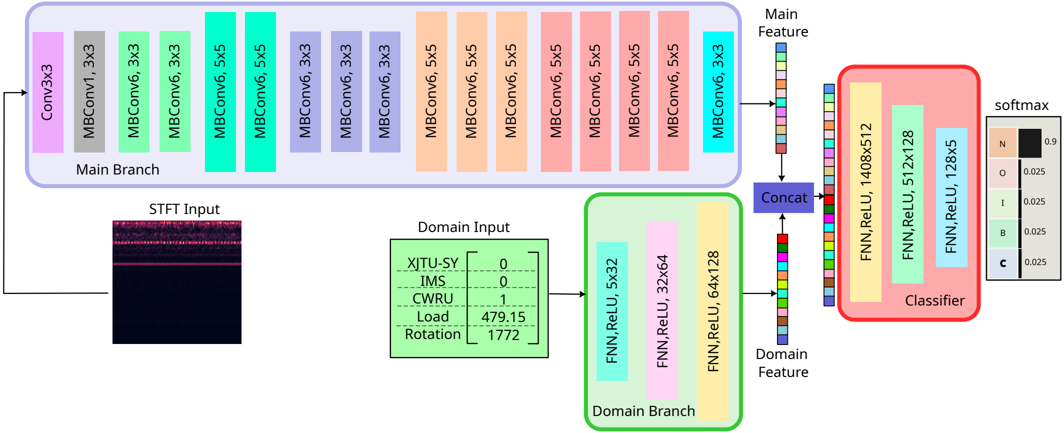
(a) Rotation speed



(b) Bearing load

- Large number → activation saturated → performance ↓
- Min-max scale: $\mathbf{x'} = \frac{\mathbf{x} - \mathbf{x_{min}}}{\mathbf{x_{max}} - \mathbf{x_{min}}}; \mathbf{x'} \in [\mathbf{0}, \mathbf{1}]$

Dual branch architecture diagram



Dual branch architecture overview

Main Branch:

- Utilize EfficientNetB0 as backbone.
- Extract importance features from time-frequency STFT dataset:

$$\mathbf{F}_{\text{main}} = \mathsf{EfficientNetB0}(\mathbf{X}_{\text{STFT}}); \mathbf{F}_{\text{main}} \in \mathbb{R}^{\mathbf{d}_{\text{main}}}$$

Domain Branch:

- A series of fully connected layers.
- Strengthen domain information:

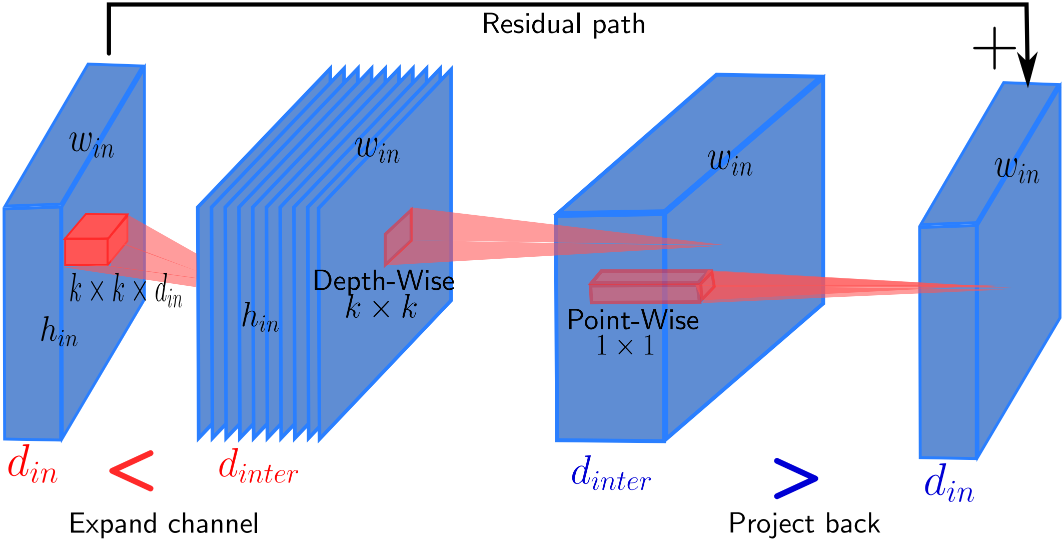
$$\mathbf{F}_{\text{domain}} = \mathsf{FCNs}(\mathbf{X}_{\text{domain}}); \mathbf{F}_{\text{domain}} \in \mathbb{R}^{\mathbf{d}_{\text{domain}}}$$

Classifier:

- Merge main features and domain features.
- Return probabilities of failure:

$$\mathbf{P}\left(\text{fault}|\text{vibration}, \text{domain}\right) = \mathsf{Softmax}\left(\text{FNCs}\left(\mathbf{F}_{\text{main}}, \mathbf{F}_{\text{domain}}\right)\right) \geq \mathbf{P}\left(\text{fault}|\text{vibration}\right).$$

Mobile Convolutional Block (MBConv) diagram



Mobile Convolutional Block (MBConv) architecture

- **Convolutional complexity of a convolutional operation**:

$$O\left(\text{Conv2d}\right) = k \times k \times d_{in} \times h \times w \times d_{out}$$

- **Inverted bottleneck**: Expands channels first before compression.
- **Pros**: Expressiveness ↑ $\neq$ **Cons** : Computational resource ↑ .
- **Depth-wise and point-wise convolution**: Single filter each channel → 1×1 filter

$$\begin{aligned} O\left(\mathrm{D_{wise}} + \mathrm{P_{wise}}\right) &= k \times k \times h \times w \times d_{out} + 1 \times 1 \times d_{in} \times h \times w \times d_{out} \\ &= k \times k \times h \times w \times d_{out} + d_{in} \times d_{out} \times h \times w \end{aligned}$$

- **Gain in complexity**:

$$\frac{O\left(\mathrm{D_{wise}} + \mathrm{P_{wise}}\right)}{O\left(\text{Conv2d}\right)} = \frac{k \times k \times h \times w \times d_{out} + d_{in} \times d_{out} \times h \times w}{k \times k \times d_{in} \times h \times w \times d_{out}} = \frac{1}{d_{in}} + \frac{1}{k \times k}$$

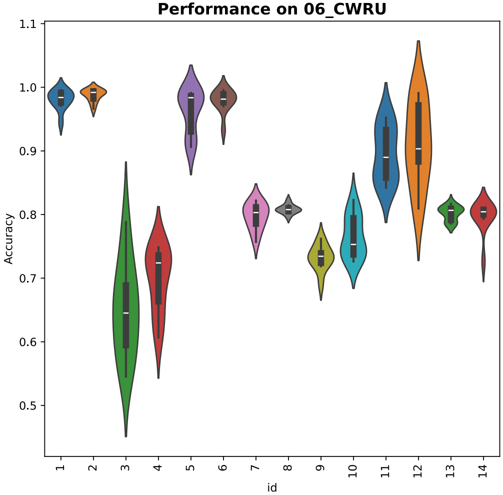
- Kernel size k ↑ → complexity ↓ .

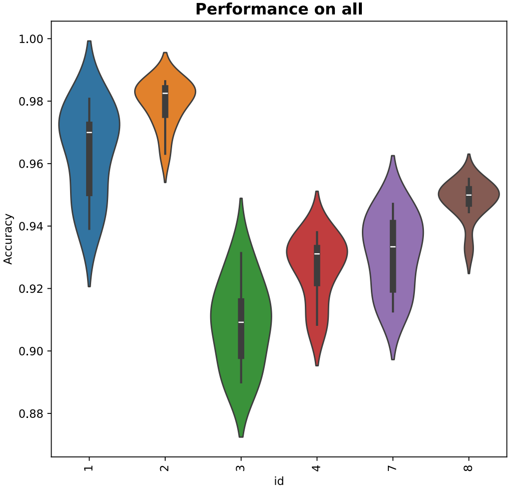
Dataset Train/Test Spliting Strategy

ID	Model	IMS (Source)	XJTU-SY (Source)	CWRU (Target)
1	Baseline	80% train ✔ ; 20% test ✔	80% train ✔ ; 20% test ✔	80% train ✔ ; 20% test ✔
2	Dual-Branch	80% train ✔ ; 20% test ✔	80% train ✔ ; 20% test ✔	80% train ✔ ; 20% test ✔
3	Baseline	20% train ↓ ; 20% test ✔	20% train ↓ ; 20% test ✔	20% train ↓ ; 20% test ✔
4	Dual-Branch	20% train ↓ ; 20% test ✔	20% train ↓ ; 20% test ✔	20% train ↓ ; 20% test ✔
5	Baseline	✖	✖	80% train ✔ ; 20% test ✔
6	Dual-Branch	✖	✖	80% train ✔ ; 20% test ✔
7	Baseline	20% Ball train ↓ ; 20% test ✔	20% Ball train ↓ ; 20% test ✔	20% Ball train ↓ ; 20% test ✔
8	Dual-Branch	20% Ball train ↓ ; 20% test ✔	20% Ball train ↓ ; 20% test ✔	20% Ball train ↓ ; 20% test ✔
9	Baseline	✖	✖	20% Ball train ↓ ; 20% test test ✔
10	Dual-Branch	✖	✖	20% Ball train ↓ ; 20% test test ✔
11	Baseline	✖	✖	20% Inner train ↓ ; 20% test test ✔
12	Dual-Branch	✖	✖	20% Inner train ↓ ; 20% test test ✔
13	Baseline	✖	✖	20% Outer train ↓ ; 20% test ✔
14	Dual-Branch	✖	✖	20% Outer train ↓ ; 20% test ✔

- Stratified split into 80-20%.
- Retain only 20% of training data for in some settings.

Accuracy distributions on target dataset and whole dataset





Results and discussion

- Dual branch models trained on the full dataset 98.76% $\pm$ 0.01
- Baseline model trained on the full dataset 98.07%$\pm$ 0.01
- $\text{mean}\left(acc_{\text{dual}}\right) \geq \text{mean}\left(acc_{\text{baseline}}\right)$ & $\text{std}\left(acc_{\text{dual}}\right) \leq \text{std}\left(acc_{\text{baseline}}\right)$

ID	Model	CWRU (Target)	All
3	Baseline	0.65 $\pm$ 0.08	0.91 $\pm$ 0.01
4	Dual-Branch	0.70 $\pm$ 0.05	0.93 $\pm$ 0.01
5	Baseline	0.96 $\pm$ 0.03	✗
6	Dual-Branch	0.98 $\pm$ 0.02	✗
7	Baseline	0.79 $\pm$ 0.02	0.93 $\pm$ 0.01
8	Dual-Branch	0.8 $\pm$ 0.007	0.95 $\pm$ 0.01

ID	Model	CWRU (Target)	All
9	Baseline	0.73 $\pm$ 0.02	✗
10	Dual-Branch	0.77 $\pm$ 0.03	✗
11	Baseline	0.90 $\pm$ 0.04	✗
12	Dual-Branch	0.91 $\pm$ 0.06	✗
13	Baseline	0.80 $\pm$ 0.01	✗
14	Dual-Branch	0.80 $\pm$ 0.02	✗

Conclusions and Future Work

Conclusions:

1. With current architecture, more data → higher performance.
2. Dual branch models give better performance and more reliable

Future Work:

1. Explore feature importance.
2. Intergrate physics-informed approach.

THANK YOU FOR YOUR ATTENTION!

References

Center for Intelligent Maintenance Systems, University of Cincinnati (2007). *IMS Bearing Data*.

Hakim, Mohammed et al. (2023). "A systematic review of rolling bearing fault diagnoses based on deep learning and transfer learning: Taxonomy, overview, application, open challenges, weaknesses and recommendations". In: *Ain Shams Engineering Journal* 14.4, p. 101945.

Neupane, Dhiraj and Jongwon Seok (2020). "Bearing fault detection and diagnosis using case western reserve university dataset with deep learning approaches: A review". In: *Ieee Access* 8, pp. 93155–93178.

Sandler, Mark et al. (2018). "Mobilenetv2: Inverted residuals and linear bottlenecks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520.

References (cont.)

Tan, Mingxing and Quoc Le (2019). "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *International conference on machine learning*. PMLR, pp. 6105–6114.

Wang, Biao et al. (2018). "A hybrid prognostics approach for estimating remaining useful life of rolling element bearings". In: *IEEE Transactions on Reliability* 69.1, pp. 401–412.