Fully learnable deep wavelet transform for unsupervised monitoring of high-frequency time series

Gabriel Michau[a], Gaetan Frusque[a], and Olga Fink[a,1]

[a]Chair of Intelligent Maintenance Systems, ETH Zürich, 8049 Zürich, Switzerland

Edited by David Donoho, Department of Statistics, Stanford University, Stanford, CA; received April 7, 2021; accepted January 10, 2022

High-frequency (HF) signals are ubiquitous in the industrial world and are of great use for monitoring of industrial assets. Most deep-learning tools are designed for inputs of fixed and/or very limited size and many successful applications of deep learning to the industrial context use as inputs extracted features, which are a manually and often arduously obtained compact representation of the original signal. In this paper, we propose a fully unsupervised deep-learning framework that is able to extract a meaningful and sparse representation of raw HF signals. We embed in our architecture important properties of the fast discrete wavelet transform (FDWT) such as 1) the cascade algorithm; 2) the conjugate quadrature filter property that links together the wavelet, the scaling, and transposed filter functions; and 3) the coefficient denoising. Using deep learning, we make this architecture fully learnable: Both the wavelet bases and the wavelet coefficient denoising become learnable. To achieve this objective, we propose an activation function that performs a learnable hard thresholding of the wavelet coefficients. With our framework, the denoising FDWT becomes a fully learnable unsupervised tool that does not require any type of pre- or postprocessing or any prior knowledge on wavelet transform. We demonstrate the benefits of embedding all these properties on three machine-learning tasks performed on open-source sound datasets. We perform an ablation study of the impact of each property on the performance of the architecture, achieve results well above baseline, and outperform other state-of-the-art methods.

fast discrete wavelet decomposition | deep learning | high-frequency signals | unsupervised anomaly detection | sparse decomposition

Monitoring of industrial assets often relies on high-frequency (HF) signal measurements, such as electric current, vibrations, or sound. One difficulty of dealing with such signals in the industrial context is the conciliation between the high-frequency sampling and low-dimensional decision states (e.g., healthy/unhealthy), in a context where, very often, labels are not available. Therefore, many industrial applications require unsupervised approaches able to extract meaningful and sparse information from HF signals, to ease the process analysis, the diagnostics, and more generally the optimization of the assets' life cycles.

Before the recent developments of large storage capacity and high computational powers, raw HF signals could not be recorded, forcing companies to spend time and budget on devising relevant features for later analysis, achieving in that way a sparse representation of the input data. These features could be of various natures, such as spectral features, based on the Fourier transform, the fast Fourier transform, or wavelets (1); on statistical features (moments, energy, entropy, etc.); or on descriptive features (envelopes, amplitude, etc.) (2).

In recent days, storing HF data has become less of a technical problem, and handling large datasets efficiently has been made possible with the rise of deep learning. However, most deep-learning tools are designed for inputs of fixed and/or very limited size. Many successful applications of deep learning to industrial context use as inputs extracted features, that is, a manually obtained compact representation of the original signals. Very often these features are a spectrogram (3), wavelet coefficient statistics (4), or others (5–7). Although such frameworks are extremely successful, they still require careful feature extraction with the right hyperparameters, which can be a time-consuming task. In addition, the extracted features might be sensitive to unexpected noise or to changing conditions, and the design of domain invariant unsupervised features, whether with postprocessing or with deep learning, is still an open research question (8).

With the development of convolutional neural networks (CNN) (9), early works realized that a temporal CNN is equivalent to a digital filter and that it could learn convolution kernels similar to a Fourier transform or to wavelets (10), or also be used to learn sparse representations (11). It was soon proposed to constrain the network to perform operations similar to the Fourier transform (12, 13) or to wavelet transform either with continuous wavelet transform (14, 15) or with discrete wavelet transform (16–19). All these works demonstrate that by using architectures or kernels inspired by spectral analysis, superior results could be obtained on supervised deep-learning tasks. Yet, these approaches are rarely adapted to unsupervised machine-learning tasks and the link with the spectral transformation is often restrained either to the network architecture only or to the initialization of the convolution kernels. In addition, Fourier transform-based deep-learning architectures become rapidly too heavy to handle when the size of the input time series increases.

To mitigate the abovementioned limitations, we propose in this work a deep-learning framework based on the fast discrete wavelet transform (FDWT) that allows an automatic and easy extraction of meaningful and sparse representation of the input

Significance

Monitoring of industrial assets often relies on high-frequency (HF) signal measurements. One difficulty of dealing with such measurements in the industrial context is the conciliation between the high-frequency sampling and low-dimensional decision states (e.g., healthy/unhealthy), in a context where, very often, labels are not available. Here, we propose a fully unsupervised deep-learning framework for high-frequency time series that is able to extract meaningful and sparse representation of raw signals and is able to handle different lengths of time series flexibly, overcoming thereby several of the limitations of existing deep-learning approaches. The decomposition framework will be very useful for handling in an automatic manner high-frequency signals and is an important basis for future applications with HF data.

Author contributions: G.M. and O.F. designed research; G.M. performed research; G.M., G.F., and O.F. analyzed data; and G.M., G.F., and O.F. wrote the paper.

The authors declare no competing interest.

This article is a PNAS Direct Submission.

[1]To whom correspondence may be addressed. Email: ofink@ethz.ch.

Published February 18, 2022.

signals. First, we propose here to mimic the FDWT cascade architecture utilizing the deep-learning framework. We, thus, propose to learn at each decomposition level on the one hand the right high- and low-pass filters and on the other hand the right hard-thresholding coefficients for denoising. Second, for the learning of the filters, we take advantage of the long-known advantageous properties of orthogonal wavelet filter banks with the conjugate quadrature filter property to structure the learning process while making sure the final network remains similar to a FDWT operation. It also results in a very light architecture with only few parameters to learn (a few hundred). This is opposite to the general trend in deep learning to learn millions or billions of parameters. Third, to learn the right hard-thresholding operation, we propose a learnable activation function that is continuous and differentiable, and approximates the hard-thresholding operation. It can, thus, be used as an integral part of the deep-learning architecture and removes the need for human analysis and decisions on these difficult tasks (20).

After presenting in *1. Background on the Cascade and FDWT* important properties and characteristics of the FDWT, we show how to translate these properties into a deep neural network in *2. Learnable Denoising Sparse Wavelet Network*. In *3. Comparison between Traditional FDWT and DeSpaWN*, we test our approach on three tasks, one classification task and two unsupervised anomaly detection tasks. Starting from the FDWT, we demonstrate how each of our different contributions contributes toward better results, well above the baseline. Finally, in *4. Comparison to Other Frameworks*, we compare our approach to several other architectures, such as the scattering transform (21), U-Net (22), and a convolutional autoencoder. We show that without the need of any preprocessing steps, with our particularly light architecture, we achieve very competitive results.

1. Background on the Cascade and FDWT

A. Cascade Algorithm. The FDWT uses wavelets designed such that the family of wavelets made of their scaling and translation by any power of 2 makes an orthonormal family (23). Using such wavelets, and their corresponding scaling function, FDWT decomposes successively each approximation of a signal f into a coarser approximation a (low-pass–filtered version of the signal f) and its detail coefficients d (also denoted as wavelet coefficients, high-pass–filtered version of f). With an orthonormal basis of $\mathbf{L}^2(\mathbb{R})$, the decomposition of any function with such a transformation is invertible. Additionally, one can demonstrate that, due to the factor 2 in scale between the levels and in translation between the coefficients, the decomposition at level l can be expressed as a function of the previous approximation, subsampled by a factor 2 and the original nondilated wavelet (interested readers are referred to ref. 23, chap. 7.3.1 for the proof). Similarly, for the reconstruction at each level, it can be expressed as the convolution of the conjugate of the original wavelet with the previous level reconstruction, where zeros have been interpolated between every sample. FDWT uses this property to apply, in cascade, the exact same operation at each level, using a single wavelet, but down-scaling and interpolating with zeros the signals at each level of the decomposition and the reconstruction, respectively. It is denoted as the cascade algorithm. Fig. 1 illustrates the FDWT cascade algorithm, where g is the wavelet, h is the scaling function, and $\bar{g}$ and $\bar{h}$ are their respective conjugate filters.

B. CQF Properties. An interesting property of the fast discrete wavelet transform was discovered in 1976 by Croisier, Esteban, and Galand (23, 24) and was extended in 1984 (23, 25, 26). It establishes the ground for finding filters, allowing a perfect reconstruction of the input signal using the conjugate quadrature filter (CQF) bank property. The quadrature property ensures a symmetric response of the decomposition filters with respect to the cutoff frequency and ensures thus an antialiasing property. To do so, the filters can be designed such that the wavelet function g is the alternative flip of the scaling function h. The conjugate property ensures that the reconstruction filters have an anticancellation property. Both properties combined are usually denoted as a CQF, which is formalized as follows:

$$\begin{cases} g[n] &= (-1)^n \cdot h[-n], \\ \bar{h}[n] &= h[-n], \\ \bar{g}[n] &= (-1)^{(n+1)} \cdot h[n], \end{cases} \quad [1]$$

where $h[-n]$ denotes the nth coefficient of h in reversed order.

To achieve a perfect reconstruction, the frequency content conservation after applying the filters imposes that the sum of the responses of both filters should be 2, which imposes further

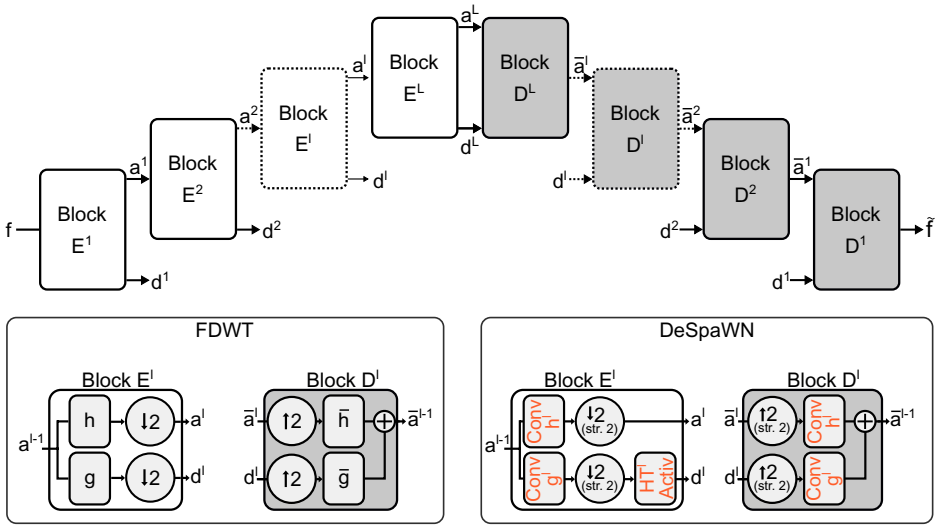

Fig. 1. Cascade: FDWT versus DeSpaWN. The FDWT can be modeled as a convolution neural network, an autoencoder of 2L layers, comprising L encoding blocks and L decoding blocks. Each encoding block has two outputs, one of which is connected to the corresponding decoding block with skip connections. In this work, we propose to make the network learnable, that is, to learn the right filters. In addition, we propose a learnable hard-thresholding activation function that allows one to learn the wavelet coefficient denoising operation at the same time. Red elements shown are learnable. The proposed architecture mimics the denoising FDWT and is denoted as DeSpaWN.

constraints on the filters' orthonormality (23), usually considered as part of the CQF properties.

C. Signal Denoising with FDWT. One of the major applications of the wavelet analysis is signal denoising (27, 28). The usual assumption is that regular and structured signals, once decomposed under the right wavelet basis, will naturally lead to a sparse decomposition (23). They will activate only certain wavelet coefficients at specific time and decomposition levels. As a consequence, noise, by nature unstructured, activates the wavelet filters at any level, but usually with a small amplitude. Thus, denoising usually consists of applying a hard thresholding function to the resulting wavelet coefficients before applying the reconstruction algorithm (29). However, finding the right thresholding parameters is a difficult task that has been the topic of extensive research (20).

2. Learnable Denoising Sparse Wavelet Network

A. Architecture Overview. In this research, we propose the denoising sparse wavelet network (DeSpaWN). DeSpaWN utilizes a fully learnable cascade network, mimicking an L-level wavelet cascade, such as illustrated in Fig. 1. It, thus, consists of L encoding blocks and L decoding blocks. Each encoding block l is composed of two learnable convolutional layers g^l and h^l with a stride of two, analogous to the wavelet and scaling filters with down-sampling in the FDWT. The resulting coefficients are fed to a specifically designed learnable hard-thresholding layer HT^l, which is similar to a wavelet denoising operation and to the next block for further decomposition. Similar to the FDWT, each decoding block takes two inputs, the coefficients from the previous block and the detail coefficients from its corresponding encoding block l. It applies to each input a learnable convolution transpose layer $\bar{g}^l$ and $\bar{h}^l$ with a stride of two and sums the results of both layers together. We designed the deep neural network with two main distinctive properties: First, all convolution kernels and second, all positive and negative hard thresholds are learnable. This makes it possible to learn fully and in a completely unsupervised way the most adapted denoising FDWT for the input signals.

In addition, according to the work in ref. 30, a sparsely connected deep neural network, such as the one proposed here, is able to approximate representation systems that encompass and are more general than the representation system provided by the wavelet.

Overall, with the proposed architecture, the network has $(k_n + 2) \cdot L$ learnable parameters, where k_n is the number of coefficients of the wavelets. For example, mimicking Daubechies-4 wavelets, k_n is set to 8 and the network has, thus, $10 \cdot L$ learnable parameters. Since L cannot be set larger than the nearest second logarithm of the training input size, the number of parameters is unlikely to exceed a few hundred.

B. Objective Function. In this work, we propose to learn the best wavelet and scaling functions for achieving a sparse decomposition. This means that we have two learning objectives: first, a good signal reconstruction and second, a sparse decomposition. Sparsity is usually measured by achieving the smallest ℓ_0 norm of the resulting coefficients. Yet, this is a nonconvex metric. A typical convex surrogate of the ℓ_0 norm is the ℓ_1 norm. Part of our objective function should be designed to minimize the ℓ_1 norm of the obtained wavelet coefficients. As a consequence, we propose that for the second part of our objective function, we also measure the ℓ_1 norm of the reconstruction error to make the two terms comparable.

We, thus, train our network with the following objective function:

$$\mathcal{L} = \frac{1}{\text{Card}(f)} |f - \tilde{f}|_1 + \gamma \cdot \mathbf{L}\left(\{\mathbf{d^l}\}_{l\in[1..L]}, \mathbf{a^L}\right), \qquad [2]$$

where the first part of the loss is the averaged ℓ_1 norm of the residuals on the reconstruction and the second part is the sparsity term, here proposed as the average of all wavelet coefficient moduli and of the last layer approximation coefficient modulus,

$$\mathbf{L}\left(\{\mathbf{d^l}\}_{l\in[1..L]}, \mathbf{a^L}\right) = \frac{1}{\text{Card}(\{\mathbf{d^l}\}_{l\in[1..L]}, \mathbf{a^L})} \sum_{l\in[1..L]} |\mathbf{d^l}|_1 + |\mathbf{a^L}|_1. \qquad [3]$$

C. CQF-Constrained Architecture. As seen in the previous section, the CQF property for the wavelet filters ensures a perfect and antialiasing reconstruction. We, thus, propose to utilize this property, at the same time to simplify the learning process and to harness its advantages. As a matter of fact, using the CQF property as defined in Eq. **1**, learning a single kernel would define the other three. The second advantage is that, based on the above idea of penalizing the ℓ_1 norm of the wavelet coefficient as a surrogate for the nonconvex ℓ_0 norm, the constraint-free learning of the kernels might lead toward a state where the coefficients of g and h are minimized to result in a small ℓ_1 value in the latent space, while the reconstruction is still possible due to large coefficients in $\bar{g}$ and $\bar{h}$, which goes against the original goal of achieving sparsity. Constraining the values of $\bar{g}$ and $\bar{h}$ based on those of g and/or h mitigates this issue.

One option would be to learn a single kernel h^0; use the CQF constraints to derive g^0, $\bar{g}^0$, and $\bar{h}^0$; and then impose that all layers of the network use the same kernels, mimicking in that way the traditional wavelet decomposition with a single wavelet basis. However, we state that this approach would not benefit from the full potential of deep learning.

Alternatively, we propose to learn one kernel per layer $\{h^l\}_{l\in[1..L]}$ and to use the CQF property to constrain the other three kernels of the layer (g^l, $\bar{h}^l$, $\bar{g}^l$). A similar alternative would be to learn independently the low-pass h^l and high-pass g^l filters but impose the partial CQF constraints such that $\bar{h}^l[n] = h^l[-n]$ and $\bar{g}^l[n] = g^l[-n]$. These two types of architectures learn at each level the best wavelets to describe the signal at this scale in a sparse way.

We implement all these different variations of the CQF property and compare them in *3. Comparison between Traditional FDWT and DeSpaWN*.

Finally, with the proposed architecture we already cover our three training objectives: a good and a sparse reconstruction due to the objective function and a stable learning due to the constraints imposed between decomposition and reconstruction filters. Therefore, we do not try to impose further constraints on the filters, in particular not the orthonormality property.

D. Learnable Denoising. One assumption behind the wavelet decomposition is that a structured signal should lead to sparse coefficients under the right wavelet basis. This is the property we encourage by minimizing the ℓ_1 norm of the wavelet coefficients in our objective function. However, the addition of noise to the input signal, which is by nature nonstructured, would lead to a random activation of the filters, independent of the chosen filters. This would, therefore, necessarily lead to a nonsparse decomposition. The sparsity of the decomposition is, thus, sought only once the noise has been canceled. This problem is usually tackled under the assumption that noise would lead to small activation of the filters. As a consequence, the impact of noise on the decomposition can be removed by hard thresholding the obtained coefficients. In ref. 31, guarantees are provided for recovering and denoising signals observed in Gaussian noise by applying the right hard-thresholding operation. Yet, finding the right thresholding parameters is a difficult task and usually depends on the use case and the specific dataset.

In this paper, we propose to make the thresholding step part of our architecture to learn the best thresholding parameters and to

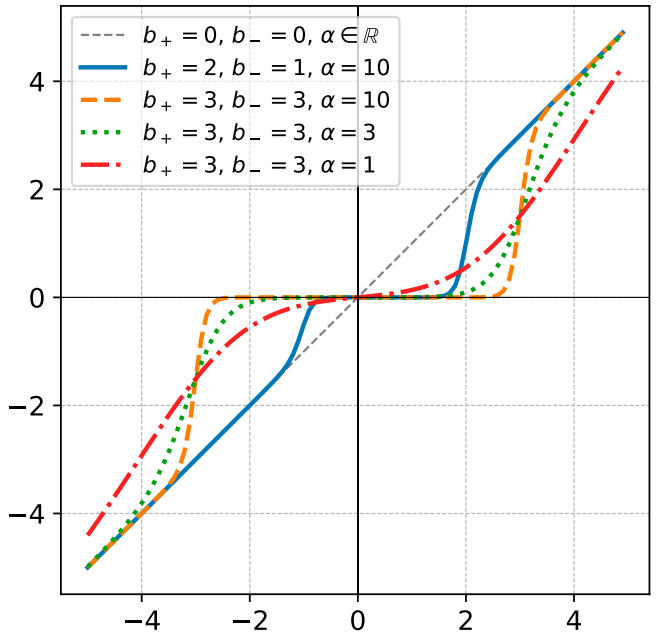

Fig. 2. Hard-thresholding activation function. Proposed activation function performs an operation close to an asymmetric hard thresholding. Both thresholds, in the negative and positive half-space, are learnable parameters. In our paper, α is set to 10 to simulate a "sharp" thresholding.

remove the need of handling it as a separate step. We introduce a learnable hard-thresholding activation function as a combination of two opposite sigmoid functions:

$$\forall x \in \mathbb{R}, \quad HT(x) = x \left[\frac{1}{1 + \exp\left(\alpha \cdot (x + b_-)\right)} + \frac{1}{1 + \exp\left(-\alpha \cdot (x - b_+)\right)} \right], \quad [4]$$

where α is a "sharpness" factor arbitrarily fixed to 10 in this paper, and b_+ and b_- are the positive and negative learnable biases acting as the thresholds on both sides of the origin, as illustrated in Fig. 2.

Denoting the sigmoid function $1/(1 + \exp(-x))$ by σ, Eq. **4** becomes

$$\forall x \in \mathbb{R}, \quad HT(x) = x \cdot \left[\sigma\left(-\alpha \cdot (x + b_-)\right) + \sigma\left(\alpha \cdot (x - b_+)\right)\right]. \quad [5]$$

To replicate the original FDWT without denoising, one can fix in this layer b_+ and b_- to zero, enforcing, thereby, a linear activation.

3. Comparison between Traditional FDWT and DeSpaWN

A. Machine Learning for Sound Data. We focus our experiments specifically on sound data as they are a fast-growing field in industrial applications (4, 32). Sound-based analysis has raised the interest of increasing numbers of companies for several reasons: Experts and machine operators already listen to the machines and are able to tell when the operation is not nominal. It should, therefore, be possible to train a machine-learning approach for continuous monitoring. Moreover, industrial processes are inherently noisy. Thus, it is expected that their sound contains information on their process state. Since the sound is by nature multiscale, it might also allow for the monitoring of several systems at once. Monitoring industrial machines with sound is also relatively simple and cheap. Hence, it is an attractive and scalable solution. Microphones are easy to install or retrofit. They are not intrusive. Hardware and software are readily available. Yet, sound data come with their own challenges such as high noise levels due to the machines usually operating in factory environments and, hence, in noisy environments. Besides, finding how to relate specific industrial processes to the recorded sound is a difficult task. Automatic and noise-robust handling of sound data is, therefore, of high interest for many industrial applications.

B. Classification and Anomaly Detection. The objective of DeSpaWN is learning of a robust autoencoder. This robustness is achieved through denoising with the hard thresholding of the learned coefficients and through sparsity, forcing the network to learn the most meaningful wavelet to describe the training data. Thus, we state that once trained, the autoencoder should produce for similar signals a similarly sparse latent space and achieve similarly low reconstruction residuals. We use these properties to design our classification and our anomaly detection strategies.

For classification, we propose an approach similar to dictionary learning, which consists of training one autoencoder per class and assigning to a sample the class corresponding to the autoencoder that led to the minimal loss of the objective function in Eq. **2**. This classification approach is common in signal processing when signal models are learned or trained to capture very specific properties of a class (17, 33).

For anomaly detection, we state that anomalies can be broadly separated into two types: local intermittent anomalies in the signal (an abnormal pulse due to an impact, e.g., a broken tooth in a gear box) and trend anomalies, when the signal behavior changes more globally (e.g., a change of frequency due to increased friction). When dealing with long signals (several seconds to minutes), capturing trend anomalies is usually achieved by looking at global indicators, such as the network objective function. However, such approaches tend to hide local anomalies, which are averaged out when the signal length increases. One possible solution can be shortening of the input time series to mitigate the impact of the average. However, this solution is not always practical. Here, we propose instead to jointly use averaged and local statistics. More precisely, per time series we propose to use the average residual (*Res*), the maximum residual (*MaxRes*), the average sparsity loss (average ℓ_1 norm of the coefficients per level, $\{\hat{\ell}_1^l\}_{l\in[1..L]}$), and the maximum sparsity loss ($\{\bar{\ell}_1^l\}_{l\in[1..L]}$). We then apply a one-class classifier on this new latent space, $Res \circ MaxRes \circ \{\hat{\ell}_1^l\} \circ \{\bar{\ell}_1^l\}_{l\in[1..L]}$, of size $(2 + 2 \cdot L)$.

In this work, we achieved similar results using the one-class isolation forest, the one-class support vector machine (SVM), an elliptic envelope, or a one-class extreme-learning machine (ELM) (34). We, therefore, report only results using the latter, an ELM with 50 neurons.

C. Datasets. We test the proposed approach on two open-source sound datasets. First, we test the model on an anomaly detection task of sound data of industrial machines with the sound dataset for malfunctioning industrial machine investigation and inspection (MIMII) (35). Second, we demonstrate that the network can learn decomposition specific to its training data by solving a dictionary learning classification task. We show that the approach performs equally well independently of the type of sound used as input to perform classification on the bird song dataset, as proposed in ref. 17. Finally, we show the consistency of the results by using this same dataset for anomaly detection. For both datasets, we analyze the obtained latent space and demonstrate that it can also be used to interpret the data at hand.

C.1. Malfunctioning industrial machine detection. The MIMII dataset (35) consists of audio recordings of four types of industrial machines, i.e., valves, pumps, fans, and slide rails, in normal and malfunctioning states. It is, therefore, a good benchmark for testing anomaly detection approaches on sound data.

The dataset has four individual machines of four machine types. For each machine, sound from normal and abnormal

Table 1. Characteristics of the MIMII dataset

Type	ID	Normal	Abnormal	Operating conditions and type of anomalies
Valve	0	991	119	Open/close repeat with
	2	708	120	different timing. More
	4	1,000	120	than two kinds of
	6	992	120	contamination.
Pump	0	1,006	143	Suction from/discharge to a
	2	1,005	111	water pool. Leakage,
	4	702	100	contamination, clogging,
	6	1,036	102	etc.
Fan	0	1,011	407	Normal operation.
	2	1,016	359	Unbalanced, voltage
	4	1,033	348	change, clogging, etc.
	6	1,015	361	
Slide rail	0	1,068	356	Slide repeat at different
	2	1,068	267	speeds. Rail damage,
	4	534	178	loose belt, no grease, etc.
	6	534	89	

operating conditions has been recorded without further label on the operating state or on the faults, making the dataset very suitable for unsupervised anomaly detection. Each machine has been recorded under three different signal-to-noise ratio (SNR) setups (0, 6, and –6 dB), where the noise denotes background noise of other industrial processes. This results in 48 experiments on which anomaly detection can be performed. It is interesting to note that various anomaly types are collected and that several anomalies can influence the same machine in different recording samples. Table 1 gives an overview of the dataset, presenting the number of samples for each machine and the conditions of operation.

The data were recorded with an eight-channel microphone array, at 16 kHz and 16 bits resolution. Each sample is 10 s long, or 160,000 timestamps. With this, we can set L to 17. In accordance with the work in ref. 35, the data recorded by the first microphone only are used.

C.2. Bird song dataset. For the second experiment, we use a different type of sound data to demonstrate the performance of the proposed framework in a different context, especially since industrial machines often make repetitive noise that is rather easier to characterize. The recordings of bird songs in their natural environment are also subject to environmental noise and differences in the recording hardware that influence the recorded sound. Since the data are labeled (contrary to the machine sound data), it allows us to test the proposed architecture both in an anomaly detection and in a classification setup.

The Xeno-canto Foundation collection bird song dataset (36) is a dataset of bird songs from all around the world that are collected by a large variety of participants. In ref. 17, the author proposes to focus on the following birds: corn bunting (CB), Eurasian skylark (ES), barn swallow (BS), sedge warbler (SW), and common nightingale (CN). These species were selected under the argument that all their recordings were recorded by the same person, implying similar recording conditions and probably similar and consistent hardware. This allows removing the hypothesis that detected fluctuations between recordings and bird species would be due to a change of recording hardware.

For each of the above species, three recordings of about 5 min are available. To establish a fair comparison, we apply the exact same preprocessing as in ref. 17: decimation of the signals by a factor of 4 since most of the signal energy is below 5 kHz and the original sampling rate is 41 kHz. The recordings are split into a collection of signals with 2^{18} samples ($\approx$ 24 s). This leads to the following number of signals:

- CB, 30 signals;
- ES, 24 signals;
- BS, 21 signals;
- SW, 20 signals;
- CN, 20 signals.

As in ref. 17, a fivefold cross-validation is used, meaning that 80% of the data are used for training and 20% for testing at each fold.

D. Ablation Study.

D.1. From FDWT to DeSpaWN. In this section, we aim to analyze how the different contributions impact the results compared to the traditional case where the coefficients from the FDWT would be used as inputs to subsequent machine-learning tasks, such as classification or anomaly detection. Thus, in this first evaluation, to demonstrate how all the steps of the transition from the traditional FDWT to our proposed framework impact the results, we compare the results for the following architectures:

- *db4* : Using the Daubechies-4 (db4) wavelets.
- *db4*+HT: Using the db4 wavelets with learnable hard thresholding (HT) of the coefficients.
- CWN (CQF wavelet network): Learning a single kernel h^0; using CQF to fix the other three g^0, $\bar{h}^0$, and $\bar{g}^0$; and using these kernels for all levels.
- LCWN (layer-wise CQF): Learning one kernel h^l per level in L, using CQF to fix the others.
- DeCWN (denoising CQF): Learning a single kernel h^0, using CQF to fix the other for all levels, and using the learnable hard-thresholding activation function.
- DeSpaWN: Learning one kernel h^l per level in L, using CQF to fix the others, and using the learnable hard-thresholding activation function.
- DeSpaWN-2: Learning two kernels, h^l and g^l, per level in L, using CQF to fix the others, and using the learnable hard-thresholding activation function.
- FreeWN: Learning all kernels of all levels independently and using the learnable hard-thresholding activation function.

For all experiments, we set L, the number of decomposition levels to the nearest second logarithm of the length of the time series. We set the kernel size to 8 and compare the results with those achieved using Daubechies db4 wavelets (since they also have eight coefficients) and with the baseline results on these datasets. In Eq. **2**, we set γ, the weight on the sparsity term in the loss arbitrarily to one. The architecture has, therefore, $(8+2)\cdot L$ learnable parameters (eight kernel coefficients, two thresholds).

For reference, we also report results from the baseline models [MIMII (35) and bird song (17)].

D.2. Results and discussion. From the comparative results presented in Table 2, it appears that the results are consistent between the three machine-learning tasks. Consequently, the impact of the different parameters and assumptions can be discussed at the general level.

DeSpaWN outperforms the baseline. On all tasks, the proposed architecture DeSpaWN significantly outperforms the baseline models found in the literature. Particularly on the MIMII dataset, it reaches globally a performance improvement of 16%. Also, compared to the baseline, DeSpaWN is much less impacted by the noise level; when the SNR changes from 6 to 0 dB, DeSpaWN experiences a drop of performance of 4% while the baseline has a 9% drop. When the noise level increases to a SNR at –6 dB, DeSpaWN performance diminishes while remaining well above the baseline (+16%). This suggests that DeSpaWN is learning a noise-independent representation of the signals, which makes it much more robust to noise than other approaches. This is a

Table 2. Comparative study on three machine-learning tasks: For the different architecture variations (one per column), comparative results on the three considered tasks

	DeSpaWN										
	$\gamma = 1$	$\gamma = 0.5$	$\gamma = 5$	*db4*	*db4*+HT	CWN	LCWN	DeCWN	DeSpaWN-2	FreeWN	Baseline
Anomaly Detection on MIMII											
Valve	92.8	92.7	91.0	92.7	92.7	92.7	92.7	92.9	**93.0**	**93.0**	61.3
Pump	**84.5**	82.0	72.6	77.9	78.3	78.2	78.1	78.0	84.2	75.8	72.3
Fan	**86.2**	84.8	84.9	84.6	84.7	84.9	84.8	85.3	85.5	83.8	79.0
Slider	**91.0**	89.4	78.7	89.8	90.0	89.4	89.6	89.7	90.1	89.3	78.6
6 dB	**94.6**	93.5	84.4	93.4	93.4	93.7	93.5	93.5	94.3	90.7	81.6
0 dB	**90.5**	87.7	81.9	82.6	88.1	87.7	87.8	88.5	90.2	87.0	72.3
-6 dB	80.8	79.6	76.5	77.6	77.9	77.5	77.5	77.5	80.0	78.8	64.4
Avg.	**88.6**	87.1	81.4	85.5	86.4	86.3	86.3	86.5	88.2	85.5	72.8
Anomaly detection (1 versus 4 bird species)											
Avg.	**99.8**	**99.8**	91.7	95.4	98.2	97.5	98.8	99.0	99.5	99.1	N.A.
#C: Classification with dictionary learning (bird song)											
2	**99.2**	98.2	91.3	50.0	87.0	73.3	92.7	93.4	99.1	89.0	97.2
3	**98.3**	96.3	88.7	33.3	77.3	55.1	86.0	87.4	97.5	78.0	88.0
4	**97.5**	94.5	87.0	25.0	70.0	43.5	80.0	81.9	97.3	67.0	74.7
5	**96.7**	92.7	85.3	20.0	64.0	37.0	74.7	77.0	95.6	56.0	70.4
Avg.	**97.9**	95.4	88.1	32.1	74.6	52.2	83.3	84.9	97.4	72.5	82.6

For the anomaly detection tasks, on MIMII and on the bird song, we report the average AUC (%). Finally, for the bird song classification by dictionary learning, we report the average accuracy (%). N.A., not applicable. Bold indicates best model for each experimental setup.

particularly important requirement in real applications that are typically impacted by different types of noise at different levels. Finally, in the baseline, the spectrogram with logarithmic mel scale (log-mel) is extracted to be used as input to an autoencoder. With DeSpaWN, the raw data are used directly, without requiring any preprocessing steps. This lightens the methodology significantly since extracting a spectrogram requires the choice of several hyperparameters such as the window type, the window length, the window stride, whether to compute the density or the magnitude, and whether to apply additional transformations (log-mel, decibels, etc.). All these choices can influence the results significantly.

Impact of the sparsity coefficient. From the first three columns of Table 2, it appears that the sparsity term in the loss of DeSpaWN (Eq. **2**) influences the results. In addition, setting γ to one, without fine-tuning, always seems to be a near optimal choice. This can be explained by our definition of the loss as the average of the ℓ_1 values, first of the residuals, and second of the coefficients moduli. Using the average makes them comparable. Using smaller γ influences the results slightly. However, increasing γ can have quite a strong impact on the performance. This signifies that even though sparsity is helping to get some robustness to external factors, too much of it would be at the expense of the reconstruction loss and at the expense of the ability to distinguish variations in the signals, including anomalies or class-specific coefficient behaviors.

Impact of hard-threshold learning. The second noteworthy observation is that the architectures without hard threshold are performing significantly worse compared to others (*db4* versus *db4*+HT or LCWN versus DeSpaWN). This highlights the importance of the denoising part of the architectures. The strength of the architecture is its ability to learn the best thresholds for the wavelet coefficients to become robust to small variations in the signal. This strength is independent of whether the wavelets are learned or not.

Impact of wavelet learning. For the classification task in particular, learning the right wavelet is pivotal for the architecture accuracy. This is to be expected since the class attribution is done based on the network loss minimization. When fixing the wavelets or even both the wavelets and the thresholding function (*db4* and *db4*+HT), there are not many parameters left to optimize. The architectures become generic and not fine-tuned per class, making the class attribution based on the loss not much better than a random guess (*db4* has random guess attribution since all architectures are identical). For anomaly detection, this effect is mitigated by the property of the wavelets: Due to the CQF property, wavelets are designed for good reconstruction and relative sparsity (wavelets are used in signal processing since they inherently tend to produce sparse signal representations). Hence, they are already good candidates to create relevant signal description and, thus, anomaly detection. Yet, learning the right wavelets still brings some nonnegligible improvements (additional +1 or 2%, averaged over several tens of experiments).

Impact of the CQF. The impact of constraining the wavelet basis can be observed by comparing *db4*+HT (fixed Daubechies-4 wavelets), the DeCWN (learning of one global wavelet), the DeSpaWN (learning one wavelet per layer), the DeSpaWN-2 (learning one wavelet and one scaling function per layer), and the FreeWN (learning all kernels). As expected, constraining the kernels tends to make the network less specific to the characteristics of the training class and affects the classification performance strongly. The most constrained architecture (*db4*+HT) has the worst results (52%); DeCWN performs better (84.9%) but not as good as DeSpaWN or DeSpaWN-2 (97.9 and 97.4%). These two architectures are in fact quite equivalent in terms of results. Leaving the kernel completely free (FreeWN) also leads to a drop in performance. This is likely due to training instabilities as explained in *2. Learnable Denoising Sparse Wavelet Network*.

E. Additional Diagnostics Potential.

E.1. Insights on MIMII. The good detection performance of DeSpaWN indicates again that the signals are probably forming well-defined clusters in the $Res \circ MaxRes \circ \{\hat{\ell}_1^l\} \circ \{\bar{\ell}_1^l\}_{l\in[1..L]}$ latent space. Anomalies would then appear outside of these clusters and finding anomaly clusters could help to diagnose the different conditions of the system. This can be visualized in two dimensions, by performing a t-distributed stochastic neighbor

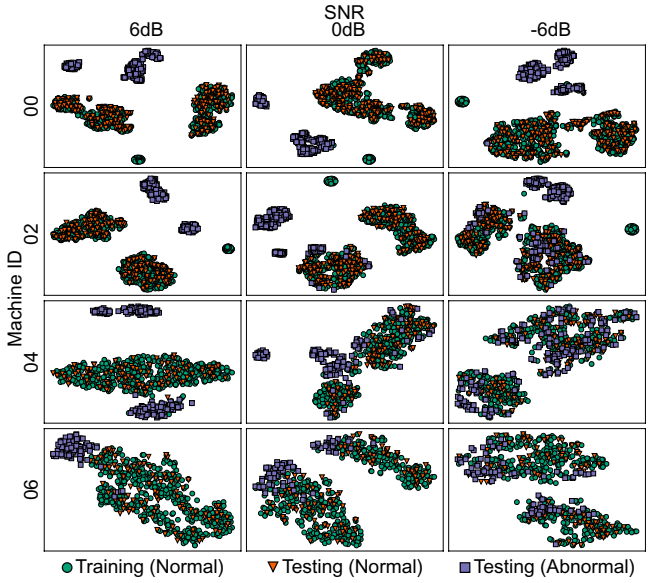

Fig. 3. t-SNE of the slide rail latent space. Shown is representation in two dimensions of the slide rail latent space for the four machines (one machine per row), at the three different experimental SNRs (columns), using t-SNE (perplexity of 30). One can distinguish different clusters, most likely representing different operating conditions and anomalies.

embedding (t-SNE) on the latent space, as illustrated in Fig. 3. In Fig. 3, the latent spaces of the different slide rail experiments are shown after a t-SNE transformation with the default perplexity of 30. Clusters can be clearly identified in this representation. They are likely to be formed by different anomaly types and operating conditions. In all experiments, one can distinguish at least two normal operating condition clusters, indicating different conditions of operation and at least two anomaly clusters, well separated from the normal conditions. With this representation, only when the SNR decreases to –6 dB, some of the anomalous points become less separated from the normal conditions. This decrease in separability could be expected from the lower area under the curve (AUC) 4 as reported in Table 2.

The diagnostics possibilities offered by extracting characteristic patterns of the learned coefficients of the proposed approach for the different clusters analysis are illustrated for the slide rail 0 at 0 dB SNR in Fig. 4. Two different signals extracted from each of the two normal measurement clusters are shown. In Fig. 4, the raw signals, their log-spectrogram, and the distribution of the learned coefficients over the 18 levels are shown. The first signal has its major components around the 10th level with some quite large coefficients at the highest level of detail, while the other signal is mostly "active" at the 12th level, with very little activation of the last two levels (highest level of details). This indicates that the operating mode has likely changed between these two samples: The main information content changed from level 10 to 12, that is, a factor of 4 in the spectrum of the original signal. Similarly, in Fig. 5, two unhealthy signals of the slider rail drawn from two different clusters are depicted. The first signal has its 12th level more activated than healthy signals; the other signal distinguishes itself with its much larger high-level coefficients. These are likely two different types of anomalies.

Finally, Fig. 6 shows exemplarily the learned filters and hard-threshold coefficients for a slider rail. The first layers, corresponding to the high frequencies, have high hard-thresholding values (up to 0.3 in absolute value). One can observe that the corresponding wavelet coefficients in Fig. 4 are almost all zero. This makes the filter of these layers irrelevant and this further explains why the corresponding filters observed in Fig. 6 are very close to the original Daubechie filters. It is probably where most of the noise is concentrated. For lower frequencies, the filters are farther away from the Daubechie wavelets and the hard thresholding is much lower. It is probably at these scales that the information required for the reconstruction is concentrated. This interpretation matches the coefficient distributions observed in Fig. 4. This gives a strong indication that the proposed architecture can indeed selectively filter and threshold the layers based on their information content.

E.2. Insights on the bird song dataset. Fig. 7 illustrates the $l1$ -residual space for all birds for the different cases. For each plot, DeSpaWN is trained on the main bird class where all other bird samples would have to be detected as anomalies. In each case, one can observe that the samples corresponding to the bird used for training form a well-defined and separated cluster, allowing the one-class classifier to identify easily all other birds

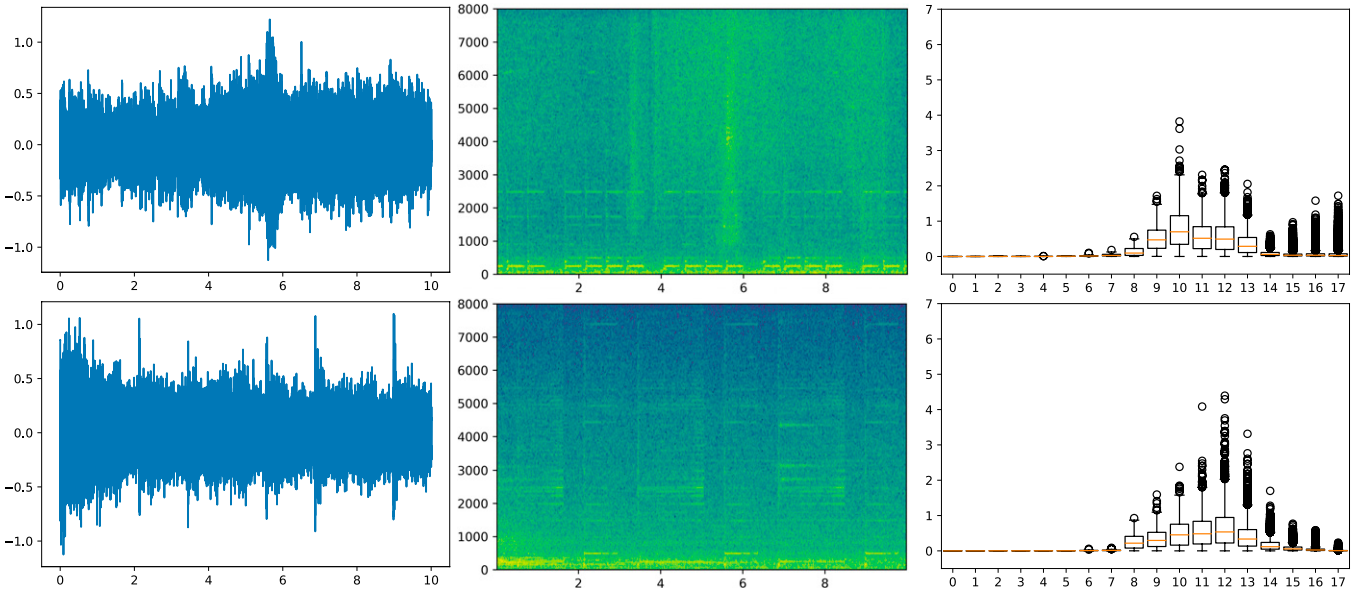

Fig. 4. Examples of normal signals. Shown are raw data, log-spectrogram, and obtained coefficient distribution per level for two normal measurements of the slider rail 0 at SNR 0 dB.

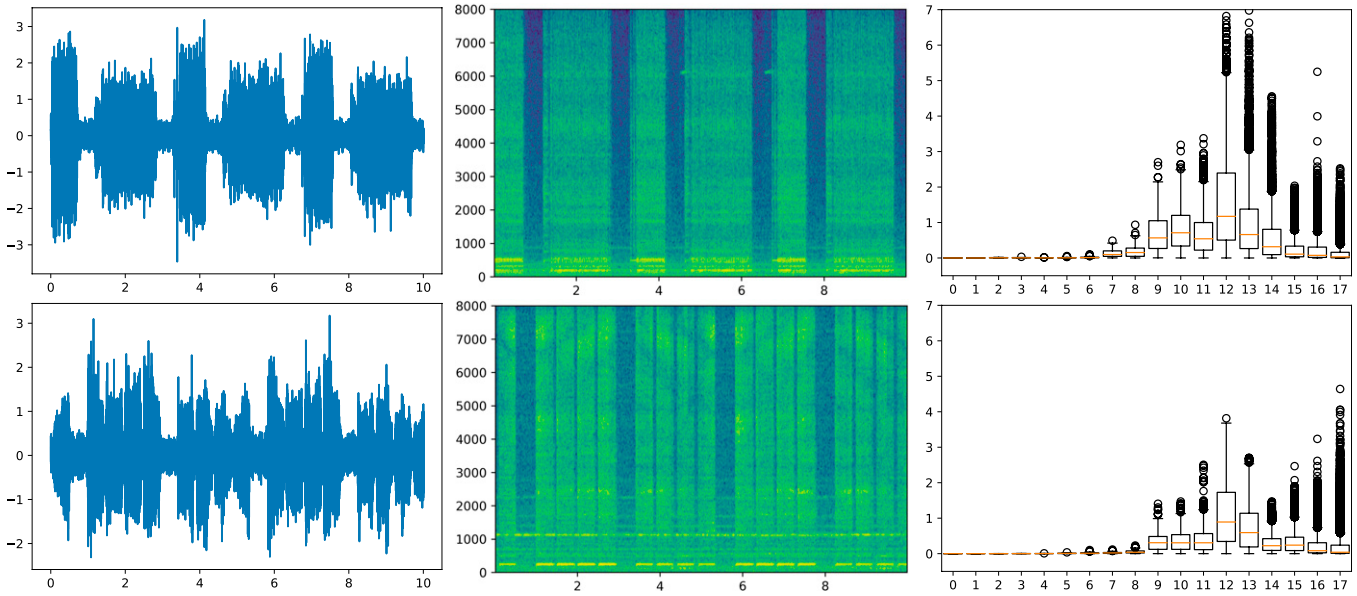

Fig. 5. Examples of abnormal signals. Shown are raw data, log-spectrogram, and obtained coefficient distribution per level for two abnormal measurements of the slider rail 0 at SNR 0 dB.

as anomalies. The only exception is the corn bunting case (Fig. 7, *Right*), which is mixed with the sedge warbler after training, particularly with this specific representation. These results explain the classification accuracy drop observed when adding the sedge warbler as the fifth species (Table 2).

4. Comparison to Other Frameworks

A. Scattering Transform, U-Net, and Autoencoders. In this section, we propose to compare the results of DeSpaWN to other state-of-the art frameworks found in the literature: the scattering transform, vanilla convolutional autoencoders (CAE), and U-Net.

First, we propose to compare to the scattering transform (21), which has been extensively used in the context of audio signal processing. It is a signal representation that is stable to small deviations in its inputs and is able to characterize transient phenomena like amplitude modulation. We use the Kymatio library (37) to compute the coefficients from a two-layer scattering transform. It requires the selection of two parameters (38): J the maximum scale of the filters used, implying that the transform will capture only frequencies superior to 2^J, and Q the number of filters per octave. We propose to analyze two combinations of parameters: 1) $J = 17$ and $Q = 1$, which will result, for the first layer, in a decomposition close to the FDWT with one filter per octave and able to characterize low frequencies up to 2^{-17}, and 2) $J = 10$ and $Q = 8$ for the first layer (which defines wavelets having the same frequency resolution as mel-frequency filters) and $Q = 1$ for the second layer. The second choice of parameters is motivated by previous research that proposed to consider mel-spectrogram features on frames of around 60 ms (35). Similar to the approach used with DeSpaWN, for the anomaly detection task, we use a one-class ELM on the scattering coefficients. For the bird song classification task, since the scattering transform does not create a signal-specific decomposition, the dictionary learning approach cannot be mimicked. Therefore, we use, for all approaches, the coefficients as input to an SVM classifier to make the results comparable.

In addition, to highlight the relevance of the proposed architectural choice of DesPaWN, we compare it numerically to standard CNN autoencoders (CAE) and CNN U-Nets. The considered autoencoders (AE) are based on the work of ref. 39. The architecture has been used for fault diagnosis of rotating machinery. We consider four encoding and decoding layers with eight coefficients per kernel and a kernel size of N_{AE} for each layer. The impact of the addition of trainable parameters is studied by considering the range of $N_{AE} = [4, 8, 16, 32]$. DeSpaWN architecture has skip-connections at each level. Therefore, it can be considered as a special case of a U-Net model. We then compare our method to another U-Net architecture based on ref. 22 that was applied to electrocardiogram detection. We replace the concatenation of the skipped connection with the addition to

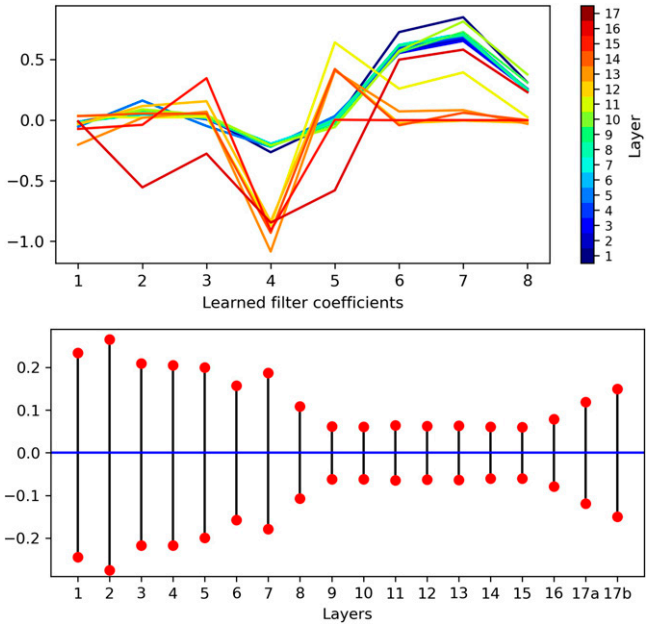

Fig. 6. Learned coefficients and biases. (*Top*) Learned kernel for all layers of a slider rail. Each color represents the filter from a different layer, from the first (high-frequency) layer in dark blue to the last layers in red. (*Bottom*) Learned positive and negative biases for each layer. The black lines represent the range of values that are set to zero by the corresponding hard-thresholding layer.

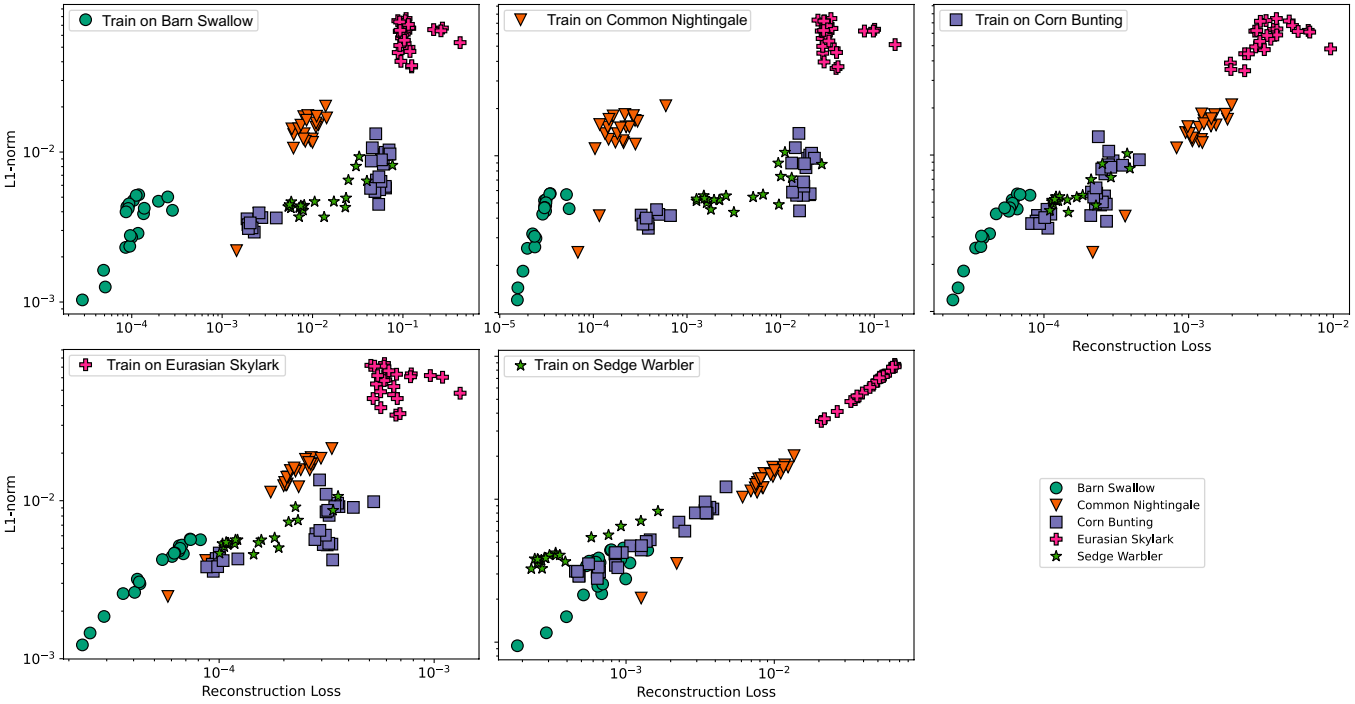

Fig. 7. *Res* $\circ \hat{\ell}_1$ space. For each bird, DeSpaWN is trained and all birds are tested and plotted together. When DeSpaWN is trained on one species, the species tends to be well separated from others and has generally a more compact representation in the *Res* $\circ \hat{\ell}_1$ than when tested with another DeSpaWN.

have an architecture closer to DesPaWN. Furthermore, we consider $L = 17$ layers with a stride of 2 for each CNN layer and eight coefficients per kernel. The kernel size N_{Unet} of each filter is studied in the range $N_{Unet} = [1, 2, 4, 8]$. This grid was chosen since larger kernel sizes led to decreasing performances. The main differences between our model and this U-Net architecture are one more filter at each skip connection, initialization of each pair of filters as high-pass and low-pass, and replacing the rectified linear unit (ReLU) activation function with the proposed learnable hard-thresholding function. For each method, the same loss function as in Eq. **2** is used for training. The exact same process is followed for anomaly detection and for classification as for the results achieved with DeSpaWN (cf. *B. Results of the Benchmark*).

B. Results of the Benchmark. The results of the benchmark are presented in Table 3, for two versions of the scattering transform, four variations of the CAE, and four variations of the U-Net. DeSpaWN and the scattering transform both provide very competitive results and are both very solid candidates to solve the tasks on the dataset studied here. One can note, however, a bigger drop in performance for the scattering transform when the noise in the data increases (–20% and –16% when comparing MIMII 6 dB with –6 dB). It is also worth noting that the strength of each approach depends on the machine type, where the scattering transform seems to be the most adapted to tackle the pump and the slider while DeSpaWN performs better on the valve and on the fan. For the bird song classification, very similar results are achieved.

The autoencoders, both the traditional CAE and the U-Nets, provide very competitive results for the valve system but not for the other machines and are overall significantly worse performing compared to the scattering transform and to DeSpaWN. It is worth noting, however, that all approaches outperform the reported baseline on these datasets.

5. Conclusion

In this paper, we proposed an architecture for learning a meaningful and sparse representation of high-frequency signals in an unsupervised manner without requiring neither preprocessing (feature extraction) nor postprocessing (e.g., denoising). This architecture achieves very good results that are well above the baselines and are competitive compared to other state-of-the-art approaches on three machine-learning tasks for anomaly detection and classification. We designed an end-to-end deep-learning architecture, mimicking the cascade algorithm of the FDWT but making it fully learnable. Using the deep-learning framework, we demonstrated the benefits of learning the right wavelets at each level of the decomposition. One of the additional contributions is the introduction of a learnable hard-thresholding function for automatic signal denoising.

The proposed methodology combines 1) a thorough theoretical foundation on the wavelet properties, including cascade, perfect reconstruction, and antialiasing filter basis with the CQF property, and denoising with coefficient thresholding with 2) the learning ability of deep learning. The proposed architecture could demonstrate a significant improvement on sound data processing, both for classification and for anomaly detection tasks. Our approach allows the use of the raw HF data as input to a deep-learning architecture, a setup usually avoided in the literature due to the difficulty of designing efficient architectures that are robust to changes in the input lengths. The proposed architecture takes root in spectral analysis and can replace the usual preprocessing steps such as spectrogram or wavelet coefficient extraction. Since it is unsupervised, it can be used as an input to subsequent learning methods. In addition, compared to other deep-learning architectures, it is a very light framework with only a few hundred learnable parameters, mitigating in that way the high risk of overfitting. With its spectral interpretation, it also provides diagnostics information to the domain experts that can potentially improve the interpretation capabilities.

This work opens several doors for future directions. First, given the high information content of the proposed latent space, other unsupervised machine-learning tasks could be explored such as system degradation monitoring; e.g., a drop in the sparsity of the decomposition could be a sign of an increased signal complexity

Table 3. Comparative study on three machine-learning tasks: For the different methodologies (one per column), comparative results on the three considered tasks

	DeSpaWN	Scattering transform		Auto Encoder				U-Net			
	γ 1	(J, Q) (17, 1)	(J, Q) (10, 8)	N_{AE} 4	N_{AE} 8	N_{AE} 16	N_{AE} 32	N_{Unet} 1	N_{Unet} 2	N_{Unet} 4	N_{Unet} 8
Anomaly detection on MIMII											
Valve	92.8	65.5	78.6	91.5	92.4	91.5	92.0	92.3	92.7	**93.0**	91.1
Pump	84.5	88.5	**90.6**	74.5	74.5	74.0	73.4	75.4	69.6	70.5	74.4
Fan	86.2	**88.7**	84.9	71.0	76.6	80.7	76.1	77.1	75.6	76.5	73.6
Slider	91.0	86.9	**96.8**	79.2	81.0	81.7	78.1	78.6	80.6	83	87.4
6 dB	94.6	91.2	**95.4**	83.4	87.5	86.6	85.0	86.3	84.0	84.7	85.2
0 dB	**90.5**	84	88.6	79.7	81.3	81.5	80.8	82.5	79.8	83.0	83.8
–6 dB	**80.8**	72	79.1	74.2	74.6	73.9	74.0	74.0	75.0	74.0	75.0
Avg.	**88.6**	82.4	87.7	79.1	81.15	80.3	80.0	81.0	79.6	80.9	81.4
Anomaly detection (1 versus 4 bird species)											
Avg.	**98.6**	93.8	94.0	85.4	86.1	86.3	86.3	89.6	88.9	89.2	90.0
#C: Classification with SVM (bird song)											
5	97.7	**97.9**	97.3	87.6	92.2	92.3	90.2	83.4	85.7	87.1	88.4

For the anomaly detection tasks, on MIMII and on the bird song, we report the average AUC (%). Bold indicates best model for each experimental setup.

or of the presence of disturbing components due to system wear. The architecture could be further analyzed in conditions with controlled noise and signals to better understand its denoising and stability properties. The architecture could also be extended, such as in particular with the imbrication of this architecture in stacked architecture to solve supervised machine-learning tasks. In these cases, the learned wavelets and thresholding coefficients could be learned not only for sparsity and for reconstruction but also to maximize a supervised objective. The use of parallel wavelets (number of kernels in a convolution layer) and the handling of multichannel inputs are further exciting potential developments.

Data and Code Availability. All study data are included in the main text. Our code is available at GitHub, https://github.com/MichauGabriel/DeSpaWN. Previously published data were used for this work (https://zenodo.org/record/3384388#.YG1g150zaUk https://archive.org/details/xccoverbl_2014).

ACKNOWLEDGMENTS. This work was supported by the Swiss National Science Foundation Grant PP00P2-176878 and by the Innosuisse Grant 27662.1 PFES-ES. We thank Christoph Preisinger for his preliminary explorations of the proposed methodology.

1. Z. Peng, F. Chu, Application of the wavelet transform in machine condition monitoring and fault diagnostics: A review with bibliography. *Mech. Syst. Signal Process.* **18**, 199–221 (2004).
2. P. Gangsar, R. Tiwari, Signal based condition monitoring techniques for fault detection and diagnosis of induction motors: A state-of-the-art review. *Mech. Syst. Signal Process.* **144**, 106908 (2020).
3. Q. Wang, G. Michau, O. Fink, Missing-class-robust domain adaptation by unilateral alignment. *IEEE Trans. Ind. Electron.* **68**, 663–671 (2020).
4. Z. Li, Y. Wang, K. Wang, A deep learning driven method for fault classification and degradation assessment in mechanical equipment. *Comput. Ind.* **104**, 1–10 (2019).
5. H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, P. A. Muller, Deep learning for time series classification: A review. *Data Min. Knowl. Discov.* **33**, 917–963 (2019).
6. H. Wang, Q. Zhang, J. Wu, S. Pan, Y. Chen, Time series feature learning with labeled and unlabeled data. *Pattern Recognit.* **89**, 55–66 (2019).
7. O. Fink *et al.*, Potential, challenges and future directions for deep learning in prognostics and health management applications. *Eng. Appl. Artif. Intell.* **92**, 103678 (2020).
8. G. Michau, O. Fink, Unsupervised transfer learning for anomaly detection: Application to complementary operating condition transfer. *Knowl. Base. Syst.* **216**, 106816 (2021).
9. S. Kiranyaz *et al.*, 1D convolutional neural networks and applications: A survey. *Mech. Syst. Signal Process.* **151**, 107398 (2021).
10. W. Zhang, C. Li, G. Peng, Y. Chen, Z. Zhang, A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load. *Mech. Syst. Signal Process.* **100**, 439–453 (2018).
11. V. Papyan, Y. Romano, M. Elad, Convolutional neural networks analyzed via convolutional sparse coding. *J. Mach. Learn. Res.* **18**, 2887–2938 (2017).
12. S. Liu, "Fourier neural network for machine learning" in *2013 International Conference on Machine Learning and Cybernetics* (IEEE, 2013) vol. 1, pp. 285–290.
13. M. Uteuliyeva *et al.*, Fourier neural networks: A comparative study. *Intell. Data Anal.* **24**, 1107–1120 (2020).
14. S. Luan, C. Chen, B. Zhang, J. Han, J. Liu, Gabor convolutional networks. *IEEE Trans. Image Process.* **27**, 4357–4366 (2018).
15. T. Li *et al.*, Waveletkernelnet : An interpretable deep neural network for industrial intelligent diagnosis. *IEEE Trans. Syst. Man Cybern. Syst.*, 10.1109/TSMC.2020.3048950 (2021).
16. J. Wang, Z. Wang, J. Li, J. Wu, "Multilevel wavelet decomposition network for interpretable time series analysis" in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (ACM, New York, 2018), pp. 2437–2446.
17. D. Recoskie, "Learning sparse orthogonal wavelet filters," PhD thesis, University of Waterloo, Waterloo, ON, Canada (2018).
18. P. Liu, H. Zhang, W. Lian, W. Zuo, Multi-level wavelet convolutional neural networks. *IEEE Access* **7**, 74973–74985 (2019).
19. M. Khalil *et al.*, An end-to-end multi-level wavelet convolutional neural networks for heart diseases diagnosis. *Neurocomputing* **417**, 187–201 (2020).
20. D. L. Donoho, I. M. Johnstone, "Threshold selection for wavelet shrinkage of noisy data" in *Proceedings of 16th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (IEEE, 1994), vol. 1, pp. A24–A25.
21. J. Andén, S. Mallat, Deep scattering spectrum. *IEEE Trans. Signal Process.* **62**, 4114–4128 (2014).
22. G. Jimenez-Perez, A. Alcaine, O. Camara, "U-net architecture for the automatic detection and delineation of the electrocardiogram " in *2019 Computing in Cardiology (CinC)* (IEEE, 2019).
23. S. Mallat, *A Wavelet Tour of Signal Processing, The Sparse Way* (Elsevier Science/Academic Press, 3rd ed., 2009).
24. A. Croisier, "Perfect channel splitting by use of interpolation/decimation/tree decomposition techniques" in *Proceedings of the International Symposium on Information, Circuits and Systems* (Proceedings of the International Symposium on Information Circuits and Systems, Patras, Greece, 1977), pp. 443–446.
25. M. Smith, T. Barnwell, "A procedure for designing exact reconstruction filter banks for tree-structured subband coders" in *ICASSP'84, IEEE International Conference on Acoustics, Speech, and Signal Processing* (IEEE, 1984), vol. 9, pp. 421–424.
26. F. Mintzer, Filters for distortion-free two-band multirate filter banks. *IEEE Trans. Acoust. Speech Signal Process.* **33**, 626–630 (1985).
27. M. Alfaouri, K. Daqrouq, ECG signal denoising by wavelet transform thresholding. *Am. J. Appl. Sci.* **5**, 276–281 (2008).
28. F. M. Bayer, A. J. Kozakevicius, R. J. Cintra, An iterative wavelet threshold for signal denoising. *Signal Processing* **162**, 10–20 (2019).
29. D. L. Donoho, "Nonlinear wavelet methods for recovery of signals, densities, and spectra from indirect and noisy data" in *Proceedings of Symposia in Applied Mathematics* (American Mathematical Society, 1993), pp. 173–205.
30. H. Bolcskei, P. Grohs, G. Kutyniok, P. Petersen, Optimal approximation with sparsely connected deep neural networks. *SIAM J. Math Data Sci.* **1**, 8–45 (2019).
31. D. L. Donoho, J. M. Johnstone, Ideal spatial adaptation by wavelet shrinkage. *Biometrika* **81**, 425–455 (1994).
32. J. Liebetrau, S. Grollmisch, Predictive maintenance with airborne sound analysis. *Process. Mag.* **1**, 15587140 (2017).
33. J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, "Discriminative learned dictionaries for local image analysis" in *2008 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2008), pp. 1–8.
34. G. Michau, Y. Hu, T. Palmé, O. Fink, Feature learning for fault detection in high-dimensional condition monitoring signals. *Proc. Inst. Mech. Eng. O. J. Risk Reliab.* **234**, 104–115 (2020).
35. H. Purohit *et al.*, MIMII dataset: Sound dataset for malfunctioning industrial machine investigation and inspection. arXiv [Preprint] (2019). https://arxiv.org/abs/1909.09347 (Accessed 3 February 2022).
36. X canto Foundation, *Dataset of bird songs* (2004). https://archive.org/details/xccoverbl_2014. Accessed 10 January 2021.
37. M. Andreux *et al.*, Kymatio: Scattering transforms in python. *J. Mach. Learn. Res.* **21**, 1–6 (2020).
38. G. Destouet, C. Dumas, A. Frassati, V. Perrier, "Wavelet scattering transform and ensemble methods for side-channel analysis" in *International Workshop on Constructive Side-Channel Analysis and Secure Design* (Springer, Cham, Switzerland, 2021), pp. 71–89.
39. X. Liu, Q. Zhou, J. Zhao, H. Shen, X. Xiong, Fault diagnosis of rotating machinery under noisy environment conditions based on a 1-d convolutional autoencoder and 1-d convolutional neural network. *Sensors (Basel)* **19**, 972 (2019).