LEARNING-BASED SCATTERING TRANSFORM FOR EXPLAINABLE CLASSIFICATION

1st Mahiout Thomas
Thales DMS
Sophia-Antipolis, France
thomas.mahiout@orange.fr

2nd Fillatre Lionel
Université Côte d'Azur, CNRS, I3S
Sophia-Antipolis, France
lionel.fillatre@i3s.unice.fr

3rd Deruaz-Pepin Laurent
Thales DMS
Sophia-Antipolis, France
Laurent.Deruaz-Pepin@fr.thalesgroup.com

***Abstract*—Vessel noise classification is generally considered as a challenging task due to its need for robustness and reliability. Thus, classification in this domain mainly relied on expert feature. Raw waveform architectures have been historically avoided, despite their performances in other domains. This paper proposes a Learning-based Scattering Transform (LST) that efficiently learns temporal dependencies within cyclostationary signals, such as vessel noises. The LST is implememented as a Convolutional Neural Network (CNN) with short filters whose structure mimics a multiscale signal decomposition. By this way, the architecture of our neural network is intrinsically explainable. Numerical simulations compare our method to an other explainable model and classic convolutional neural networks.**

***Index Terms*—Ship acoustic signal, Bayes detection, CNN, Scattering transform, Explainability**

I. Introduction

Due to a complex environment and substantial levels of ambient noises partially hiding ship signals, vessel noise classification is considered as a rather challenging task compared to speech recognition [1] or environmental sound classification [2]. Fortunately, ship noise signatures present some forms of cyclostationarity or periodicity in second order statistics [3]–[5], while ambient noise is mostly stationary or impulsive. In other acoustics domains, several publications have shown that 1D CNNs [6], [7] or convolutional restricted Boltzmann machines [8] could outperform architectures employing extracted features [9], [10], with a cascade of convolutional layers and non-linear operations applied to raw acoustic signals.

In safety-critical applications such as vessel noise classification, a reliable classification is crucial. Reliability concerns interpreting the reasons behind an algorithm's decision, even if the algorithm is considered as a black box [11]. In image recognition, it is mainly done through various visualization methods such as GradCAM [12] or LIME [13] in order to highlight which feature in the image is responsible for a given prediction. However, interpretability with visualization is not satisfactory for raw acoustic signal where most of the information is accumulated in second-order statistics. Furthermore, considering human user expectations, explainability by design [14] is preferable. In other words, the classification architecture must have a clear mathematical structure with identifiable mappings between the classification steps.

The optimal solution of a classification problem for which we seek to maximize the classification accuracy is the Bayes detector. In [15], the authors proposed a structurally explainable deep network whose structure matches the Bayes detector structure through neural network approximation theory. A notable loss of accuracy with this approximation was due to the excessive size of filters that must be learned to extract long temporal dependencies. Indeed, reports on the impact of convolutional filter sizes suggest that models with larger filters tend to perform worse than small ones [16]. To improve the architecture in [15], while not increasing model complexity to the point of losing understanding of its internal working, we turn our attention to signal processing methods implementable within deep learning framework.

Our contributions are threefold. First, we propose the LST CNN module, a learning-based adaptation of the scattering transform [17] that keeps most of the mathematical structure of the scattering transform. This module creates different subsampled filtered versions of the signal which collectively learn a bank of small convolutional filters. Second, by aggregating all these filtered versions of the input signal, we show that we obtain an interpretable signal representation whose mathematical structure is trainable as a regular CNN. This CNN has significantly less coefficients to estimate that the original CNN proposed in [15]. Third, we compare our approach to usual Fully Convolutional Neural Networks (FCNN). For these experimentations, we exploit both a simplified acoustic model of ship acoustic noise [18] and the real data set ShipsEar [19] of underwater sounds produced by vessels of various types.

This paper is organized as follows. Section II describes the problem of classifying received signal as either ambient noise or ship acoustic signals. Section III presents our explainable first-order LST CNN. Section IV is dedicated to numerical experiments and section V concludes the paper.

Thanks to ANRT agency for funding.

II. Problem Statement

A. Observation model

Let $t \mapsto x(t) \in \mathbb{R}$ be a signal received after sensor array pre-processing. It is the sum of a ship-radiated noise $s_\theta(t)$ with signature parameters θ and an ambient noise $n_a(t)$ [20]:

$$x(t) = s_\theta(t) + n_a(t). \tag{1}$$

The ambient noise $n_a(t)$ is a zero-mean white Gaussian stationary noise with variance σ_a^2. It is independent from $s_\theta(t)$. The ship-radiated noise $s_\theta(t)$ is modeled as a merchant ship propeller zero mean Gaussian noise with the variance denoted $\sigma_\theta^2(t)$. The detailed model is described in [18] but it does not matter in this paper that is focused on the approximation of the optimal Bayes detector with a neural network. Let us assume that $x(t)$ is sampled into K samples $x(k) = x(t_k)$ at some sampling times t_k such that we obtain the vector $\boldsymbol{x} = [x(1), \dots, x(K)]$. From (1), it follows that each sample $x(k)$ follows the Gaussian distribution

$$x(k) \sim \mathcal{N}\left(0, \sigma_\theta^2(k) + \sigma_a^2\right). \tag{2}$$

Let us assume that θ is unknown but it belongs to a finite set of known ship signatures $\Theta = \{\theta_1, \dots, \theta_M\}$. Our classification problem consists in deciding if the signal $\boldsymbol{x}$ contains only ambient noise (hypothesis H_0) or if it contains a ship signal s_θ with $\theta \in \Theta$ (hypothesis H_1):

$$H_0{:}\{x(k) \sim \mathcal{N}\left(0, \sigma_a^2\right),\ k \in [\![K]\!]\}, \tag{3}$$

$$H_1{:}\{x(k) \sim \mathcal{N}\left(0, \sigma_\theta^2(k) + \sigma_a^2\right), \theta \in \Theta, k \in [\![K]\!]\}, \tag{4}$$

where $[\![K]\!] = \{1, \dots, K\}$. It must be noted that hypothesis H_1 is composite [21] since θ is unknown. We consider a data set $\mathcal{S} = \{(\boldsymbol{x}^{(1)}, y^{(1)}), ..., (\boldsymbol{x}^{(N)}, y^{(N)})\}$ where $(\boldsymbol{x}^{(i)}, y^{(i)})$ is composed of a received signal $\boldsymbol{x}^{(i)} \in \mathbb{R}^K$ and its label $y^{(i)} \in \{0, 1\}$ such that $y^{(i)} = j$ means that $\boldsymbol{x}^{(i)}$ follows the hypothesis H_j. The training samples $(\boldsymbol{x}^{(i)}, y^{(i)})$ are independent and follow the mixture distribution $\mathcal{D}(\boldsymbol{x}, y)$:

$$\mathcal{D}(\boldsymbol{x}, y) = q_0 \mathrm{Pr}_0(\boldsymbol{x}) + q_1 \sum_{m=1}^{M} \pi_m \mathrm{Pr}_m(\boldsymbol{x}) \tag{5}$$

where $\mathrm{Pr}_0(\cdot)$, resp. $\mathrm{Pr}_m(\cdot)$, denotes the probability measure as $\boldsymbol{x}$ follows H_0, resp. H_1 with signature θ_m. The probability $q_0 = \Pr(y = 0)$ and $q_1 = \Pr(y = 1) = 1 - q_0$ are the probability of occurrence of H_0 and H_1 respectively, while π_m is the probability to get the signature θ_m when H_1 occurs.

B. Deep neural network optimization

A detection rule for solving (3)-(4) is a function $\delta : \mathbb{R}^K \to \{0, 1\}$ that decides H_i when $\delta(\boldsymbol{x}) = i$. We consider a binary sigmoid decision function $\widehat{\delta}_{\boldsymbol{\mu}} : \mathbb{R}^K \to [0, 1]$ based on a neural network output $\hat{h}_{\boldsymbol{\mu}}(\boldsymbol{x}) : \mathbb{R}^K \to \mathbb{R}$,

$$\widehat{\delta}_{\boldsymbol{\mu}}(\boldsymbol{x}) = \mathbb{1}_{\{\rho_s(\hat{h}_{\boldsymbol{\mu}}(\boldsymbol{x})) \geq 0.5\}} = \begin{cases} 0 & \text{if} \quad \hat{h}_{\boldsymbol{\mu}}(\boldsymbol{x}) < 0, \\ 1 & \text{if} \quad \hat{h}_{\boldsymbol{\mu}}(\boldsymbol{x}) \geq 0, \end{cases} \tag{6}$$

where $\mathbb{1}_{\mathcal{A}}$ is the indicator function, $\rho_s : t \mapsto 1/(1 + e^{-t})$ is the sigmoid function and $\boldsymbol{\mu} \in \mathbb{R}^P$ denotes the set of trainable parameters [22]. The neural network is trained on the training set $\mathcal{S}$ by minimizing the empirical risk $\mathcal{R}_N(\boldsymbol{\mu})$ [23]

$$\mathcal{R}_N(\boldsymbol{\mu}) := \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\left(\hat{h}_{\boldsymbol{\mu}}(\boldsymbol{x}^{(i)}), y^{(i)}\right), \tag{7}$$

where $\mathcal{L}$ is the binary cross-entropy defined such that $\mathcal{L}(z, y) = y \log(z) + (1 - y) \log(1 - z)$ for any $0 \leq z, y \leq 1$. The general performance of $\widehat{\delta}_{\boldsymbol{\mu}}(\boldsymbol{x})$ is measured with the population risk $\mathcal{R}(\boldsymbol{\mu})$ for the ideal "0-1" loss function $\mathcal{L}_{0-1}(z, y) = \mathbb{1}_{\{z = y\}}$ and compared to the minimum risk $\mathcal{R}^*$ attained by the theoretical Bayes detector $\delta^*(\boldsymbol{x})$:

$$\mathcal{R}(\boldsymbol{\mu}) := \mathbb{E}_{(\boldsymbol{x}, y) \sim \mathcal{D}}[\mathcal{L}_{0-1}(\widehat{\delta}_{\boldsymbol{\mu}}(\boldsymbol{x}), y)], \tag{8}$$

$$\mathcal{R}^* := \mathbb{E}_{(\boldsymbol{x}, y) \sim \mathcal{D}}[\mathcal{L}_{0-1}(\delta^*(\boldsymbol{x}), y)]. \tag{9}$$

When $\mathcal{D}(\boldsymbol{x}, y)$ in (5) is known, a short calculation detailed in [15] shows that the optimal Bayes detector $\delta^*(\boldsymbol{x})$ is

$$\delta^*(\boldsymbol{x}) = \mathbb{1}_{\{h^*(\boldsymbol{x}) \geq 0\}} = \begin{cases} 0 & \text{if} \quad h^*(\boldsymbol{x}) < 0, \\ 1 & \text{if} \quad h^*(\boldsymbol{x}) \geq 0, \end{cases} \tag{10}$$

$$h^*(\boldsymbol{x}) = c_0 + \sum_{m=1}^{M} \left(c_m \exp\left(\sum_{k=1}^{K} b_{m,k}\, x^2(k) \right) \right), \tag{11}$$

where $h^*(\boldsymbol{x})$ is the decision function, $c_0 = -q_0/q_1$ and the coefficients c_m and $b_{m,k}$ depends on the $\sigma_\theta^2(k)$'s and σ_a^2. The structure of the Bayes decision function $h^*(\boldsymbol{x})$ is intrinsically explainable. The square function $t \mapsto p_2(t) = t^2$ extracts the variance information (second order statistics) from the samples $x(k)$. Each signal $k \mapsto b_{m,k}$ plays the role of a matched filter that computes a profile score related to the consistency of the input signal variance with a given signature occurring in H_1. The exponential function $t \mapsto e(t) = \exp(t)$ amplifies the most likely signature profile score. Finally, the signal $m \mapsto c_m$ acts as a filter that averages the contributions of the profile scores to get the final decision value $h^*(\boldsymbol{x})$.

In statistical learning [23], the quality of the training, i.e. the convergence of a model $\widehat{\delta}_{\boldsymbol{\mu}}(\boldsymbol{x})$ toward $\delta^*(\boldsymbol{x})$, can thus be assessed through the difference $\mathcal{R}(\boldsymbol{\mu}) - \mathcal{R}^*$. The goal is to build a neural network $\widehat{\delta}_{\boldsymbol{\mu}}(\boldsymbol{x})$ whose structure is structurally explainable (i.e., close to the natural structure of the Bayes detector) and that minimizes the error $\min_{\boldsymbol{\mu}} \mathcal{R}(\boldsymbol{\mu}) - \mathcal{R}^*$.

C. Class of explainable deep networks

Noting that the Bayes detector (10) can be rewritten as a sigmoid-based decision $\delta^*(\boldsymbol{x}) = \mathbb{1}_{\{\rho_s(h^*(\boldsymbol{x})) \geq 0.5\}}$, [15] has developed a class of neural networks $\mathcal{F}^*$ than can approximate accurately $h^*(\boldsymbol{x})$. Deep networks in $\mathcal{F}^*$ mimic the mathematical structure of $h^*(\boldsymbol{x})$ in (11) by approximating each of its internal functions and operations separately and accurately with specifically designed neural network modules. The set $\mathcal{F}^*$ is called the class of Bayes explainable neural networks. By definition, it contains the networks $\hat{f}_{\boldsymbol{\omega}}(\boldsymbol{x})$ with the structure

$$\hat{f}_{\boldsymbol{\omega}}(\boldsymbol{x}) = \alpha_0 + \sum_{m=1}^{M} \alpha_m \Phi_{e, \gamma_2}\left(\sum_{k=1}^{K} \varphi_{m,k} \Phi_{p_2, \gamma_1}(x(k)) \right), \tag{12}$$

where $\Phi_{p_2,\boldsymbol{\gamma}_1} : \mathbb{R} \mapsto \mathbb{R}$ and $\Phi_{e,\boldsymbol{\gamma}_2} : \mathbb{R} \mapsto \mathbb{R}$ are specific neural network with parameters $\boldsymbol{\gamma}_1$ and $\boldsymbol{\gamma}_2$, dedicated to the approximation of the square function $p_2(t) = t^2$ and exponential function $e(t) = e^t$ in $h^*(\boldsymbol{x})$. These two neural networks are composed of L hidden layers with exactly W neurons per layer. The whole neural network $\hat{f}_{\boldsymbol{\omega}}(\boldsymbol{x})$ is implemented as a CNN. The parameters $\alpha_0, \dots, \alpha_m$ and $\varphi_{m,k}$ play respectively the role of the coefficients $c_0, \dots, c_m$ and $b_{m,k}$ in (11). The imposed structure on $\hat{f}_{\boldsymbol{\omega}}(\boldsymbol{x}) \in \mathcal{F}^*$ allows us to later interpret the role of each layer or group of layers in $\hat{f}_{\boldsymbol{\omega}}(\boldsymbol{x})$.

The main default of $\mathcal{F}^*$ is its loss of performance for small data sets compared to state of the art CNNs. This loss of performance is mainly due to the estimation of the M filters $k \mapsto b_{m,k}$ of size K by $k \mapsto \varphi_{m,k}$. In literature, 1D CNNs are generally designed with medium sized filters (≈ 128) on their first layer [6], [8], which often end up with most filters learning collectively a logarithmically distributed bandpass filter bank [24] with same patterns found at different scales. Convolutional layers generally use small filter sizes [24].

III. Deep Network Architecture

This section proposes to replace the filters $b_{m,k}$, in the space of high temporal resolution, with shorter filters in a space of reduced dimension. There exist expert features such as the scattering transform [17] whose structure is explainable. Our objective is to design a neural network architecture that is inspired by the scattering transform.

A. Filter size reduction with scattering transform

The scattering transform [17] is a representation of modulation spectrum at multiple orders, where frequency bands are equally spaced on the exponential scale. Its mathematical structure is similar to a CNN architecture, as its computation is done through a cascade of wavelet transforms and modulus non-linearities. It relies on wavelet theory and it does not involve any learning. For a signal $\boldsymbol{x}$, the first-order scattering transform $\mathcal{S}[\boldsymbol{x}](\lambda)$, indexed by the frequency λ, is given by

$$\mathcal{S}[\boldsymbol{x}](\lambda) = |\boldsymbol{x} \star \psi_\lambda| \star \phi_T. \tag{13}$$

In practice, it is obtained by convolving the signal $\boldsymbol{x}$ with the wavelet filer ψ_λ of central frequency $\lambda \in \Lambda$, where Λ is a grid composed of Q central frequencies per octave and $\star$ denotes the discrete convolution. The filter ϕ_T locally averages the signal over a time duration T. Here we set it to the signal size $T = K$ (similar to a global average pooling operation in CNN), making coefficient $\mathcal{S}[\boldsymbol{x}](\lambda)$ a time invariant real value and not a vector. Analyzing the signal $\boldsymbol{x}$ with filters ψ_λ distributed in a Mel scale enables us to reduce its dimension. Hence, we can prove (the proof is omitted due to the lack of space) the following result.

Theorem 1: Let $\boldsymbol{x}$ be a signal defined by (1) and M real filters $k \mapsto b_{m,k}$. Then, there exist some sequences $\lambda \in \Lambda \mapsto \beta_{m,\lambda}$ of coefficients such that

$$\left| \sum_{\lambda \in \Lambda} \beta_{m,\lambda} \mathcal{S}[\boldsymbol{x}^2](\lambda) - \sum_{k=1}^{K} b_{m,k}\, x^2(k) \right| \leq \epsilon, \forall m \in [\![M]\!], \tag{14}$$

where ε is a constant depending on the $\sigma_\theta^2(k)$'s and Λ. This theorem tells us that we can avoid the long filters $k \mapsto b_{m,k}$. In practice, the number of parameters is significantly reduced since $|\Lambda| \ll K$.

B. Learning-based scattering transform (LST)

Based on our Theorem 1, we propose a learning-based scattering transform, denoted as LST, in order to obtain a more flexible representation of the signal inside a deep neural network architecture. Let us build a family of learned filters ψ_λ where $\lambda = (s, f) \in \Lambda = [\![S]\!] \times [\![F]\!]$ is interpreted as follows: s corresponds to an imposed scale (the input signal size is reduced recursively) and f to a filter identifier number (several different filters are used in the LST). By this way, we propose a LST $\widetilde{\mathcal{S}}[\boldsymbol{x}] : \mathbb{R}^K \mapsto \mathbb{R}^{S \times F}$, that takes the signal $\boldsymbol{x}$ and returns the set of values $\widetilde{\mathcal{S}}[\boldsymbol{x}](\lambda)$ for $\lambda = (s, f)$.

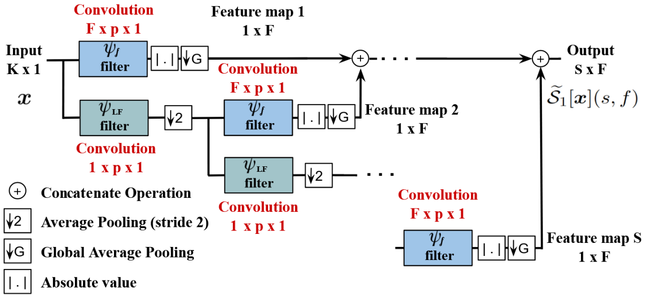

Fig. 1. Architecture of the LST $\widetilde{\mathcal{S}}[\boldsymbol{x}]$ applied to a signal $\boldsymbol{x}$.

Fig. 1 describes our first order LST module $\widetilde{\mathcal{S}}[\boldsymbol{x}](\lambda)$. Firstly, let us introduce the temporal scale decomposition. This corresponds to the green branch from top to down in Fig. 1. A 1D convolutional filter ψ_{LF}, of size $p \times 1$, is first applied to the input signal $\boldsymbol{x}$. An average pooling operation ϕ_2 of size and stride 2 is then applied for subsampling, reducing the signal size by two. The filter ψ_{LF} is not learned. It is a low pass Morlet wavelet which constrains the next layers to analyze lower frequency octaves. Hence, we define the scale recursion

$$\boldsymbol{x}_{\dagger s+1} = (\boldsymbol{x}_{\dagger s} \star \psi_{LF}) \star \phi_2, \tag{15}$$

with the initialization $\boldsymbol{x}_{\dagger 1} = \boldsymbol{x}$. Each decomposition corresponds to a layer in our neural network. The size of a signal $\boldsymbol{x}_{\dagger s}$ must be larger that the size p of ψ_{LF}. Hence, S must satisfy $2p > \frac{K}{2^S} \geq p$. This subsampling allows us to learn filters ψ_λ with small receptive fields.

Next, let us introduce the learned filters ψ_λ with $\lambda = (s, f)$. Actually, the family of filters $\psi_{(s,f)}$ does not depend on s : we use the same family of filters for all scale levels. Hence, at the layer of scale s, the LST applies F filters ψ_f with $1 \leq f \leq F$:

$$\widetilde{\mathcal{S}}[\boldsymbol{x}](s, f) = |\boldsymbol{x}_{\dagger s} \star \psi_f| \star \phi_K, \tag{16}$$

where ϕ_K is a global average pooling of size K and $\lambda = (s, f)$. Each filter ψ_f of size p is learned. By this way, each layer s analyzes a different octave of $\boldsymbol{x}$ with F different filters ψ_f instead of the Q wavelets per octave used in the usual scattering transform [17].

Finally, at each node of the architecture, two feature maps are produced, giving birth to two branches equivalent to the high frequency and low frequency decompositions of the usual discrete wavelet transform. In the first branch (on the top of each node in Fig. 1), we apply to the signal $\boldsymbol{x}_{\dagger s}$ the F filters ψ_f to get $\widetilde{\mathcal{S}}[\boldsymbol{x}](s,f)$ for all f. The output $\widetilde{\mathcal{S}}[\boldsymbol{x}](s,f)$ at scale s has a size $1 \times F$ because of the global pooling. This output is concatenated with earliest output layers, creating shortcut connections [25], between each layer and the final network output $\widetilde{\mathcal{S}}[\boldsymbol{x}]$, of size $S \times F$. Since the LST is inspired from the scaterring transform, we can show the following theorem.

Theorem 2: Let $\boldsymbol{x}$ be a signal defined by (1) and M real filters $k \mapsto b_{m,k}$. Then, there exist some sequences $\lambda \in \Lambda = [\![S]\!] \times [\![F]\!] \mapsto \beta_{m,\lambda}$ of coefficients such that

$$\Big| \sum_{\lambda \in \Lambda} \beta_{m,\lambda} \widetilde{\mathcal{S}}[\boldsymbol{x}^2](\lambda) - \sum_{k=1}^{K} b_{m,k}\, x^2(k) \Big| \leq \epsilon, \forall m \in [\![M]\!], \quad (17)$$

where ε is a constant depending on S, F and the $\sigma_\theta^2(k)$'s.

C. Explainable LST CNN

We can now describe our full neural network architecture that extends (12). The LST CNN architecture is given by

$$\hat{h}_{\boldsymbol{\mu}}(\boldsymbol{x}) = \alpha_0 + \sum_{m=1}^{M} \alpha_m \Phi_{e,\boldsymbol{\gamma}_2}\left(\sum_{\lambda \in \Lambda} \beta_{m,\lambda} \widetilde{\mathcal{S}}[\Phi_{p_2,\boldsymbol{\gamma}_1}(\boldsymbol{x})](\lambda) \right), \quad (18)$$

where $\boldsymbol{\mu}$ contains the α_m's, the $\beta_{m,\lambda}$'s, $\boldsymbol{\gamma}_1$ and $\boldsymbol{\gamma}_2$. Since the function (18) combines two structurally explainable CNNs given both in (12) and Theorem 2, it is still structurally explainable. The class of all LST CNNs defined by (18) is denoted $\mathcal{H}^*$. Theorem 3 extends Theorem 1 in [15] with our novel architecture to guarantee the approximation of $h^*(\boldsymbol{x})$.

Theorem 3: Let $\epsilon \in (0, \frac{1}{2})$, $\mathcal{X}$ is a bounded subset of $\mathbb{R}^K$, and $D > 0$ such that $p_2(t)$ and $e(t)$ are defined over $[-D, D]$. Then, there exist a LST CNN in $\mathcal{H}^*$ and $Q > 0$ such that

$$\inf_{\hat{h}_{\boldsymbol{\mu}} \in \mathcal{H}^*} \sup_{\boldsymbol{x} \in \mathcal{X}} |\hat{h}_{\boldsymbol{\mu}}(\boldsymbol{x}) - h^*(\boldsymbol{x})| \leq \epsilon, \quad (19)$$

when the number of layers L satisfies $L \leq Q \log_2^2(1/\epsilon) + \log_2(1 + 1/D)$. Furthermore, there exists $C' > 0$ such that

$$\inf_{\hat{h}_{\boldsymbol{\mu}} \in \mathcal{H}^*} \mathcal{R}(\boldsymbol{\mu}) - \mathcal{R}^* \leq C' \epsilon. \quad (20)$$

IV. Experiments

A. Experiments on simulated data

This section compares the accuracy of our architecture $\hat{h}_{\boldsymbol{\mu}}$, our previous work architecture $\hat{f}_{\boldsymbol{\omega}}$ [15], the optimal Bayes detector (10) and a classical FCNN architecture [6], for $M = 16$ merchant ship like signatures. The Bayes detector optimal risk $\mathcal{R}^*$ is computed accurately with a Monte-Carlo procedure and shown as the red curve in Fig. 2. The test and train curves are obtained by averaging the performances of 10 models of each architecture, on a test set $\mathcal{S}_{test}$ of $2 \cdot 10^5$ independent samples and a training set $\mathcal{S}$ containing up to $N_{max} = 8 \cdot 10^5$ signals of length $K = 512$.

The architecture $\hat{h}_{\boldsymbol{\mu}}$ is the same as the one presented in [15] except that it contains the LST module proposed in this paper. This module contains $F = 4$ filters of size $p = 32$ randomly initialized. It requires learning $(p+1)F = 132$ parameters for its filters, plus $S \cdot F$ = 16, with $S = 4 \approx \ln_2(K) - \ln_2(p)$, for the $\beta_{m,\lambda}$'s, compared to $K = 512$ parameters for the $b_{m,k}$'s. The Adam optimizer is used during training with a maximum kernel constraint on convolutional layers. The stopping criterion is an absence of improvement in validation loss during 15 consecutive epochs, or after 100 training epochs.

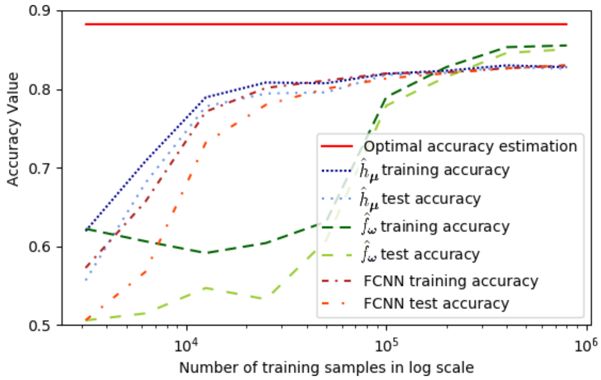

Fig. 2. Average accuracy on train and test sets for 10 trainings of $\hat{h}_{\boldsymbol{\mu}}$, $\hat{f}_{\boldsymbol{\gamma}}$ and a FCNN, as a function of data set size.

For low data set sizes, $\hat{h}_{\boldsymbol{\mu}}$ with its 2711 parameters has a similar accuracy than the FCNN with 1 037 281 parameters, but without being a black box algorithm. The classifier $\hat{f}_{\boldsymbol{\omega}}$, with larger convolutional filter and slightly more parameters (10 515), is however closer to the optimal test for huge data set, which may be due to our LST approximation loss. This experiment shows that learning long temporal dependencies with long convolutional filters is detrimental during learning. Our approach of learning long temporal dependencies in a learned space of reduced dimensions, modeled by the LST module, is a solution which guarantees explainability.

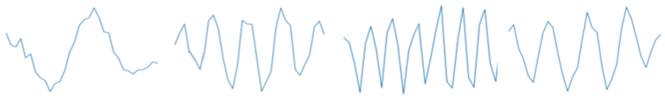

Fig. 3. Examples of learned filters ψ_f within $\hat{h}_{\boldsymbol{\mu}}$.

A few filters ψ_f are shown on Fig. 3. They are not very noisy and they learn different frequency responses within a same octave, as reflected by the different periodicities.

B. Experiments on public data set

To validate our approach, we test our LST module within another CNN architecture on the ShipsEar data set [19], which contains acoustic recordings from $M = 11$ vessel types. The class H_0 is composed of ambient recordings and some parasitic recordings present in the data set. The class H_1 is composed of all the signals with vessel sounds. All recordings are split into segments of 10 seconds with a same sampling frequency $f_s = 22050$ Hz. We analyze all resulting segments to keep only audible vessel signals in H_1 and duplicate segments from H_0 to balance class samples and obtain a total of 1412 training samples and 352 validation samples. Fig. 4 shows some typical spectrograms of the samples.

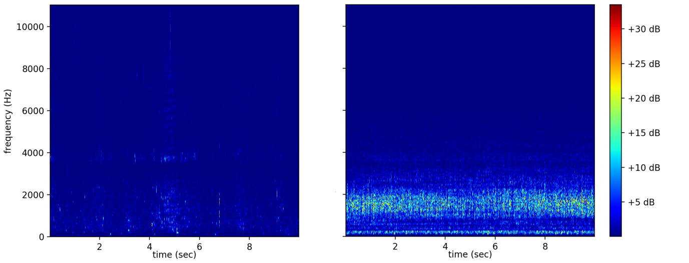

Fig. 4. Spectrogram of samples from H_0 and H_1.

Our architecture is tuned on the data set. Hence, our CNN first includes a time dependant LST module by using an average pooling filter ϕ_T, of duration $T = 0,25$ seconds, in place of the global average pooling operation ϕ_K in (16) :

$$\widetilde{\mathcal{S}}[\boldsymbol{x}](s, f, t) = |\boldsymbol{x}_{\dagger s} \star \psi_f| \star \phi_T(t). \tag{21}$$

The multiscale decomposition criteria for S is such that $\frac{K}{2^S} \geq 0.2\, f_s$. Hence, $\widetilde{\mathcal{S}}[\boldsymbol{x}]$ gives a multi channel 1D output feature of dimension ($K/2^S$, $S \times F$) = (81, 112), by concatenating the different scales of analysis s with the indexes f, with $S = 7$ and $F = 16$. The CNN architecture is then followed by a succession of 4 convolutional layers and average pooling operations, with dimensions [(32, 16), (9, 24), (9, 32), (9, 48)], as well as a global pooling operation and a final dense layer.

On the validation set, our model trained with the previous procedure obtains an accuracy of $81,25\%$. It is lower than the $94,3\%$ obtained by [26] with a ResNet on the same data set and a slightly different setting. The better performance of ResNet is not surprising since it uses an aggregation of expert features including Frequency Cepstral Coefficients (MFCC) and Log-Mel Spectrogram. Our neural network is trained only on raw acoustic signal with very few samples.

V. Conclusion

This paper proposed a LST CNN module that can efficiently learn long temporal dependencies of cyclostationary signals, while providing explainable representations. This module was used in a class of explainable CNNs to mimic an optimal Bayes detector structure for a ship noise classification problem. The main difference with the first-order scattering transform is the use of learned convolutional filters instead of predefined wavelets, offering more flexibility for feature extraction.

References

[1] Li Deng, Geoffrey Hinton, and Brian Kingsbury, "New types of deep neural network learning for speech recognition and related applications: An overview," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 8599–8603.

[2] Karol J Piczak, "Esc: Dataset for environmental sound classification," in *Proceedings of the 23rd ACM international conference on Multimedia*, 2015, pp. 1015–1018.

[3] Sung-Hoon Byun, Sea-Moon Kim, Cheolsoo Park, Kihun Kim, and Chong-Moo Lee, "Cyclostationary analysis of underwater noise for vehicle propeller monitoring," in *OCEANS*. IEEE, 2016, pp. 1–4.

[4] Kil Woo Chung, Alexander Sutin, Alexander Sedunov, and Michael Bruno, "Demon acoustic ship signature measurements in an urban harbor," *Advances in Acoustics and Vibration*, vol. 2011, 2011.

[5] I Kirsteins, P Clark, and L Atlas, "Maximum-likelihood estimation of propeller noise modulation characteristics," *Underwater Acoustic Measurements: Technologies and Results*, 2011.

[6] Wei Dai, Chia Dai, Shuhui Qu, Juncheng Li, and Samarjit Das, "Very deep convolutional neural networks for raw waveforms," in *IEEE ICASSP*, 2017, pp. 421–425.

[7] Boqing Zhu, Changjian Wang, Feng Liu, Jin Lei, Zhen Huang, Yuxing Peng, and Fei Li, "Learning environmental sounds with multi-scale convolutional neural network," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.

[8] Hardik B Sailor, Dharmesh M Agrawal, and Hemant A Patil, "Unsupervised filterbank learning using convolutional restricted boltzmann machine for environmental sound classification.," in *Interspeech*, 2017, vol. 8, p. 9.

[9] Suraj Kamal, Shameer K Mohammed, PR Saseendran Pillai, and MH Supriya, "Deep learning architectures for underwater target recognition," in *2013 Ocean Electronics (SYMPOL)*. IEEE, 2013, pp. 48–54.

[10] Michele Valenti, Stefano Squartini, Aleksandr Diment, Giambattista Parascandolo, and Tuomas Virtanen, "A convolutional neural network approach for acoustic scene classification," in *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2017, pp. 1547–1554.

[11] Thomas Fel and David Vigouroux, "Representativity and consistency measures for deep neural network explanations," *arXiv preprint arXiv:2009.04521*, 2020.

[12] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929.

[13] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin, """ why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[14] Ning Xie, Gabrielle Ras, Marcel van Gerven, and Derek Doran, "Explainable deep learning: A field guide for the uninitiated," *arXiv preprint arXiv:2004.14545*, 2020.

[15] Thomas Mahiout, Lionel Fillatre, and Laurent Deruaz-Pepin, "Explainable deep learning detection of gaussian propeller noise with unknown signal-to-noise ratio," in *MLSP 2021*, October 2021, p. 110.

[16] Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio, "Fantastic generalization measures and where to find them," *arXiv preprint arXiv:1912.02178*, 2019.

[17] Joakim Andén and Stéphane Mallat, "Deep scattering spectrum," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4114–4128, 2014.

[18] Thomas Mahiout, Lionel Fillatre, and Laurent Deruaz-Pepin, "Propeller noise detection with deep learning," in *ICASSP*, 2020, pp. 306–310.

[19] David Santos-Domínguez, Soledad Torres-Guijarro, Antonio Cardenal-López, and Antonio Pena-Gimenez, "Shipsear: An underwater vessel noise database," *Applied Acoustics*, vol. 113, pp. 64–69, 2016.

[20] Johan G Lourens, "Passive sonar ML estimator for ship propeller speed," in *Proceedings of the 1997 South African Symposium on Communications and Signal Processing*. IEEE, 1997, pp. 13–18.

[21] H Vincent Poor, *An introduction to signal detection and estimation*, Springer Science & Business Media, 2013.

[22] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep learning*, MIT press, 2016.

[23] Luc Devroye, László Györfi, and Gábor Lugosi, *A probabilistic theory of pattern recognition*, Springer Science & Business Media, 2013.

[24] Pavel Golik, Zoltán Tüske, Ralf Schlüter, and Hermann Ney, "Convolutional neural networks for acoustic modeling of raw time signal in LVCSR," in *Sixteenth annual conference of the international speech communication association*, 2015.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on CVPR*, 2016, pp. 770–778.

[26] Feng Hong, Chengwei Liu, Lijuan Guo, Feng Chen, and Haihong Feng, "Underwater acoustic target recognition with resnet18 on shipsear dataset," in *2021 IEEE 4th International Conference on Electronics Technology (ICET)*. IEEE, 2021, pp. 1240–1244.