A Robust Approach for Securing Audio Classification Against Adversarial Attacks

Mohammad Esmaeilpour, Patrick Cardinal, and Alessandro Lameiras Koerich, *Member, IEEE*

***Abstract*—Adversarial audio attacks can be considered as a small perturbation unperceptive to human ears that is intentionally added to the audio signal and causes a machine learning model to make mistakes. This poses a security concern about the safety of machine learning models since the adversarial attacks can fool such models toward the wrong predictions. In this paper we first review some strong adversarial attacks that may affect both audio signals and their 2D representations and evaluate the resiliency of the most common machine learning model, namely deep learning models and support vector machines (SVM) trained on 2D audio representations such as short time Fourier transform, discrete wavelet transform (DWT) and cross recurrent plot against several state-of-the-art adversarial attacks. Next, we propose a novel approach based on pre-processed DWT representation of audio signals and SVM to secure audio systems against adversarial attacks. The proposed architecture has several preprocessing modules for generating and enhancing spectrograms including dimension reduction and smoothing. We extract features from small patches of the spectrograms using the speeded up robust feature (SURF) algorithm which are further used to generate a codebook using the K-Means++ algorithm. Finally, codewords are used to train an SVM on the codebook of the SURF-generated vectors. All these steps yield to a novel approach for audio classification that provides a good trade-off between accuracy and resilience. Experimental results on three environmental sound datasets show the competitive performance of proposed approach compared to the deep neural networks both in terms of accuracy and robustness against strong adversarial attacks.**

***Index Terms*—Spectrograms, Environmental Sound Classification, Adversarial Attack, K-Means++, Support Vector Machines (SVM), Convolutional Denoising Autoencoder.**

I. Introduction

Adversarial attacks pose security issues since they can be unrecognizable to human eyes (in case of spectrograms) or human ears (in case of audio signals) while they can easily fool any trained machine learning model with very high confidence. As these machine learning models are becoming more present in many devices and applications, there exists an urgent need for improving their robustness against adversarial attacks. Basically, an adversarial attack algorithm formulates an optimization equation toward finding the smallest possible perturbation value to be added to a given legitimate input file (image, audio, spectrogram, etc.) aiming at changing its true label. This perturbation should be as small as possible to be imperceptible to human hearing or visual system. Adversarial attacks have been attracting the attention of many researchers, mainly in the domain of computer vision [1], [2], [3]. However, adversarial attacks may also pose a serious threat to voice assistant devices, speech and speaker recognition as well as other audio-related applications. In spite of that, few studies have addressed adversarial attacks for audio signals [4]. One of the possible reasons is the considerable optimization overhead of adversarial algorithms when applied to audio signals due to their high dimensionality. In the big picture, adversarial examples of audio signals can be crafted during sound production or post production via changing amplitude or frequency of audio mainly into the region inaudible by humans. This is difficult and needs to be treated carefully because there is no guarantee for producing an adversarial example and the output could be just a noisy sample. In the case of post-production of adversarial examples, the adversary can either solve for an optimization problem (costly) or develop an adversarial filter in order to apply some perturbations on a given legitimate audio before passing it through a machine learning model. In both cases, the victim model could be fooled toward bad wishes of adversaries and make the system misbehave.

In this paper, we investigate the threat of adversarial attacks on generic environmental audio signals due to the diversity that we may find, ranging from baby crying to engines, horns to dog barking or people chatting with numerical text-free labels. These types of attacks are quite useful for other relevant domains of speaker recognition and music classification and they may be generalizable to speech-to-text applications, though the latter is not discussed in this paper. Environmental sound classification has been a challenging problem in machine learning research [5]. Both traditional (e.g. SVM) and advanced deep learning classifiers have shown highly competitive performances on benchmarking datasets such as ESC-10 [6], ESC-50 [6], and UrbanSound8K [7]. Besides the supervised classifiers, there are some carefully engineered unsupervised models such as spherical K-means [5], [8] for sound representation learning. The classifiers have mainly been trained either on audio clips as a 1D signal or on 2D representation spaces of spectrograms. In both cases, convolutional neural networks (CNNs) have shown better performances compared to other classifiers. For instance, on the UrbanSound8K dataset, the CNN proposed by Salamon and Bello [9] which implements three convolutional layers followed by two fully connected layers outperforms its prior cutting-edge classifier as class-conditioned learning [5]. Also, for ESC-10 and ESC-50 datasets, a 1D CNN with eight convolution layers (SoundNet) [10] outperforms random forest

M. Esmaeilpour, P. Cardinal and A. L. Koerich are with the Department of Software and Information Technology Engineering, École de Technologie Supérieure (ÉTS), University of Quebec, Montreal, QC, Canada, e-mail: (mohammad.esmaeilpour.1@etsmtl.ca, patrick.cardinal@etsmtl.ca, alessandro.koerich@etsmtl.ca).

Accepted for publication November, 2019.

[6], SVM using MFCCs [6], and convolutional autoencoders [10]. In addition to these CNNs, two advanced deep learning classifiers of AlexNet and GoogLeNet which have shown remarkable performances on image classification tasks (e.g. ImageNet dataset) have been used for environmental sound classification. Interestingly, these two networks on the 2D audio representation spaces have been achieving the highest recognition performance for the three aforementioned datasets as reported by Boddapati *et al.* [11]. Similar to the computer vision perspective of classification, the main open problem in environmental sound classification seemingly is no longer improving recognition accuracy of classifiers but improving their strengths against some carefully crafted misleading examples, the adversarial ones. Therefore, the proposed approach for environmental sound classification is based on two findings: (i) deep learning models, particularly AlexNet and GoogLeNet outperform conventional classifiers on handcrafted features such as SVM; (ii) SVM in general is more robust against adversarial attacks, potentially because it learns from low-dimensional feature vectors that might decrease the chance of being affected by adversarial perturbations compared to deep models which learn from raw data. Following these facts, we developed an SVM-based classifier and tuned it toward achieving recognition accuracy competitive to above mentioned deep models aiming at making a good trade-off between recognition accuracy and robustness against adversarial attacks. Since there is no standard metric for evaluating the quality of such a trade-off, we introduce a distance metric based on the error rate versus the fooling rate.

Our contribution in this paper is threefold: (i) we present common adversarial attacks for audio and how they can affect the security of audio applications; (ii) we characterize the vulnerability of state-of-the-art models based on 2D representations to adversarial attacks and the transferability of these attacks between different models; (iii) we propose a novel approach for environmental sound classification that in addition to providing a high recognition accuracy, competitive to the state-of-the-art, it is also robust against several adversarial attacks without developing any reactive or proactive defense processes.

This paper is organized as follows. Section II introduces adversarial attacks in general (or that affects image applications) and describe the most important ones. It also presents the adversarial attacks that may affect audio applications based on 2D representations of the audio signal and discuss adversarial attacks that may affect 1D audio signal. Section III presents the main 2D representations for audio signals. Section IV proposes the classification approach that aims both achieving good classification accuracy and robustness to adversarial attacks, but which does not use any kind of reactive or proactive defense. In Section V we characterize the vulnerability of the state-of-the-art models in the problem of environmental sound classification, measure the resiliency of the proposed model versus its dense CNNs counterpart and review adversarial example transferability among these models. The conclusions and perspectives of future work are presented in the last section.

II. Adversarial Attacks

Adversarial attacks can be considered as carefully crafted perturbations that when intentionally added to a legitimate example, lead machine learning models to misbehave [12]. Considering x as a legitimate sample, then an adversarial example x' can be crafted in such a way that:

$$x \approx x', \qquad f^*(x) \neq f^*(x') \tag{1}$$

where f^* is the post-activation function. Assuming that x is an image or an audio signal, the crafted x' should be unrecognizable by human visual or hearing system.

There are several algorithms for generating x', mainly in the context of images. The adversarial attacks can be categorized into different groups. For instance, if the adversary has access to the model architecture, parameters, training dataset, etc. it is categorized as a white-box attack, otherwise it is called black-box. Also, adversarial attacks can have other taxonomy such as targeted, where the adversarial perturbation is crafted having in mind a specific target label, and non-targeted, where the adversarial perturbation is crafted to induce a machine learning model to any incorrect label. Due to the importance of studying adversarial threats for data-driven machine learning models, many attack algorithms have been proposed and they have shown a great performance in fooling advanced models. However, the main challenge of almost all attack algorithms is their computational complexity, which makes adversarial training very time-consuming. Among all these proposed algorithms, the Fast Gradient Sign Method (FGSM) [13] was one of the first attacks, which still remains one of the most effective adversarial attacks. FGSM was originally built to attack deep CNNs but it can also be a serious threat for non-deep neural network architectures. Using the FGSM attack, an adversarial example x' can be generated by Eq. 2.

$$x' = x + \epsilon \times \operatorname{sign}(\nabla_x J(\theta, x, l)) \tag{2}$$

where x and y are the original input image and its true label respectively, ϵ is a constant value which can be determined by an optimization scheme, and J is the cost function for the model parameter θ obtained after completing the training process. FGSM is a white-box attack which means that the model parameter θ should be accessible to fetch its gradient information and generate the adversarial input x'. In other words, by providing the trained model and the training dataset, FGSM can generate adversarial inputs x' using Eq. 2 with unrecognizable differences to the original input x which can perhaps make the model θ to predict a wrong label with high confidence.

The iterative version of the FGSM attack is known as Basic Iterative Method (BIM) [1] and its attack frequency is higher than one. In fact, BIM's optimization procedure can stop after generating the first adversarial example (BIM-a) or continue up to a pre-defined number of iterations (BIM-b). These two attacks are actually the improved version of FGSM which increase the attack rate to the cost of higher computational complexity.

Carlini and Wagner [14] have proposed a strong optimization-based attack known as CWA, which is able to generate samples very similar to a legitimate sample regarding the similarity metric d_i defined in Eq. 3.

$$d_i = \|x_i - x'_i\| \tag{3}$$

where i is the sample index. CWA attempts to minimize d_i as:

$$\min_{c} \|d_i\| + c \times g(x + d_i) \quad s.t. \quad x + d_i \in [0,1]^n \tag{4}$$

where $g(d) \geq 0 \iff f(d) = l'$. Reminding that l and l' refer to the correct label for clean and adversarial examples respectively. The intuition behind Eq. 4 is similar to the dropout variational inference introduced by Li *et al.* [15]. This attack is very similar to the FGSM attack with two main differences: (i) it changes the input (x_i) using the $\tanh$ function; (ii) it uses a difference between logits (the vector of non-normalized predictions that a model generates) instead of optimizing a cost function for regular cross-entropy. This attack is one of the strongest iterative and targeted adversarial attacks which could be very effective in fooling deep neural networks, though, it is slow in runtime.

The adversarial attacks presented so far are designed for deep neural networks. Since the approach that we are proposing in this paper is based on SVMs and DWT representation of audio signals, we present two well-known adversarial attacks that were originally designed to attack SVM models: Evasion attack (EA) and Label Flipping attack (LFA). EA [16] and LFA [17] are two strong adversarial attacks against SVM models, though EA is also effective against neural networks. The main difference between these two attacks is that LFA contaminates the training data by flipping the true labels of the samples, while EA manipulates the sample distribution aiming to change the true labels. In both cases, the decision boundary of the model is shifted toward maximum loss for the test set. The general intuition behind EA is to map a given input x over a support vector(s) by simply flipping its label. This flipping can be toward the trained weight direction(s) of the SVM as given by Eq. 5.

$$\vec{x'} = \vec{x} - \epsilon \odot \frac{\vec{w_i}}{\|\vec{w_i}\|} \tag{5}$$

where x' is the crafted adversarial example and w_i is the weight vector discriminating support vectors. Also, ϵ is a small constant value. The intuition behind these two attacks is the geometrical definition of support vectors as given by Eq. 6.

$$\min \vec{w} \qquad s.t. \quad l_i(\vec{w}\vec{x_i} - b) \geq 1 \quad i = 1, \ldots, n \tag{6}$$

where $\vec{w}$ is a vector normal to the hyperplane ($\vec{w}\vec{x} - b = 0$), b is a bias term. The position of the support vectors ($\vec{x_i}$) can be depicted as shown in Fig. 1.

In other words, the SVM model will be fooled by moving a datapoint perpendicularly toward the opposite direction of its weight vector. This attack is generalizable to soft margin SVM by simply optimizing the value of ϵ in Eq. 7.

$$\frac{1}{n}\sum_{i=1}^{n} \max(1 - l_i(\vec{w}\vec{x} - b), 0) + \epsilon \|\vec{w}\|^2 \tag{7}$$

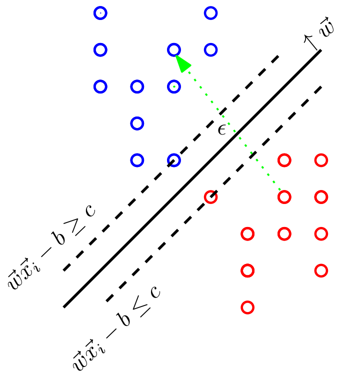


Fig. 1: Simplified visualization of hard margin form of SVM adversarial attack. The dotted arrow depicts the Eq. 5. Datapoints are represented in two colors: red (adversarial) and blue (legitimate).

As long as the optimization of ϵ is perpendicularly directed toward the $\vec{w}_k$ (the weight vector in one-versus-all SVM), the SVM model cannot distinguish an adversarial example from legitimate samples. This data contamination in EA can be implemented by taking advantage of gradient information and local search for achieving the best data perturbation with a specific budget as introduced by Biggio *et al.*[16]. The gradient information can be exploited for different kernels. For an RBF kernel of $k(x, x_i) = \exp(-\gamma \|x - x_i\|^2)$, the gradient can be computed by Eq. 8.

$$\nabla k(x, x_i) = -2\gamma \exp(-\gamma \|x - x_i\|^2)(x - x_i) \tag{8}$$

Also, for a polynomial kernel $k(x, x_i) = (\langle x, x_i \rangle + c)^p$, its gradient is achievable by Eq. 9.

$$\nabla k(x, x_i) = p(\langle x, x_i \rangle + c)^{p-1} x_i \tag{9}$$

Therefore, the adversarial example x' can be computed by:

$$x' = x - t\nabla f(x) \tag{10}$$

where t is a small value of step size and f denotes the learned filters in the hypothesis space H where $k(x, x_i) = \Phi(x)^\top \Phi(x_i)$ and Φ is a mapping function from input to the feature space. Different from EA, LFA does not generate an adversarial example via distorting the legitimate samples, but it contaminates the labels of such samples. This should result in maximum loss in the test set while it is expected to be minimum for the training set. The LFA attack can be implemented by solving the following optimization problem:

$$\min_{q,w,\epsilon,b} \gamma \sum_{i=1}^{2n} q_i(\epsilon_i - \xi_i) + \frac{1}{2}\|w\|^2 \tag{11}$$

subject to:

$$y_i(w)^\top x_i + b \geq 1 - \epsilon_i \quad \epsilon_i \geq 0, \quad i = 1, \cdots, 2n \tag{12}$$

having the budget:

$$\sum_{i=n+1}^{2n} c_i q_i \leq C \tag{13}$$

where $q_i \in \{0,1\}$ and c_i and C denote the flipping cost of each sample and the total flipping cost, respectively, from adversary's point of view. The hinge loss function ($\mathcal{L}$), defined in Eq. 14

$$\mathcal{L}(y, f(x)) := \max(0, 1 - y_i f_{S'}(x)) \tag{14}$$

where $f_{S'}$ is the contaminated dataset which also includes the original dataset S. Also, q_i is an indicator variable for controlling over the legitimate (q=0) and contaminated sample (q=1). Similarly, ξ_i refers to the hinge loss of the classifier f_S:

$$f_S(x) = w^\top x + b, \qquad w := \sum_{i=1}^{n} \alpha_i \Phi(x_i). \tag{15}$$

Herein, b is the bias term and α denotes the Mercer kernel coefficient of the SVM.

A. Transferability of Adversarial Attacks

One of the main characteristics of adversarial attacks described so far is that they are non-targeted toward a specific label as they maximize the probability of any label other than the true one. This is very tricky since it opens the opportunity of adversarial transferability to other data-driven models. This means that adversarial examples maintain their effectiveness against models different from those targeted by the attack. For instance, the FGSM attack, which targets CNNs, could completely fool a maxout network trained on the MNIST dataset [13]. Goodfellow *et al.* [13] have shown that the linear manner of FGSM can be transferred to other classifiers including SVMs even with radial basis kernel function. This was a break-point of studying adversarial transferability for all classifiers, from logistic regression (simple) to very-deep CNNs (complex). Recently, Sabour *et al.* [2] have shown the great effectiveness of FGSM on fooling other deep architectures with and without convolution layers.

A lot of effort has been made on improving transferability of adversarial attacks. From expanding input patterns (data-wise) [18] to developing ensemble models that produce more misleading adversarial examples (model-wise) [19]. Therefore, this is a real threat since adversarial attacks can be transferred among almost all models, e.g. from CNN to SVM, logistic regression and decision trees [20]. Besides that, models trained for speech-to-text translation have also been successfully fooled by crafted adversarial examples [4]. Empirically, machine learning models designed for audio applications, based on either in 1D or 2D representation are very vulnerable against adversarial attacks and the current defense schemes, such as those proposed by Das *et al.* [21], do not work appropriately.

B. Adversarial Attacks for Audio Signals

Adversarial attacks have been mainly studied in the domain of computer vision to manipulate images (2D). It has been shown that 2D deep learning models are quite vulnerable against white-box and black-box optimization-based attacks [13]. However, these optimization-based attacks are usually very costly and they require too many callbacks to each legitimate sample, pixel-by-pixel. Generalizing these optimization-based attacks to audio signals (1D) is not straightforward since the audio signal is high-dimensional data, even considering a single audio channel. For instance, five seconds of mid-quality audio corresponds to an array of 110,250 points. Therefore, computing a similarity measure such as the $L2$-norm between legitimate and crafted examples as a part of an adversarial optimization criterion is very challenging compared to 2D arrays.

Alzantot *et al.* [22] and Du *et al.* [23] have proposed speech-to-text adversarial attacks where the optimization process is replaced with heuristic algorithms like genetic algorithms [22] or particle swarm optimization [23] to mitigate the considerable cost of the optimization process for audio signals. Basically, these greedy and evolutionary algorithms introduce random noise to a legitimate sample which in turn increases the chance of having a dissimilarity between original and crafted adversarial examples. However, this also paves the way for an easy detection of adversarial samples by simple algorithms. On the other hand, in the most effective adversarial attacks for images (e.g. FGSM, BIM, CWA, etc.), adversarial perturbations are generated by an optimization process that has two key constraints: producing a wrong label; having a visual similarity between legitimate and adversarial examples.

It is difficult to satisfy these constraints for adversarial audio because it would be very challenging and time-consuming optimizing for these key constraints considering the high dimensionality of audio signals. Moreover, in contrast with images, audio signals are not convolved in rows and columns and this also makes very difficult solving the optimization problem for adversarial audio perturbation. These constitute enough grounds for introducing evolutionary algorithms to randomly search for possible adversarial perturbations which basically can only solve for the first mentioned key constraint. The main side effect of this approach is staying close to the manifold of legitimate samples and therefore being detected by a tuned classifier or a simple adversarial detector such as downsampling or upsampling. In this case, the crafted adversarial example via the greedy algorithm-generated perturbation lies in the submanifolds close to legitimate samples, which basically noisy sample lie.

Some adversarial attacks explicitly add noise to the audio signals mainly by manipulating the frequency components [24], [25]. Backdoor attack [24] is based on adding non-linearity to a given audio signal in frequency ranges inaudible to the human hearing system (over 20 kHz). This non-linearity can be captured by unmodified microphones but does not show recognizable effects on human ends. Taking advantage of this type of attack, perturbations can be computed in frequency domain and then applied to a given audio signal. Finally, the perturbed signal can fool a machine learning model. Backdoor attack lacks in defining a general optimization formulation for computing adversarial frequency perturbations (the shadow signal) [24]. In other words, there is no analytical way for conveniently computing the perturbation. The potential perturbation value may change depending on the audio signal and therefore it makes computing proper shadow signals very cumbersome and time-consuming. Moreover, audio frequency manipulation, even if unrecognizable to humans, can be easily detected in its 2D representation (STFT, DWT, etc.). Adversarial examples generated by Backdoor attack can be easily detected by a simple post-processing module which analyzes their spectrograms. An ideal case for an adversarial example generated from a given audio signal is to be unrecognizable in both 1D and 2D representations. Similarly, DolphinAttack [25] implements phase domain manipulations that are unrecognizable by the human hearing system to change the sample label toward other than the legitimate one.

The detectability of the adversarial examples generated by Backdoor and DolphinAttack algorithms can be assessed by computing the local intrinsic dimensionality score (LID) [26] for their 2D representations. For such an aim, three groups of inputs should be defined: normal, noisy and adversarial where the latter is generated by both Backdoor and DolphinAttack algorithms. Next, each sample can be divided into mini-batches and a LID score can be calculated for each mini-batch of these three groups with respect to their corresponding normal sample, by Eq. 16:

$$\hat{\mathrm{LID}}(\mathrm{x}) = -\left(\frac{1}{k}\sum_{i=1}^{k}\log\frac{r_i(x)}{r_k(x)}\right)^{-1} \tag{16}$$

where $r_i(x)$ refers to distance between x and its nearest neighbors, $r_k(x)$ denotes to the maximum of the neighbor distances, and k is the number of neighbour samples. The computed LID scores of noisy and normal samples should be appended into negative class; and the LID scores of adversarial samples should be assigned to the positive class. Finally, a logistic regression can be trained on these two classes. The experiments carried out on 2D representations in Section V-A show that the adversarial examples generated by Backdoor and DolphinAttack can change the correct label of a given audio signal, although they cannot be categorized as adversarial attacks because of two main reasons: (i) the generated adversarial examples lie in the subspace of legitimate and random noisy signals when they basically should fall into different sub-regions; (ii) since there is not an analytical or an optimization-based approach for computing a small adversarial perturbation (δ) for high-dimensional audio, the values of δ are actually generated manually or by greedy algorithms and therefore, this highly increases the chance of detecting the adversarial signal even by a simple defense model.

As it has been discussed so far, there are many open problems in adversarial studies on raw audio signals and there is no reliable adversarial attack to 1D signals. This could also be interpreted as a good point if we disregard the fact that audio can be converted to a 2D representation (spectrogram) where strong adversarial attacks developed for images (e.g. FGSM, BIM, etc.) are quite applicable for 2D audio representations. This is a critical issue and poses a security concern for machine learning models for audio, either conventional one (e.g. SVM) or deep learning models (e.g. CNNs). However, addressing the transferability of adversarial examples from 1D audio signal to 2D audio representation (or vice versa) is out of the scope of this paper. In fact, one of our goals in this paper is to assess the resiliency of machine learning models based on different types of 2D audio representations to some strong adversarial attacks aiming to better understand their functionalities.

III. 2D Audio Representation

The vulnerability of advanced machine learning models such as CNNs and long short-term memory networks (LSTM) on 1D waveform has been studied by Carlini and Wagner [4]. They have shown the weaknesses of these models against FGSM-like adversarial attacks. However, state-of-the-art for several audio tasks, such as music genre classification [27], [28], speaker identification [29], environmental sound classification [29], etc. are based on 2D representation. This aroused our interest to evaluate the robustness of models based on 2D representation against adversarial attacks. To the best of our knowledge, the resiliency against adversarial attacks of 2D CNNs such as AlexNet and GoogLeNet, which have achieved the highest performances on environmental sound datasets, has not been studied in 2D representation spaces. To such an aim, we use Fourier and wavelet transform to convert raw audio signals into 2D representations. The first transformation is used to produce short-term frequency spectrograms for training AlexNet and GoogLeNet [11]. We also use wavelet transform for producing more informative spectrograms, which after some pre-processing steps are used in the proposed approach to train an SVM classifier. A brief description of these two types of spectrogram is presented as follows.

Considering an audio signal $x[n]$, where $n{=}0, 1, \ldots, N{-}1$ denotes the number of samples and its decomposed signal X using Fourier (time-frequency) transform using $\{g_{l,k}\}_{l,k}$ atoms, as:

$$X[l,k] = \langle x, g_{l,k}\rangle = \sum_{n=0}^{N-1} x[n] g^*_{l,k}[n] \tag{17}$$

where the operator $*$ denotes the complex conjugate, and l, k are time and frequency localization indices, respectively. This representation is widely used in sound and speech processing [30], [31]. Given a Hanning window $H[n]$ of size K which is shifted by a step $u{\leq}K$, then the function $\{g_{l,k}\}_{l,k}$ in Eq. 17 can be written as [32]:

$$g_{l,k}[n] = H[n - lu] \exp\left(\frac{i2\pi kn}{K}\right) \tag{18}$$

Time and frequency (scale) indices abide $0{\leq}l{\leq}N/u$ and $0{\leq}k{\leq}K$. Finally, the Fourier spectrogram is represented as:

$$\mathrm{Spec}^f[l,k] = \log|F[l,k]| \tag{19}$$

The final appearance of a spectrogram depends on the parameters l and k specified in Eqs. 18 and 19. Similar to this transform is the continuous wavelet transform (CWT) as denoted in Eq. 20:

$$CWT(l,k;x(t),\psi(t)) = \frac{1}{\sqrt{l}} \int_{-\infty}^{+\infty} x(t)\psi(\frac{t-k}{l})dt \qquad (20)$$

where $\psi(t)$ denotes the mother wavelet and l, k, and t denote scale, translation and time, respectively. The discretized representation of CWT is given by Eq. 21, where it is determined on a grid of m scales and n discrete time with dilation parameter k.

$$DWT(m,n) = 2^{m/2} \sum_{k=0}^{n-1} x(k)\psi(2^m, k-n) \qquad (21)$$

For ψ, we use Morlet function which is defined in Eq. 22 where β is set to 0.8431:

$$\psi(t) = e^{-(\beta^2 t^2)/2} \cos(\pi t) \qquad (22)$$

Finally, the wavelet spectrogram can be obtained as:

$$\text{Spec}^w[l,k] = |DWT(l,k)|^2 \qquad (23)$$

In summary, for an audio signal $x[n]$, there will be two different 2D representations: Spec^f and Spec^w. Moreover, for Spec^w we use three scales for the magnitude, which provide different visualization schemes: linear, logarithmic, and logarithmic real. Linear scale highlights high-frequency magnitudes which represent high variation areas in the spectrogram. Logarithm scale highlights low-frequency information which expands distance of magnitudes in different scales. Finally, logarithm real scale highlights the energy of the signal which is related to the signals mean.

IV. A Robust Approach for 2D Audio Representation and Classification

In general, the current approaches for audio classification are able to achieve high recognition rates but they are vulnerable to adversarial attacks, what means that they can be easily fooled. Therefore, our aim is to design a novel approach for audio classification that provides a good trade-off between recognition accuracy and low vulnerability to some of the most threatening adversarial attacks. The proposed approach for environmental sound classification has three main parts: spectrogram preprocessing, feature extraction, and classification. Fig. 2 presents an overview of the proposed preprocessing approach which, given an audio signal produces three spectrogram representations as output. The audio signal undergoes through color compensation, highboost filtering, dimensionality reduction, and smoothing and at the end, we have three enhanced spectrograms. Next, speeded up robust features (SURF) are extracted from zoning blocks that slide over the spectrograms as shown in Fig. 3. The geometrical distance of feature vectors is maximized by a K-means++ algorithm and finally a multiclass SVM trained on such features makes the prediction.

A. Spectrogram Preprocessing

The goal of the spectrogram preprocessing is threefold: (i) improve the recognition accuracy; (ii) improve the robustness against adversarial attacks; (iii) and augmenting the number of samples. It starts by color compensation of the spectrogram Spec(i,j) by mapping each spectrogram to three different color spaces: black-blue-green (BBG), purple-gold (PG), and white-black (WB) as shown in Fig. 4. Empirically, these compensations boost and improve the final classification performance because they affect pixel intensity values, though keeping their distribution. The second preprocessing operation is highboost filtering [33] which enhances color compensated spectrograms focusing on their high-frequency elements while maintaining low-frequency components. This filtering is given in Eq. 24.

$$\text{Spec}_{enh} = (F_{ap} + cF_{hf}) \times \text{Spec}_{org} \qquad (24)$$

where Spec$_{enh}$ and Spec$_{org}$ denote the enhanced and original spectrograms, respectively. Also, F_{hf} represents the high frequency band-pass filter (5×5 Laplacian operator) which is multiplied by constant value c which acts as a scaling factor, and F_{ap} denotes an all-pass filter.

The three spectrogram representation and color compensations increase in nine times the total number of samples into the datasets in addition to the pitch-shifting augmentation that is also applied, but on the 1D signal prior to the spectrogram representation. Pitch-shifting increases by eight times the number of samples. Therefore, to alleviate the computational complexity both in execution and storage, we reduce the dimensionality of the spectrograms. Though, there are many algorithms for such an aim, we use singular value decomposition (SVD) because of its pivotal properties in reducing the dimensionality of 2D data without changing the perceived visual appearance, if the reduction rank is chosen appropriately. Somewhat similar to the Fourier transform, SVD can describe a 2D matrix by basis functions in such a way that, linear combination of those function can reconstruct the matrix [34]. Basis functions in Fourier transform are sine and cosine, but SVD produces individual basis matrices for each given input. For an enhanced spectrogram Spec$_{enh}$, SVD decomposes it as shown in Eq. 25.

$$\text{Spec}_{enh} = \sum_{i=1}^{m} S_i U_i V_i^\top \qquad (25)$$

where S, U, and V are derived matrices from decomposing Spec$_{enh}$ into singular value, hanger, and aligner matrices, respectively. Also, m is the minimum dimension of the spectrogram either in width or height. The matrix S is diagonal and its elements are in descending order which indicates the importance of hanger and aligner column vectors. The basis functions associated with Spec$_{enh}$ are the product of U_i and $V_i^\top$ weighted by S_i. This gives us the capability of reconstructing Spec$_{enh}$ by its most important components, from low to high frequency components. By setting the m in Eq. 25 to m/n where n>1, we can make a balance between dimensionality reduction and quality of reconstruction. This

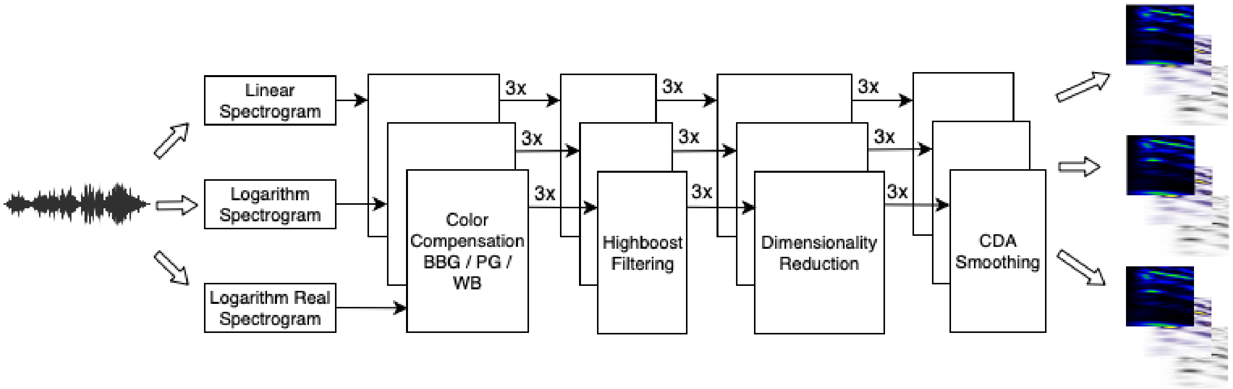


Fig. 2: Block diagram of spectrogram generation and preprocessing. From a single audio waveform, three spectrogram representations are generated and processed through several blocks with the aim of enhancing the 2D representation.

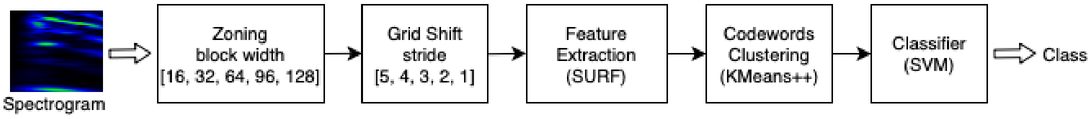

Spectrogram
Zoning block width [16, 32, 64, 96, 128]
Grid Shift stride [5, 4, 3, 2, 1]
Feature Extraction (SURF)
Codewords Clustering (KMeans++)
Classifier (SVM)
Class

Fig. 3: Block diagram of the proposed classifier. Values in the first block indicate sizes of square zones (blocks) from 16×16 to 128×128. Stride values in the second block correspond to the zone sizes in the first block. For instance, a 96×96 block has stride 2, and so on.

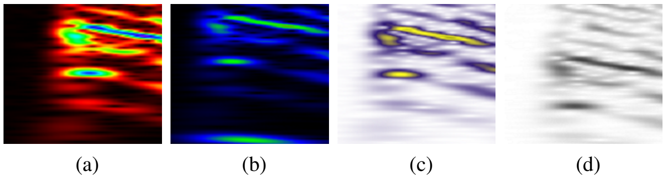


Fig. 4: Three color compensation: (a) original spectrogram; (b) black-blue-green (BBG); (c) purple-gold (PG); (d) white-black (WB).

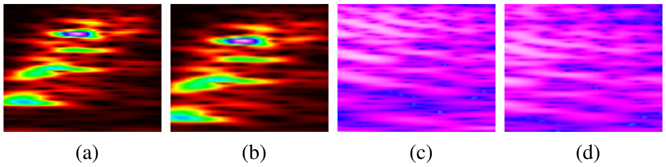
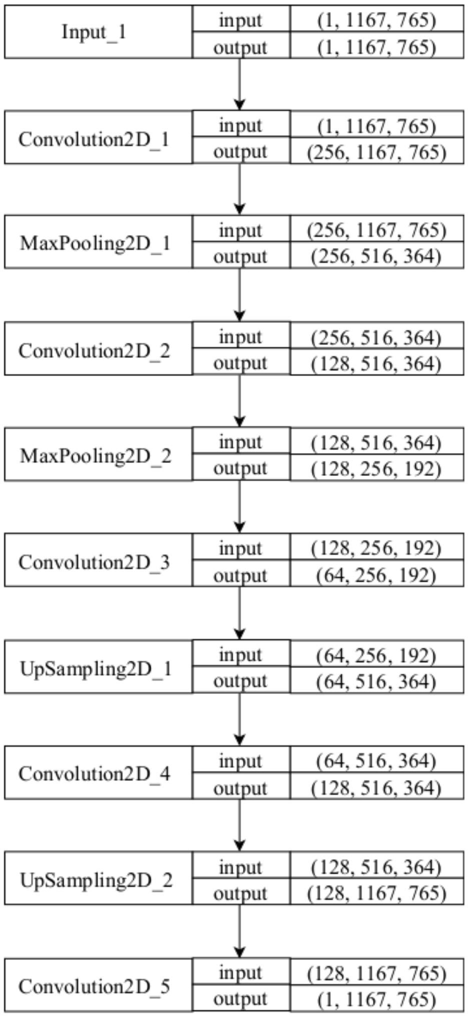


Fig. 5: Dimension reduction effect: (a) linear magnitude representation; (b) reconstruction of (a) after reduction in half; (c) logarithmic magnitude representation; (d) reconstruction of (c) after reduction in half.

operation actually acts as principal component analysis [35]. Empirically, the magnitudes of the matrix S will be less than the pixel precision ($1/255$ for an 8-bit representation) at indices around n=2 (in Eq. 25) and therefore they can be pruned without any visual impact on the spectrograms. Though this dimension reduction resizes spectrogram dimension to half, the quality of the reconstructed image is quite good, and differences are imperceptible to the human visual system (see Fig. 5). The outputs of the dimensionality reduction block in Fig. 2 are linear, logarithmic, and logarithmic real spectrograms visualized in three color spaces (BBG, PG, and WB) which are all reduced to half of its original dimension.

Though highboost filtering enhances high frequency components in spectrograms and therefore it leads to a better feature extraction, it may also boost noise, especially for the PG and WB color compensated representations. This problem can be minimized to some extent by the dimensionality reduction by SVD, but it is still necessary to improve the quality of the final compensated representations of spectrograms. For addressing this issue, highboost filtered spectrograms are smoothed using a denoising autoencoder with three convolution layers [36].

The main advantage of convolutional denoising autoencoder (CDA) over traditional smoothing algorithms is its flexibility in data adaptation and fine reconstruction. Besides, another important reason for using the CDA is to make spectrograms more robust against small adversarial perturbations which machine learning models are very sensitive to. The architecture of the proposed CDA is data-dependent and it is depicted in Fig. 6. Such architecture considers as input, spectrograms with dimensions 1167×765. For corrupting the input data, we used the spectrograms derived from SVD as well as the technique introduced by Vincent *et al.* [37]. The architecture of the encoder shown in Fig. 6 has three convolutional layers with 5×5 receptive fields, stride 1, *relu* activation function and dropout of 0.5, and two max pooling layers. Finally, after all steps of preprocessing, the enhanced spectrograms are ready to undergo to feature extraction and classification, as described in the following subsection. Besides that, the enhanced spectrograms can also be used with pre-trained CNN architectures such as AlexNet or GoogLeNet, as described in Section V.



Layer		
Input_1	input	(1, 1167, 765)
	output	(1, 1167, 765)
Convolution2D_1	input	(1, 1167, 765)
	output	(256, 1167, 765)
MaxPooling2D_1	input	(256, 1167, 765)
	output	(256, 516, 364)
Convolution2D_2	input	(256, 516, 364)
	output	(128, 516, 364)
MaxPooling2D_2	input	(128, 516, 364)
	output	(128, 256, 192)
Convolution2D_3	input	(128, 256, 192)
	output	(64, 256, 192)
UpSampling2D_1	input	(64, 256, 192)
	output	(64, 516, 364)
Convolution2D_4	input	(64, 516, 364)
	output	(128, 516, 364)
UpSampling2D_2	input	(128, 516, 364)
	output	(128, 1167, 765)
Convolution2D_5	input	(128, 1167, 765)
	output	(1, 1167, 765)

Fig. 6: Architecture of the proposed CDA to smooth the spectrogram representations. Numbers denote the number of filters, and width and height of each spectrogram, respectively.

B. Feature Extraction and Classification

The feature extraction and classification approach includes five steps as depicted in Fig. 3. The main idea of our classification approach is firstly extracting features from a static sized moving aperture (a.k.a. grid shifting block) which spans a spectrogram with a dynamic stride within a block with dynamic size. Secondly, maximizing the geometrical distance among feature vectors with different class labels and finally training an SVM classifier on the organized feature space, which is a quadratic optimization problem that avoids local minimum issues. The proposed approach aims of achieving both recognition accuracy and robustness against adversarial attacks. Therefore, we evaluated several handcrafted features and representation learning methods to finally came up with SURF instead of CNN features. Empirically, this feature encoding scheme outperforms deep neural network features (with/without convolution layers) both in terms of recognition accuracy and robustness against adversarial attacks. Our main hypothesis relies on the nature of these features which are projected gradients compared to CNN features. Generally, features generated by CNNs are accurate and boost the recognition accuracy but empirically have a negative effect on the robustness and the final model is quite vulnerable to adversarial attacks.

The first step is zoning, which breaks a given spectrogram into blocks that may vary from 16×16 to 128×128 pixels. Empirically, a block size of 16×16 is small enough for capturing subtle pixel density changes and a block size of 128×128 is preferable for areas with less high frequency components. Then, a shifting grid of static size of 8×8 will span through them. It should be noted that, there is no preprocessing step to understand which parts of a given spectrogram have low or high frequency components. Therefore, we simply span the spectrogram with variational sizes of grid shifting block and there is no heuristic algorithm involved in either. The stride of shifting grid varies from five to one with respect to the block size where the grid shifts in smaller stride on larger blocks. This scheme supports the idea of detail scanning of spectrograms aiming at extracting more distinguishable features. Different values have been tested for the stride size and finally its minimum and maximum obtained to one and five, respectively (see Fig. 7).

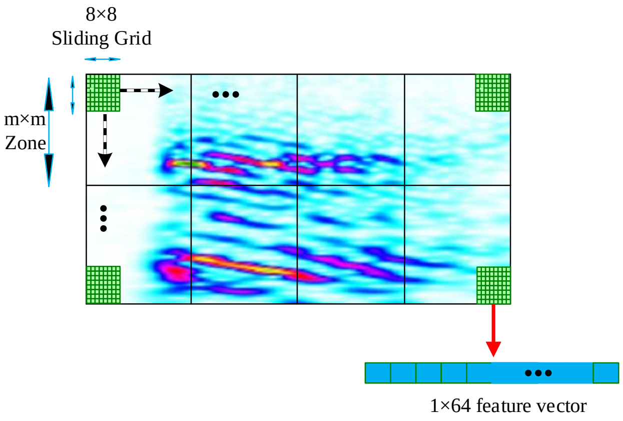


Fig. 7: Sample grid shifting illustration over a given spectrogram. The sliding grid shifts through the square zone with different values of pixel strides.

Different features could be extracted from each 8×8 grid. We firstly evaluated scale invariant feature transform as feature extractor (SIFT) [38] and then utilized SURF [39]. Although SURF provides fewer feature vectors compared to SIFT, it is much faster in run-time. We applied SURF on shifting grids within each zone as shown in Fig. 3. At the end, each spectrogram zone is represented by a 64-dimensional feature vector. For increasing the geometrical distance among extracted feature vectors which are not of the same class, an unsupervised algorithm K-means++ [40] has been implemented. This algorithm clusters codewords (feature vectors) into an organized distribution with respect to their geometrical linear distance (D_i). Kernel centroids are selected randomly and their distances are computed with respect to their local seed following the probability proportional to D_i^2. We refer readers to [41] for further details. Finally, we train a multiclass support vector machine (SVM) classifier with polynomial kernel on the codewords. We have also evaluated the SVM with radial basis function (RBF) kernel on the clusters which could not improve the accuracy. In the following section we evaluate the proposed approach on the three datasets and compare the results with other state-of-the-art approaches.

V. Experimental Results

We have carried out several experiments on three benchmarking datasets with the aim of: (i) evaluate the detectability of the current adversarial attacks for 2D audio representations; (ii) assessing the performance of the proposed approach on the enhanced spectrograms and benchmark it against other deep architectures that have been used for audio classification, such as AlexNet and GoogLeNet; (iii) evaluating the resiliency of the proposed approach and the two deep architectures against several types of adversarial attacks; (iv) characterize the transferability of the adversarial audio attacks across two different classification paradigms, CNNs and SVMs.

The UrbanSound8k dataset has 8,732 audio samples shorter than four seconds of 10 classes (air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren, and street music). The ESC-50 dataset includes 2,000 5-second samples of 50 classes including major groups of animals, natural sound capes & water sounds, human non-speech sounds, domestic sounds, and exterior noises. The ESC-10 dataset is a subset of ESC-50 which includes 400 recordings of 10 classes (dog bark, rain, sea waves, baby cry, clock tick, person sneeze, helicopter, chainsaw, rooster, and fire crackling).

A. Detectability of Adversarial Audio Attacks

The definition of an adversarial attack relies on whether the attack is easily identified or not. We have carried out some experiments to evaluate two of the most threaten attacks on audio: Backdoor and the DolphinAttack adversarial attacks. For such an aim, we generated STFT, DWT, and CRP spectrograms for the audio recordings of the UrbanSound8K dataset, and computed the LID score considering different values of k as shown in Eq. 16. Basically, the LID score should be able to discriminate between negative and positive classes which means returning higher values. In other words, small values of the LID score denote indistinguishable difference between positive and negative classes which can in turn be interpreted as positive classes may not be considered as adversarial. Tab. I shows that the differences between LID scores of positive and negative classes are quite small and also shows that the logistic regression classifier trained on these classes has a very low recognition accuracy. As Tab. I shows, legitimate, noisy, and adversarial examples lie in the same subspace and in fact they lie in the same manifolds because they have very similar LID scores. In other words, the adversarial examples generated by both Backdoor and DolphinAttack are almost equivalent to examples corrupted by random noise in terms of functionality, which basically does not seem to satisfy the generic definition of adversarial examples. Moreover, the performance of the logistic regression is quite low and shows poor discrimination between negative and positive classes which desirably should be higher than 60%.

TABLE I: LID score for different representations of UrbanSound8K samples. Mean difference is generated for two classes of negative (legitimate and random noisy) and positive (adversarial by Backdoor and DolphinAttack).

Representation	k	Mean Difference of LID Scores	Recognition Accuracy (%)
DWT	50	0.082	11.23
	75	0.071	10.04
	100	0.036	10.01
	125	0.032	09.46
STFT	50	0.076	13.05
	75	0.074	12.94
	100	0.066	12.92
	125	0.061	11.87
CRP	50	0.089	15.01
	75	0.084	14.56
	100	0.079	14.32
	125	0.078	13.77

B. Accuracy and Resilience of CNNs and SVMs

Deep neural networks require a large amount of data for training. For increasing the size of datasets aiming at extracting more information from them, we augmented the number of samples by stretching (speeding up) and shrinking (slowing down) recordings in time (pitch shifting) using MUDA library [42], which includes highly optimized functions for sound data augmentation. This is a reliable common approach in sound processing which affects favourably the classifier's performance [9]. The scale values that were applied for pitch-shifting are: 0.5, 0.75, 0.9, 1.1, 1.25, 1.5, and 1.75. This operation boosts the size of each dataset in eight times.

TABLE II: Scale operators (c) for color compensations.

Dataset	Color Compensation	c
ESC-10	BBG	0.57
	PG	0.74
	WB	0.46
ESC-50	BBG	0.81
	PG	0.79
	WB	0.58
UrbS8k	BBG	0.72
	PG	0.85
	WB	0.67

For generating the spectrogram Spec^f, we used the approach suggested by Boddapati *et al.* [11] by setting sampling frequency to 8 kHz, 16 kHz, and 8 kHz for ESC-10, ESC-50, and UrbanSound8K datasets, respectively. Also, the frame length was set to 50 ms (ESC-10), 30 ms (ESC-50), and 50 ms (UrbanSound8K) with a fixed overlapping of 50%. These values have been found after conducting exploratory experiments on these datasets. For generating the spectrogram Spec^w, we used 256 frequency bins with a Morlet mother function as proposed by Cowling and Sitte [43] and linear, logarithmic, and logarithmic real magnitude scales for enhancing high, low and medium frequencies, respectively. The scale operators c as described in Eq. 24, are shown in Tab. II. The SVM uses a quadratic kernel as $k(x, l) = (\gamma \cdot x \cdot l \cdot + c)^2$ where x and l denote the input sample and its label, respectively. Also, $\|c\| \leq 0.1$ is the cost parameter and $\|\gamma\| < 0.003$ is the kernel parameter. Besides the quadratic SVM, we also evaluated a linear SVM, which is referred simply as SVM in several tables in this section. We have used the scikit-learn [44] package for implementing SVMs.

In the first experiment, we trained AlexNet and GoogLeNet with the same setup proposed by Boddapati *et al.* [11] which leads to the highest recognition performance reported in the literature for 2D representations. These two deep convolutional neural networks were trained on the 2D representation (M), which consists of a linear pooling of STFT (M_{FR}), MFCC (M_{MFCC}), and CRP (M_{CRP}) spectrograms, as denoted by Eq. 26:

$$M = clip\left(M_{\text{FR}}(i,j) + M_{\text{MFCC}}(i,j) + M_{\text{CRP}}(i,j), [0,1]\right) \tag{26}$$

These matrices are derived from the environmental sound datasets (ESC-10, ESC-50, and UrbanSound8K). The augmentation procedure is also applied in the same way we did for audio waveforms, but with different values for pitch shifting. In addition, to training our classifier on the pooled representation space (a.k.a. pooled), we trained it on the proposed 2D representation space as shown in Fig. 2 (a.k.a. DWT). These two representations are also evaluated for AlexNet and GoogLeNet networks. In other words, we measure the performance of AlexNet and GoogLeNet on the spectrograms obtained from our data preprocessing approach. These two experiments are implemented using 5-fold cross validation with a ratio of 0.2 for testing. We used four parallel GPUs GTX580 based on an implementation proposed in [45]. We stopped training after 83 epochs using early stopping scenario for AlexNet and GoogLeNet. The results achieved by these two classifiers are reported in Tab. III. As Tab. III shows, AlexNet and GoogLeNet have achieved the best performances for both representation spaces, although the proposed approach presents competitive results. The differences between the best deep model and the proposed approach range from 4.32% for UrbanSound8K to 11.02% for ESC-50. We also repeated this experiment with 10-fold cross validation as specified in the first CNN attempt on the UrbanSound8k dataset in [9]. The results were almost the same as reported in Tab. III.

TABLE III: Mean classification accuracy (5-fold CV) of four classifiers on two representation spaces (pooled and DWT).

Dataset	Repres.	Mean Accuracy (%)			
		GoogLeNet	AlexNet	SVM	Proposed
ESC-10	Pooled	**83.19**	82.54	64.23	78.31
	DWT	**83.21**	82.90	70.45	79.10
ESC-50	Pooled	**71.36**	64.09	52.37	60.10
	DWT	**71.20**	66.41	55.09	60.41
UrbS8k	Pooled	**91.08**	90.06	72.03	86.15
	DWT	86.85	**90.10**	72.89	86.39

However, a high accuracy does not translate to a high robustness against adversarial attacks. In Tab. IV, we assess the robustness of the classifiers of Tab. III against several adversarial attacks as well as the transferability of such adversarial attacks across different models. For such an aim we have developed the FGSM, BIM-a, BIM-b, and CWA adversarial attacks (deep model attacks) for AlexNet and GoogLeNet and the EA and regular Evasion attacks (SVM attacks) for SVM classifiers. The total number of adversarial examples crafted using each attack for different datasets is equivalent to the number of samples in the legitimate dataset. In other words, for each legitimate sample, one adversarial example is crafted by each adversarial attack algorithm. Since FGSM and CWA are targeted, adversarial examples of these two attacks are crafted toward a random wrong label. This not only makes our evaluations fair against non-targeted attacks, but also reduces the cost of crafting adversarial examples of datasets with more than 10 classes, which is the case of the ESC-50 dataset that has 50 classes. Then, these crafted examples are fed to both deep learning and SVM models to compute the ratio of successful fooling over the total number of adversarial examples (fooling rate) in a black-box scenario.

TABLE IV: Mean fooling rate (5-fold CV) of two CNNs and two SVMs against six strong adversarial attacks. The best performances are shown in bold (lowest values).

Dataset (Repres.)	Adv. Attack	Mean Fooling Rate (%)			
		GoogLeNet	AlexNet	SVM	Proposed
ESC-10 (Pooled)	FGSM	95.23	94.04	60.78	**43.12**
	BIM-a	94.07	90.13	61.68	**48.60**
	BIM-b	94.26	91.30	62.46	**46.03**
	CWA	95.89	93.66	94.01	**51.77**
	LFA	**51.23**	63.01	94.43	60.47
	EA	**43.79**	44.12	94.14	58.34
ESC-10 (DWT)	FGSM	94.30	93.36	64.05	**50.02**
	BIM-a	92.15	92.87	59.57	**51.13**
	BIM-b	93.58	92.33	57.92	**43.07**
	CWA	95.36	94.89	64.35	**53.18**
	LFA	57.36	**56.35**	95.58	71.64
	EA	49.66	**48.00**	92.89	61.78
ESC-50 (Pooled)	FGSM	96.78	95.61	69.22	**51.99**
	BIM-a	95.01	96.08	67.17	**50.20**
	BIM-b	94.77	95.17	69.71	**50.03**
	CWA	96.02	97.14	72.10	**53.04**
	LFA	62.12	66.35	95.27	**60.25**
	EA	55.47	**52.01**	95.94	59.03
ESC-50 (DWT)	FGSM	96.30	95.80	66.16	**50.01**
	BIM-a	93.36	94.05	69.02	**49.36**
	BIM-b	91.25	92.53	67.11	**45.92**
	CWA	95.73	94.11	70.09	**49.31**
	LFA	60.08	**58.01**	92.21	62.84
	EA	51.37	**49.61**	90.36	58.15
UrbS8k (Pooled)	FGSM	94.68	93.22	60.50	**45.17**
	BIM-a	94.65	95.32	58.22	**42.36**
	BIM-b	90.22	91.24	53.39	**42.16**
	CWA	92.08	93.62	**60.17**	60.25
	LFA	**55.01**	78.36	96.14	65.35
	EA	44.02	**41.07**	95.16	62.30
UrbS8k (DWT)	FGSM	94.14	93.02	57.31	**48.33**
	BIM-a	92.43	93.21	62.01	**51.07**
	BIM-b	94.01	93.61	63.32	**53.03**
	CWA	95.27	93.84	62.14	**50.48**
	LFA	**54.33**	55.03	92.06	63.52
	EA	47.01	**45.50**	91.02	59.01

Tab. IV also shows the transferability of crafted deep attacks to SVM models and correspondingly SVM attacks to AlexNet and GoogLeNet. A high adversarial transferability rate represents a serious threat for data-driven classifiers. In other words, a reliable classifier should not only be robust against adversarial attacks designed to cause its type of model to make a mistake, but it should also be reasonably resistant against attacks designed to attack other types of model. Tab. IV shows the results achieved on both experiments. The mean fooling rate, which measures the success rate of adversarial examples in fooling the machine learning models in terms of the percentage of adversarial samples misclassified by the models is computed for comparing the performance of CNNs and SVMs against the six adversarial attacks. Tab. IV shows that all classifiers are quite vulnerable against the adversarial attacks designed to attack its own model, with fooling rates higher than 90% for both CNNs and SVM. On the other hand, the proposed approach not only is quite robust, but also has the lowest fooling rate against adversarial attacks (EA and LFA) designed for such a model, with fooling rates between 58.15% and 71.64%. Tab. IV also reveals that, there is a higher chance of fooling SVM models by deep attacks compared to fooling AlexNet and GoogLeNet by adversarial examples crafted by EA or LFA. Additionally, AlexNet is more robust against SVM-based adversarial attacks compared to GoogLeNet, though its recognition accuracy is a little lower than GoogLeNet.

TABLE V: Average ranking considering the accuracy and fooling rate for all models, datasets and adversarial attacks.

Approach	Accuracy		Fooling	
	$\bar{r}$	Rank	$\bar{r}$	Rank
GoogLeNet	1.17	1	2.97	4
AlexNet	1.83	2	2.78	3
SVM	4.00	4	2.67	2
Proposed	3.00	3	1.61	1

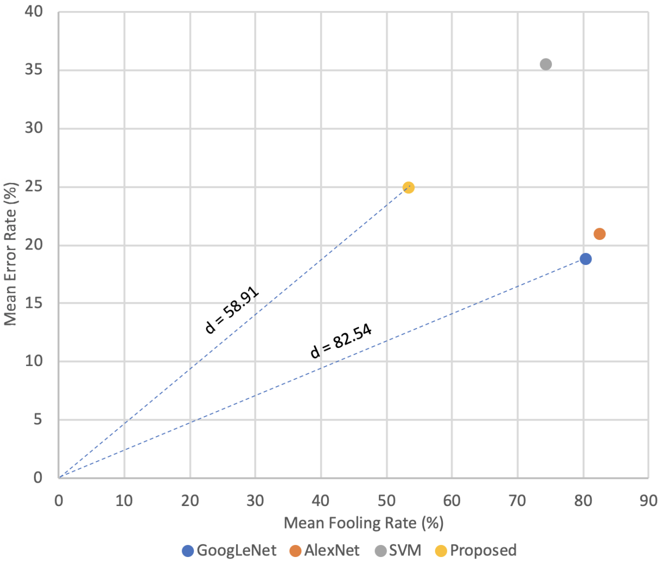


Fig. 8: Comparison between the deep models, SVM, and the proposed approach over all datasets and representations of Tab. III. The Euclidean distance to the origin for the two best approaches is also shown.

Tab. V shows average rankings of our evaluation metrics of recognition accuracy and fooling rate with respect to the statistics provided in Tab. IV. Regarding this table, the smaller the $\bar{r}$ is, the better are the recognition and the fooling rates. Although the proposed approach comes third rank in recognition accuracy, it is the first in resiliency against the six types of adversarial attacks. Therefore, this indicates a good trade-off between accuracy and resiliency. This is also shown in Fig. 8, where the proposed approach is the one closest to the origin (zero error rate and zero fooling rate) according to the Euclidean distance (d = 58.91). Fig. 8 also shows that while the mean error rate of the proposed approach is 6.07% higher than GoogLeNet, the proposed approach is 26.96% more robust to adversarial attacks than GoogLeNet. Furthermore, the mean error rate of the proposed approach is 10.57% lower than the SVM and it is also 20.98% more robust to adversarial attacks. Notwithstanding the good trade-off achieved by the proposed approach, there is still a large room for improvements.

C. Analysis of the Proposed Approach

The proposed approach provides the best trade-off between accuracy and resilience to adversarial attacks than deep models and SVM. For understanding the reason(s) of such a best trade-off, we dig into the preprocessing (Fig. 2) of the proposed approach. We safely remove each module (or submodule) from the preprocessing part and measure its positive or negative contribution to the recognition accuracy and robustness against the six types of adversarial attacks. Tab. VI reveals that the propose approach benefits from both CDA and SVD compression. The most straightforward impact of these two operations is affecting (smoothing) high frequency components where subtle changes of adversarial examples probably lie on. It has been proved that utilizing autoencoders as a reactive defense scheme can clean adversarial examples and therefore defend the targeted trained models from the adversarial attacks [46], [47]. Moreover, for measuring the effect of the first two modules of Fig. 3 on final classification performance, we carried out some additional experiments including removing them and changing block size and grid shifting stride on a 5-fold cross validation. In Tab. VII, we only report some of the top recognition accuracy with respect to block size and stride length.

TABLE VI: The average effect of removing each module from Fig. 2 on the classification and robustness of the proposed model against deep and SVM adversarial attacks. Positive (+) and negative (−) effects are shown by their signs.

Module	Mean Accuracy (%)	Robustness Against (%)	
		SVM Attacks	Deep Attacks
Spectr. Vis.	−16.47	−4.07	−3.14
Color Comp.	−7.36	−0.36	+2.64
Highb. Filt.	−9.52	−0.75	−1.96
SVD	−8.21	−6.18	−4.17
CDA	−7.94	−9.18	−6.33

VI. Conclusion

In this paper, we discussed the serious threat that adversarial attacks may pose to machine learning models trained either on 1D or 2D audio representations. While there is no reliable adversarial attack on raw audio signals, there is a bijective

TABLE VII: The effect of selected zoning size and shifting grid length on the overall recognition accuracy of the proposed approach on DWT representation of the UrbanSound8K dataset.

Zoning Size	Grid Shifting Stride	Mean Accuracy (%)
[16, 32, 64, 96, 128]	[1, 2, 3, 4, 5]	**79.33**
[16, 32, 64, 96, 128]	[2, 2, 2, 2, 2]	77.29
[8, 16, 32, 64, 128]	[1, 2, 3, 4, 5]	76.18
[16, 32, 64, 96, 128]	[4, 3, 3, 3, 4]	74.22
[32, 64, 128]	[3, 2, 1]	73.91
[64, 96, 128]	[3, 2, 1]	72.63
None	None	70.92

relation between 1D signals and spectrograms which opens the avenue for adversarial transferability between these two representation spaces and that poses a real security concern. Besides that, considering that the majority of state-of-the-art approaches for audio classification rely on 2D representations, most of them based on CNNs originally designed for image classification tasks, we showed that CNNs trained on 2D representations of environmental sound signals achieve state-of-the-art performance in terms of accuracy. However, these CNNs are not reliable at all, as they can be easily fooled by adversarial examples, with fooling rates higher than 90%.

Therefore, we proposed a novel approach for environmental audio classification based on 2D representations that provides a good trade-off between accuracy and resiliency to the most threatening adversarial attacks designed to fool both deep neural models and SVMs. The proposed approach was compared to AlexNet, GoogLeNet, and a linear SVM classifier on three publicly available datasets. The highest mean recognition rates were achieved for GoogLeNet (81.15%), AlexNet (79.15%), the proposed approach (75.08%), and the linear SVM (64.51%), respectively. However, in addition to the competitive recognition performance, the proposed approach outperforms by far all three mentioned classifiers in terms of robustness against adversarial attacks since the mean fooling rates for these four models are 95.15%, 94.36%, 50.56%, and 66.74% by deep attacks and 52.62%, 54.79%, 61.89%, and 93.77% by SVM attacks respectively. However, as shown in Fig. 8, there is still a large room for improvements. As a future study, we are really interested in to employ reactive adversarial detection algorithms (e.g., LID detector) as a postprocessing operation aiming at increasing the robustness of the proposed approach.

We are also really inclined to explore the resiliency of our classification scheme for raw audio signals rather than spectrograms against audio attacks and measure its capability against audio played back over the air. To this end, we may need to remove/add some of our preprocessing steps which have shown positive impacts on the robustness of the proposed approach against adversarial attacks (e.g. CDA); and consequently, simplify our approach which requires several steps of processing. Another important aspect that deserves further studies is the adversarial example transferability bijectively from 1D audio signal to 2D spectrograms and vice versa. In other words, we would like to explore the possibility of crafting an adversarial audio for a model trained on 1D signals in which its 2D representation be able to fool the model trained on spectrograms, also the other way around. Since many audio classification approaches implement different types (ensemble) of data-driven models (both 1D and 2D) aiming at improving their prediction confidence, hence if a crafted adversarial example can fool both 1D and 2D models, it constitutes a constant real-life threat to all sound recognition/processing systems and devices (e.g. voice id devices).

References

[1] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv Prepr arXiv:1607.02533*, 2016.

[2] S. Sabour, Y. Cao, F. Faghri, and D. J. Fleet, "Adversarial manipulation of deep representations," *arXiv Prepr arXiv:1511.05122*, 2015.

[3] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille, "Adversarial examples for semantic segmentation and object detection," in *IEEE Intl Conf Comp Vis*, 2017, pp. 1369–1378.

[4] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," *arXiv Prepr arXiv:1801.01944*, 2018.

[5] J. Salamon and J. P. Bello, "Unsupervised feature learning for urban sound classification," in *Intl Conf Acous Speech Sign Proc*, 2015, pp. 171–175.

[6] K. J. Piczak, "Esc: Dataset for environmental sound classification," in *23rd ACM Intl Conf Multim*, 2015, pp. 1015–1018.

[7] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *22st ACM Intl Conf Multim*, Orlando, FL, USA, 2014.

[8] J. Salamon and J. P. Bello, "Feature learning with deep scattering for urban sound analysis," in *23rd Europ Sign Proc Conf*, 2015, pp. 724–728.

[9] ——, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Sign Proc Lett*, vol. 24, no. 3, pp. 279–283, 2017.

[10] Y. Aytar, C. Vondrick, and A. Torralba, "Soundnet: Learning sound representations from unlabeled video," in *NIPS*, 2016, pp. 892–900.

[11] V. Boddapati, A. Petef, J. Rasmusson, and L. Lundberg, "Classifying environmental sounds using image recognition networks," *Procedia Comp Sci*, vol. 112, pp. 2048–2056, 2017.

[12] T.-W. Weng, H. Zhang, P.-Y. Chen, J. Yi, D. Su, Y. Gao, C.-J. Hsieh, and L. Daniel, "Soundnet: Learning sound representations from unlabeled video," in *6th Intl Conf Learn Repres*, 2018.

[13] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv Prepr arXiv:1412.6572*, 2014.

[14] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *IEEE Symp Secur Priv*, 2017, pp. 39–57.

[15] Y. Li and Y. Gal, "Dropout inference in bayesian neural networks with alpha-divergences," *arXiv Prepr arXiv:1703.02914*, 2017.

[16] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Joint Europ Conf Mach Learn Knowl Discov Datab*, 2013, pp. 387–402.

[17] H. Xiao, H. Xiao, and C. Eckert, "Adversarial label flips attack on support vector machines." in *ECAI*, 2012, pp. 870–875.

[18] C. Xie, Z. Zhang, J. Wang, Y. Zhou, Z. Ren, and A. Yuille, "Improving transferability of adversarial examples with input diversity," *arXiv Prepr arXiv:1803.06978*, 2018.

[19] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," *arXiv Prepr arXiv:1611.02770*, 2016.

[20] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," *arXiv Prepr arXiv:1605.07277*, 2016.

[21] N. Das, M. Shanbhogue, S.-T. Chen, L. Chen, M. E. Kounavis, and D. H. Chau, "Adagio: Interactive experimentation with adversarial attack and defense for audio," *arXiv Prepr arXiv:1805.11852*, 2018.

[22] M. Alzantot, B. Balaji, and M. Srivastava, "Did you hear that? adversarial examples against automatic speech recognition," *arXiv Prepr arXiv:1801.00554*, 2018.

[23] T. Du, S. Ji, J. Li, Q. Gu, T. Wang, and R. Beyah, "Sirenattack: Generating adversarial audio for end-to-end acoustic systems," *arXiv Prepr arXiv:1901.07846*, 2019.

[24] N. Roy, H. Hassanieh, and R. Roy Choudhury, "Backdoor: Making microphones hear inaudible sounds," in *15th Intl Conf Mob Sys App Serv*, 2017, pp. 2–14.

[25] L. Song and P. Mittal, "Inaudible voice commands," *arXiv Prepr arXiv:1708.07238*, 2017.

[26] X. Ma, B. Li, Y. Wang, S. M. Erfani, S. Wijewickrema, M. E. Houle, G. Schoenebeck, D. Song, and J. Bailey, "Characterizing adversarial subspaces using local intrinsic dimensionality," *arXiv Prepr arXiv:1801.02613*, 2018.

[27] C. Liu, L. Feng, G. Liu, H. Wang, and S. Liu, "Bottom-up broadcast neural network for music genre classification," *arXiv Prepr arXiv:1901.08928*, 2019.

[28] Y. M. G. Costa, L. E. S. Oliveira, A. L. Koerich, F. Gouyon, and J. G. Martins, "Music genre classification using LBP textural features," *Sign Proc*, vol. 92, no. 11, pp. 2723–2737, 2012.

[29] S. Sengupta, G. Yasmin, and A. Ghosal, "Speaker recognition using occurrence pattern of speech signal," in *Recen Trends Sign Image Proc*. Springer, 2019, pp. 207–216.

[30] G. Yu, S. Mallat, and E. Bacry, "Audio denoising by time-frequency block thresholding," *IEEE Trans Sign Proc*, vol. 56, no. 5, pp. 1830–1839, 2008.

[31] S. Mallat, *A wavelet tour of signal processing: the sparse way*. Academic Press, 2008.

[32] G. Yu and J.-J. Slotine, "Audio classification from time-frequency texture," *arXiv Prepr arXiv:0809.4501*, 2008.

[33] R. C. Gonzalez, "Digital image processing," 2016.

[34] M. Esmaeilpour, A. Mansouri, and A. Mahmoudi-Aznaveh, "A new svd-based image quality assessment," in *8th Iranian Conf Mach Vis Image Proc*, 2013, pp. 370–374.

[35] M. E. Wall, A. Rechtsteiner, and L. M. Rocha, "Singular value decomposition and principal component analysis," in *A practical approach to microarray data analysis*. Springer, 2003, pp. 91–109.

[36] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT Press Cambr, 2016, vol. 1.

[37] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *25th Intl Conf Mach Learn*, 2008, pp. 1096–1103.

[38] D. G. Lowe, "Object recognition from local scale-invariant features," in *7th IEEE Intl Conf Comp Vis*, vol. 2, 1999, pp. 1150–1157.

[39] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Europ Conf Comp Vis*, 2006, pp. 404–417.

[40] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *18th ACM-SIAM Symp Discrete Alg*, 2007, pp. 1027–1035.

[41] A. Coates and A. Y. Ng, "Learning feature representations with k-means," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 561–580.

[42] B. McFee, E. J. Humphrey, and J. P. Bello, "A software framework for musical data augmentation." in *ISMIR*, 2015, pp. 248–254.

[43] M. Cowling and R. Sitte, "Comparison of techniques for environmental sound recognition," *Patt Recog Lett*, vol. 24, no. 15, pp. 2895–2907, 2003.

[44] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *J Mach Learn Research*, vol. 12, pp. 2825–2830, 2011.

[45] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

[46] A. Nayebi and S. Ganguli, "Biologically inspired protection of deep networks from adversarial attacks," *arXiv Prepr arXiv:1703.09202*, 2017.

[47] D. Meng and H. Chen, "Magnet: a two-pronged defense against adversarial examples," in *ACM SIGSAC Conf Comp and Commun Secur*, 2017, pp. 135–147.

PLACE
PHOTO
HERE

Mohammad Esmaeilpour received his B.Sc. degree in computer software engineering in 2012, M.Sc. of artificial intelligence in 2014 (recognized as top straight A+ student), and currently doing his PhD at École de Technologie Supérieure (ÉTS), Montréal, Quebec. As a physically-based animation programmer, he has been involved in several character locomotion modeling projects at Ubisoft gaming company (Montréal studio.) In an industrial collaboration with AudioZ company in Montréal, he is implementing a voice activity detection system for speech processing. His research interests include developing machine learning algorithms for high dimensional data such as animation, wild audio, and speech.

PLACE
PHOTO
HERE

Patrick Cardinal received the B. Eng. degree in electrical engineering in 2000 from École de Technologie Supérieure (ÉTS), M.Sc. from McGill University in 2003 and PhD from ÉTS in 2013. From 2000 to 2013, he has been involved in several projects related to speech processing, especially in the development of a closed-captioning system for live television shows based on automatic speech recognition. After his postdoc at MIT, he joined ÉTS as a professor. His research interests cover several aspects of speech processing for real life and medical applications.

PLACE
PHOTO
HERE

Alessandro Lameiras Koerich is an Associate Professor in the Dept. of Software and IT Engineering of the École de Technologie Supérieure (ÉTS), University of Québec, Montréal, Canada. He received the B.Sc. degree in electrical engineering from the Federal University of Santa Catarina, Brazil, in 1995, the MSc degree in electrical engineering from the University of Campinas, Brazil, in 1997, and the Ph.D. degree in engineering from the ÉTS, in 2002. From 1997 to 1998, he was a lecturer at the Federal Technological University of Paraná. From 1998 to 2002, he was a visiting scientist at the CENPARMI, Montréal, Canada. From 2003 to 2015 he was with the Pontifical Catholic University of Parana, Curitiba, Brazil, where he became professor in 2010 and served as chair of Graduate Studies in CS from 2006 to 2008. From 2009 to 2015 he was also an associate professor in the Dept. of Electrical Engineering of Federal University of Paraná. In 2004, he was nominated IEEE CS Latin America Distinguished Speaker. He was a visiting researcher at INESC-Porto, Portugal from 2009 to 2012 and served as a Fulbright Visiting Professor in the Dept. of Electrical Engineering at Columbia University, New York, USA, in 2013. Prof. Koerich is the author of more than 100 papers and holds four patents. He is an associate editor of the Pattern Recognition journal and served as the general chair of the 14th Intl Society for Music Information Retrieval Conference, which was held in Curitiba, Brazil in 2013. His current research interests include computer vision, machine learning and music information retrieval.