Non-Linear Prediction with LSTM Recurrent Neural Networks for Acoustic Novelty Detection

Erik Marchi[1], Fabio Vesperini[2], Felix Weninger[1], Florian Eyben[1] Stefano Squartini[2], and Björn Schuller[3,4]
[1]Machine Intelligence & Signal Processing Group, TU München, Germany
[2]A3LAB, Department of Information Engineering, Università Politecnica delle Marche, Italy
[3]Chair of Complex and Intelligent Systems, University of Passau, Germany
[4]Department of Computing, Imperial College London, UK
Email: erik.marchi@tum.de

***Abstract*—Acoustic novelty detection aims at identifying abnormal/novel acoustic signals which differ from the reference/normal data that the system was trained with. In this paper we present a novel approach based on non-linear predictive denoising autoencoders. In our approach, auditory spectral features of the next short-term frame are predicted from the previous frames by means of Long-Short Term Memory (LSTM) recurrent denoising autoencoders. We show that this yields an effective generative model for audio. The reconstruction error between the input and the output of the autoencoder is used as activation signal to detect novel events. The autoencoder is trained on a public database which contains recordings of typical in-home situations such as talking, watching television, playing and eating. The evaluation was performed on more than 260 different abnormal events. We compare results with state-of-the-art methods and we conclude that our novel approach significantly outperforms existing methods by achieving up to 94.4 % F-Measure.**

I. Introduction

Novelty detection is a challenging task, and it aims at recognising situations in which unusual events occur. The problem can be treated as one-class classification task: typically the amount of normal data consists of a very large set, and the normal class can be accurately modelled, whereas the acoustic events belonging to the class are considered *novel* events. Many approaches have been proposed due to the practical importance of the novelty detection, especially for automatic monitoring systems.

In the past years, many systems have been deployed for surveillance applications. Surveillance can be seen as control of public safety or as the supervision of private environments where people may live alone. In fact, the increasing requirement of public security over the past decades has motivated the installation of sensors such as cameras or microphones in public places (stores, subway, airports, etc.). Thus, the need of unsupervised situation assessment stimulated the signal processing community towards experimenting with several automated frameworks.

Usually, the research in the area of automatic surveillance systems is mainly focused on detecting abnormal events based on the acquired video information. Anyway, the information given by the acoustic signal offers several advantages, such as low computational needs or the fact that the illumination conditions of the space to be monitored do not have an immediate effect on sound; the same applies for possible occlusion or fast events like shots or explosions. The statistical approach is the most widely used for this problem. Its principle is to model data based on its statistical properties and using this information to estimate whether a test sample comes from the same distribution or not.

A. Related work

Statistical and probabilistic approaches are the most commonly used in the field of novelty detection. Novelty detection ranges from automatic recognition of handwriting, the recognition of cancer [1], informatic intrusion detection systems, non-destructive inspection for the analysis of mechanical components [2], audio segmentation [3], to many others.

As early as in 1994, a pioneering study investigated the relationship between the degree of novelty of the input data and the corresponding reliability of the outputs from the neural network, and demonstrated its performance using an application on the control of the oil flow in multiphases pipelines [4].

Subsequent works proposed the application of a compression autoencoder neural network to detect abnormal CPU data usage [5], [6]. In further works [7], [8], [9], [10], the use of a compression autoencoder for outlier detection was studied and in [11] the autoencoder was applied for the task of damage classification under changing environmental conditions. A technique based on a neural network with the task of classifying mixed acoustic events is presented in [12]. The system uses a feed-forward network and splits the input signals into classes or novelty events; it has been tested in a real underwater environment and realises the detection of recurrent events. Overall, several studies exists in the field of acoustic event classification applying Gaussian Mixture Models (GMMs) and Hidden Markov Models (HMMs) to detect human presence (speech, laughter, cough), animal sounds, sounds of objects [13], [14] and sounds caused by various types of guns [15]. However, it is to be emphasised that there is a fundamental difference between event classification and novelty detection: the latter has the greater difficulty of not having *a priori* knowledge of any element of the novelty class. It can be argued that this makes generative models, such as GMMs and HMMs, particularly suited for this task. For example, studies investigated HMM- and GMM-based approaches for acoustic surveillance of abnormal situations [16], [17] and for automatic space monitoring [18]. A (pseudo-)generative model for acoustic novelty detection in the form of a denoising autoencoder has been introduced in a recent study [19].

B. Contribution

A novel weakly supervised method based on a non-linear prediction denoising autoencoder with LSTM recurrent neural networks is introduced for novelty acoustic detection. The use of LSTM as generative model [20] for text [21], handwriting [21], and music [22] generation was recently investigated, however – to our best knowledge – the use of LSTM as a model for audio generation is novel. In our approach the LSTM are trained to predict next frames from previous ones. The auditory spectral features are processed by the autoencoder, which acts as a one-class classifier. Our approach relies on the reconstruction error which the denoising autoencoder commits trying to predict and reconstruct a novel sound which the network has never seen in the training phase. We compare results with state-of-the-art methods and we conclude that our improved approach significantly outperforms existing methods by achieving up to 94.4 % F-Measure.

This contribution is structured as follows: First, a basic description of the different autoencoder-based schemes for acoustic novelty detection is given (Section II), together with the presentation of the non-linear prediction approach. Then the LSTM recurrent neural networks, thresholding strategy, and features employed in experiments are described in Section III. The used database and the experimental set-up are jointly discussed with the evaluation of obtained results in Section IV. Section V finally draws the paper conclusions.

II. Autoencoders for Acoustic Novelty Detection

This section introduces the basic concepts of autoencoders and describes the basic autoencoder, compression autoencoder, denoising autoencoder, and non-linear predictive autoencoder.

A. Basic Autoencoder

A basic autoencoder (AE) – a kind of neural network typically consisting of only one hidden layer –, sets the target values to be equal to the input. Deep neural networks use it, as an element, to find common data representation from the input [23], [24]. Formally, in response to an input example $x \in \mathbf{R}^n$, the hidden representation $h(x) \in \mathbf{R}^m$ is

$$h(x) = f(W_1 x + b_1), \tag{1}$$

where $f(z)$ is a non-linear activation function, typically a logistic sigmoid function $f(z) = 1/(1 + \exp(-z))$ applied component-wise, $W_1 \in \mathbf{R}^{m \times n}$ is a weight matrix, and $b_1 \in \mathbf{R}^m$ is a bias vector.

The network output maps the hidden representation h back to a reconstruction $\tilde{x} \in \mathbf{R}^n$:

$$\tilde{x} = f(W_2 h(x) + b_2), \tag{2}$$

where $W_2 \in \mathbf{R}^{n \times m}$ is a weight matrix, and $b_2 \in \mathbf{R}^n$ is a bias vector.

Given an input set of examples $\mathcal{X}$, autoencoder training consists in finding parameters $\theta = \{W_1, W_2, b_1, b_2\}$ that minimise the reconstruction error, which corresponds to minimising the following objective function:

$$\mathcal{J}(\theta) = \sum_{x \in \mathcal{X}} \|x - \tilde{x}\|^2 . \tag{3}$$

The minimization is usually realised by stochastic gradient descent as in the training of neural networks. The structure of the AE is given in Figure 1a.

B. Compression Autoencoder

In the case of having the number of hidden units m smaller than the number of input units n, the network is forced to learn a compressed representation of the input. For example, if some of the input features are correlated, then this Compression Autoencoder (CAE) is able to learn those correlations and reconstruct the input data from a compressed representation. The structure of the AE is given in Figure 1b.

C. Denoising Autoencoder

The denoising autoencoder [25] forces the hidden layer to retrieve more robust features and prevent it from simply learning the identity. In such a configuration the autoencoder is trained to reconstruct the input from a corrupted version of it.

Formally, the initial input x is corrupted by means of additive isotropic Gaussian noise in order to obtain: $x'|x \sim N(x, \sigma^2 I)$. The corrupted input x' is then mapped, as with the basic autoencoder, to a hidden representation

$$h(x') = f(W_1' x' + b_1'), \tag{4}$$

from which we reconstruct a the original signal as follows

$$\tilde{x}' = f(W_2' x + b_2'). \tag{5}$$

The parameters $\theta' = \{W_1', W_2', b_1', b_2'\}$ are trained to minimise the average reconstruction error over the training set, to have $\tilde{x}'$ as close as possible to the uncorrupted input x, which corresponds to minimising the objective function in Equation 3. The structure of the denoising autoencoder is shown in Figure 1c.

D. Non-Linear Predictive Autoencoder

The idea of a non-linear predictive autoencoder is quite intuitive. The autoencoder is trained to predict the next frame from the previous one. Formally, the input up to a given time frame x_t is mapped to a hidden representation h

$$h(x_t) = f(W_1^*, b_1^*, x_{1,\ldots,t}), \tag{6}$$

where W and b denote weights and bias, respectively. From this we reconstruct an approximation of the original signal as follows,

$$\tilde{x}_{t+k} = f(W_2^*, b_2^*, h_{1,\ldots,t}), \tag{7}$$

where k is the prediction delay, and $h_i = h(x_i)$. The parameters $\theta^* = \{W_1^*, W_2^*, b_1^*, b_2^*\}$ are trained to minimise the average reconstruction error over the training set, to have $\tilde{x}_{t+k}$ as close as possible to the prediction delay. A prediction delay of $k = 1$ corresponds to a shift of 10 ms in the audio signal (cf.

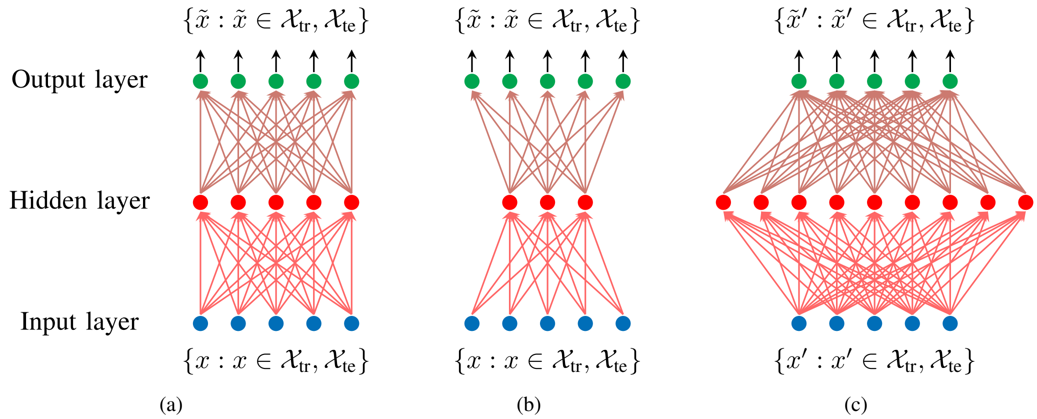
$\{\tilde{x} : \tilde{x} \in \mathcal{X}_{\text{tr}}, \mathcal{X}_{\text{te}}\}$
$\{\tilde{x} : \tilde{x} \in \mathcal{X}_{\text{tr}}, \mathcal{X}_{\text{te}}\}$
$\{\tilde{x}' : \tilde{x}' \in \mathcal{X}_{\text{tr}}, \mathcal{X}_{\text{te}}\}$
Output layer
Hidden layer
Input layer
$\{x : x \in \mathcal{X}_{\text{tr}}, \mathcal{X}_{\text{te}}\}$
$\{x : x \in \mathcal{X}_{\text{tr}}, \mathcal{X}_{\text{te}}\}$
$\{x' : x' \in \mathcal{X}_{\text{tr}}, \mathcal{X}_{\text{te}}\}$
(a)
(b)
(c)

Fig. 1. Structure of the (*a*) basic autoencoder (AE), (*b*) compression autoencoder (CAE), and (*c*) denoising autoencoder (DAE) on the training set $\mathcal{X}_{\text{tr}}$ or testing set $\mathcal{X}_{\text{te}}$. $\mathcal{X}_{\text{tr}}$ contains data of non-novel acoustic events; $\mathcal{X}_{\text{te}}$ consists of *novel* and *non-novel* acoustic events.

Section III-C). The training of the parameters is performed by minimising the objective function (3) – the difference is that $\tilde{x}$ is now based on non-linear prediction according to (6) and (7). The training set $\mathcal{X}_{\text{tr}}$ consists of background environmental sounds, and the test set $\mathcal{X}_{\text{te}}$ consists of recordings containing abnormal sounds. In our approach, the initial input x_t is corrupted by means of additive isotropic Gaussian noise in order to obtain: $x'|x \sim N(x, \sigma^2 I)$. The resulting structure of the non-linear predictive denoising autoencoder (NP-DAE) is shown in Figure 2. An overall block diagram of the proposed novelty detector is depicted in Figure 3. In our study, the equations (6) and (7) are implemented as LSTM-RNNs (cf. below).

III. LSTM Recurrent Neural Networks, Thresholding, and Features

This section introduces the LSTM recurrent neural networks, describes the thresholding strategy, and the features employed in our experiments.

A. LSTM

LSTM networks were introduced in [26]. Compared to a conventional RNN, the hidden units are replaced by so-called memory blocks. These memory blocks can store information in the cell variable $\boldsymbol{c}_t$. In this way, the network can exploit long-range temporal context. Each memory block consists of a memory cell and three gates: the input gate, output gate, and forget gate, as depicted in Fig. 4.

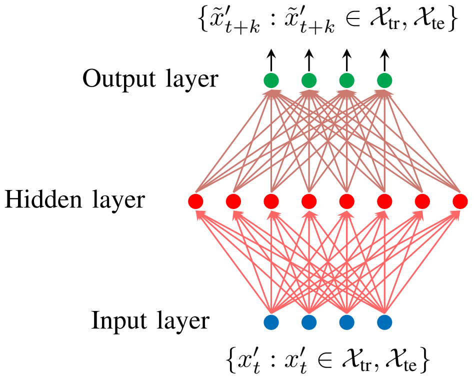
$\{\tilde{x}'_{t+k} : \tilde{x}'_{t+k} \in \mathcal{X}_{\text{tr}}, \mathcal{X}_{\text{te}}\}$
Output layer
Hidden layer
Input layer
$\{x'_t : x'_t \in \mathcal{X}_{\text{tr}}, \mathcal{X}_{\text{te}}\}$

Fig. 2. Structure of the non-linear predicting denoising autoencoder (NP-DAE) on the training set $\mathcal{X}_{\text{tr}}$ or testing set $\mathcal{X}_{\text{te}}$. $\mathcal{X}_{\text{tr}}$ contains data of non-novel acoustic events; $\mathcal{X}_{\text{te}}$ consists of *novel* and *non-novel* acoustic events.

These gates control the behaviour of the memory block. The forget gate can reset the cell variable which leads to ‘forgetting’ the stored input $\boldsymbol{c}_t$, while the input and output gates are responsible for reading input from $\boldsymbol{x}_t$ and writing output to $\boldsymbol{h}_t$, respectively:

$$\boldsymbol{c}_t = \boldsymbol{f}_t \otimes \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \otimes \tanh(\boldsymbol{W}_{xc}\boldsymbol{x}_t + \boldsymbol{W}_{hc}\boldsymbol{h}_{t-1} + \boldsymbol{b}_c) \quad (8)$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \otimes \tanh(\boldsymbol{c}_t) \quad (9)$$

where $\otimes$ denotes element-wise multiplication and $\tanh$ is also applied in an element-wise fashion. The variables $\boldsymbol{i}_t$, $\boldsymbol{o}_t$ and $\boldsymbol{f}_t$ are the output of the input gates, output gates and forget gates, respectively, $\boldsymbol{b}_c$ is a bias term, and $\boldsymbol{W}$ is the weight matrix. Each memory block can be regarded as a separate, independent unit. Therefore, the activation vectors $\boldsymbol{i}_t$, $\boldsymbol{o}_t$, $\boldsymbol{f}_t$, and $\boldsymbol{c}_t$ are all of same size as $\boldsymbol{h}_t$, i. e., the number of memory blocks in the hidden layer. Furthermore, the weight matrices from the cells to the gates are diagonal, which means that each gate is only dependent on the cell within the same memory block.

In addition to LSTM memory blocks, we use bidirectional RNNs [27]. A bidirectional RNN can access context from both temporal directions. This is achieved by processing the input data in both directions with two separate hidden layers. Both hidden layers are then fed to the output layer. The combination of bidirectional RNNs and LSTM memory blocks leads to

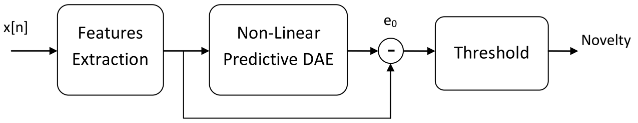

Fig. 3. Block diagram of the proposed acoustic novelty detector with a denoising autoencoder. Features are extracted from the input signal and the reconstruction error between the input and the reconstructed features is then processed by a thresholding block which detects the *novel* or *non-novel* event.

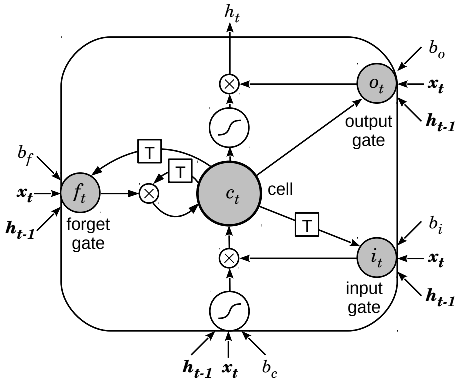

Fig. 4. Long Short-Term Memory block, containing a memory cell and the input, output, and forget gates

bidirectional LSTM networks [28], where context from both temporal directions is exploited. It has to be noted that, using bidirectional LSTM networks makes it impossible to use the system for online processing.

A network composed of more than one hidden layer is referred to as a deep neural network (DNN) [29]. By stacking multiple (potentially pre-trained, but not in our system) hidden layers on top of each other, increasingly higher level representations of the input data are created (deep learning). When multiple hidden layers are employed, the output of the network is (in the case of a bidirectional RNN) computed as

$$\boldsymbol{y}_t = \boldsymbol{W}_{\overrightarrow{\mathbf{h}}^N y} \overrightarrow{\boldsymbol{h}}_t^N + \boldsymbol{W}_{\overleftarrow{\mathbf{h}}^N y} \overleftarrow{\boldsymbol{h}}_t^N + \boldsymbol{b}_y, \qquad (10)$$

where $\overrightarrow{\boldsymbol{h}}_t^N$ and $\overleftarrow{\boldsymbol{h}}_t^N$ are the forward and backward activations of the N-th (last) hidden layer, respectively.

We conducted several preliminary evaluations to find the best network layout by varying the number of hidden layers and their size (i. e., the number of LSTM units for each layer). The best network layout for RNN has three hidden layers with 216 LSTM units each. The best network layout for bidirectional LSTM (BLSTM) has six hidden layers (three for each direction) with 156, 256, and 156 LSTM units respectively.

Supervised learning was applied up to 100 epochs for training the network. Network weights are recursively updated by standard gradient descent with backpropagation of the sum squared error. The gradient descent algorithm requires the network weights to be initialised with non zero values; thus, we initialise the weights with a random Gaussian distribution with mean 0 and standard deviation 0.1.

B. Thresholding

The input and output layer of the network have 54 units. Thus, the trained autencoder is able to reconstruct each sample and novel events are identified by processing the reconstruction error with an adaptive threshold. The input x is segmented into sequences of 30 seconds of length. For every time-step the Euclidean distance between each standardised input feature value and the networks output is computed. The distances are summed up and divided by the number of coefficients in order to represent the reconstruction error of each time-step with a single value. A threshold θ_{th} is then applied to obtain a binary signal. The threshold is shifted from the median of the error signal of a sequence e_0 by a multiplicative coefficient β, which ranges from $\beta_{min} = 1$ to $\beta_{max} = 2$:

$$\theta_{th} = \beta * \text{median}(e_0(1), ..., e_0(N)) \qquad (11)$$

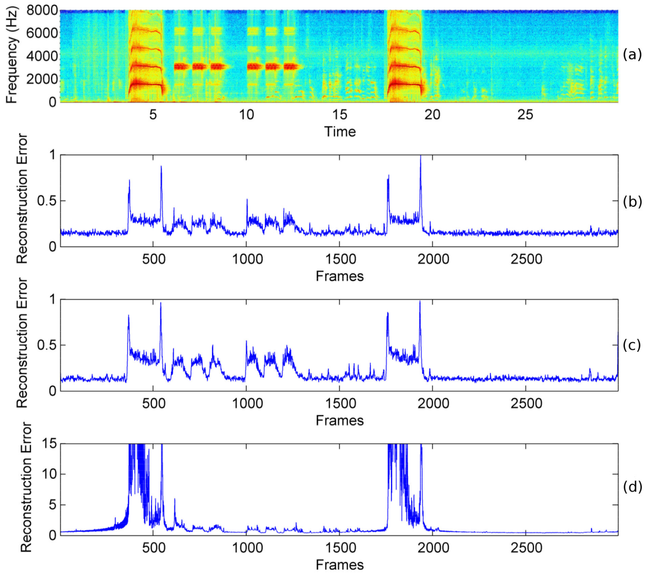

Fig. 5. *(a)*: Spectrogram of a 30 seconds sequence containing three novel events, such as a siren and two screams. *(b)*: Reconstruction error signal of the related sequence obtained with a BLSTM-DAE. *(c)*: Reconstruction error signal of the related sequence obtanined with a non-linear predictive BLSTM-DAE (NP-BLSTM-DAE) with a prediction delay of 3 frames. *(d)*: Reconstruction error signal of the related sequence obtanined with a NP-BLSTM-DAE with a delay of 5 frames.

Fig. 5 shows the reconstruction error for a given sequence. The figure clearly depicts a low reconstruction error in reproducing normal input such as talking, television sounds, and other normal environmental sounds. On the other hand, the denoising autoencoder shows a high reconstruction error when it comes to reproduce novel acoustic events such as a scream, or an alarm. Additionally it shows how the different prediction delays modify the activation.

TABLE I. ACOUSTIC NOVEL EVENTS IN THE TEST SET. SHOWN ARE THE NUMBER OF DIFFERENT EVENTS, THE AVERAGE DURATION, AND THE TOTAL DURATION IN SECONDS PER EVENT TYPE.

Type	# Events	Avg. Duration (s)	Total Duration (s)
Alarm	76	6.0	435.8
Scream	111	1.9	214.6
Falls	48	1.8	89.5
Fracture	32	2.2	70.4
Total	267	2.4	810.3

C. Features

Auditory Spectral Features (ASF) [30] are computed by applying Short Time Fourier Transform (STFT) using a frame size of 30 ms and a frame step of 10 ms. Each STFT yields the power spectrogram which is converted to the Mel-Frequency scale using a filter-bank with 26 triangular filters obtaining the Mel spectrograms $M_{30}(n,m)$, with n being the frame index, and m the frequency bin index. Finally, to match the human perception of loudness, a logarithmic representation is chosen:

$$M^{30}_{log}(n,m) = log(M_{30}(n,m) + 1.0) \qquad (12)$$

In addition the positive first order differences $D_{30}(n,m)$ are calculated from each Mel spectrogram following:

$$D_{30}(n,m) = M^{30}_{log}(n,m) - M^{30}_{log}(n-1,m) \qquad (13)$$

Furthermore, the frame energy and its derivative are also included as feature ending up in a total number of 54 features. The features are extracted with our open-source audio analysis toolkit openSMILE [31].

IV. Experiments and Results

This section contains the data set used for our evaluation (Section IV-A), the experiments' setup (Section IV-B), and a description of the performances obtained with the proposed approach (Section IV-C).

A. Evaluation Data Set

Our evaluation dataset is composed by around three hours of recordings of a home environment, taken from the PASCAL CHiME speech separation and recognition challenge dataset [32]. It consists of a typical in-home scenario (a living room), recorded during different days and times, while the inhabitants (two adults and two children) perform common actions, such as talking, watching television, playing, eating. We used randomly chosen sequences to compose 100 minutes of background for training set, and around 70 minutes for testing set. The original dataset was recorded in 2 channels (with a binaural microphone) and a sample-rate of 16 kHz. The test set[1] was generated adding different kinds of sounds[2], such as screams, alarms, falls and fractures (cf. Table I). The test set did not include any overlapping events, the events were normalised to the volume of the background recordings, and they were added at random position thus the distance between one event and another is not fixed.

B. Experimental Setup

Several experiments were conducted, to find the the most suitable setup. The networks were trained with gradient steepest descent algorithm on the sum of squared errors (SSE) with a fixed momentum of 0.9, at different constant values of learning rate $l = \{1e^{-4}, 1e^{-5}, ..., 1e^{-8}\}$, and different noise sigma values $\sigma = \{0.01, 0.1, 0.25\}$. In the case of the basic (AE) and the compression autoencoder (CAE) with BLSTM and LSTM no additive Gaussian noise was applied. The prediction delay was applied at different values: $k = \{1, 2, 3, 4, 5, ..., 10\}$. One prediction delay unit corresponds to 10 ms. The autoencoders were trained using our open-source CUDA RecurREnt Neural Network Toolkit (CURRENNT) [33]. As evaluation metrics we used Precision, Recall, and F-measure. We evaluated several topologies for the non-linear predictive denoising autoencoder ranging from 54-128-54 to 270-370-270, and from 54-20-54 to 54-54-54 in the case of compression/basic autoencoder. Every network topology was evaluated for each 100 epochs of training. In order to compare our results with the state of the art methods, we reported the performances obtained with normal basic, compression, and denoising autoencoders [19]. We employed further two typical approaches based on GMM and HMM. In the case of GMMs, models were trained at different numbers of Gaussian components 2^n with $n = \{1, 2, ..., 8\}$, whereas left-right HMMs were trained with different numbers of states $s = \{3, 4, 5\}$ and 2^n Gaussian components with $n = \{1, 2, ..., 7\}$. GMMs and HMMs were trained using the *Torch* [34] toolkit. The log-likelihood signal produced as output of the probabilistic models was post-processed with a similar thresholding algorithm (cf. Section III) in order to fairly compare the performances among the different methods. For all the experiments and settings we maintained the same feature set.

TABLE II. Best setups for non-linear prediction (NP) autoencoders in the different layouts: compression autoencoder (CAE) with (B)LSTM, basic autoencoder (AE) with (B)LSTM, and denoising autoencoder (DAE) with (B)LSTM.

Method	**Layout**	**Delay** (k)	**P** (%)	**R** (%)	$\mathbf{F_1}$ (%)
NP-LSTM-CAE	54-30-54	1	93.7	91.3	92.5
NP-BLSTM-CAE	54-30-54	3	94.6	91.1	92.8
NP-LSTM-AE	54-54-54	1	92.5	91.6	92.1
NP-BLSTM-AE	54-54-54	2	95.7	92.5	94.1
NP-LSTM-DAE	**216-216-216**	**1**	**95.2**	**93.2**	**94.2**
NP-BLSTM-DAE	**156-256-156**	**3**	**94.9**	**93.9**	**94.4**

C. Results

Figure 6 reports performances for progressive values of the prediction delay (k) – from 0 up to 10 – using a Compression Autoencoder (CAE), Basic Autoencoder (AE), and Denoising Autoencoder (DAE) with both, LSTM and BLSTM neural networks. We evaluated several layouts (cf. Section IV-B) per network type, but we show only the best ones. Setting a prediction delay of 3 frames, which corresponds to a total prediction delay of 30 ms, leads to the best performances of up to 94.4 % F-Measure in the NP-BLSTM-DAE network, whereas for NP-LSTM-DAE we observe better performances with a delay of one frame (10 ms) of up to 94.2 % F-Measure (cf. Table II). For all the approaches best values of F-Measure is obtained with $k = 2$ or $k = 3$.

It must be observed that, increasing the prediction delay led to a significant decrease of an F-Measure down to 86 2%. This is due to the fact that, in presence of higher prediction delays, quick periodic events induce a higher reconstruction error, as shown in Fig. 5, where the activation in subplot c clearly presents higher errors with respect to the one depicted in in subplot d.

Table II reports the best results for each autoencoder type and networks. In general, the superiority of DAE schemes with respect to the CAE/AE ones is motivated by the the strength of a denoising autoencoder of encoding the input by preserving the information about the input itself and simultaneously reversing the effect of a corruption process applied to the input of the auto-encoder: The combination of these two learning

[1] available for download here: http://a3lab.dii.univpm.it/webdav/audio/testset_PASCAL_CHiME_plus_novel.tar.gz

[2] taken from www.freesound.org

processes proved to be effective in the experimental results. Here, we combined the ability of the denoising autencoder with the non-linear prediction, allowing us to achieve the best performance (up to 94.4 % F-Measure).

As an overall evaluation on the test set, Fig. 7 shows the comparison between state-of-the-art methods and our proposed approach in terms of F-Measure, Precision, and Recall. We observe that, the proposed NP-BLSTM-DAE method provided the best performance in terms of Precision, Recall, and F-Measure of up to 94.9 %, 93.9 %, and 94.4 % respectively (cf. Table III). A significant absolute improvement (one-tailed z-test [35], p<0.01) of 3.0 % F-Measure is observed against the HMM-based approach, while an absolute improvement of 4.0 % F-measure is exhibited with respect to the GMM-based method; an absolute improvement of 1 % is observed over the 'ordinary' BLSTM-DAE. It must be also noted that, with non-linear prediction, the compression autoencoders NP-(B)LSTM-CAE (92.1 %) also improved relevantly when compared to the (B)LSTM-CAE (89.1 %). Thus, while in a previous paper [19] applying only a single compression learning process seemed to be not sufficient to encode effectively information about the input, here CAE works better when the non-linear prediction scheme is applied.

The beneficial impact of the proposed approach is also evident in the case of 'normal' autoeconders (i. e., with no compression or denoising): a value of 94.1 % F-Measure for the NP-BLSTM-AE is achieved in this case. Moreover, if we look at the combination of a nonlinear prediction and a denoising autoencoder, which gave us the best performance, we can also notice that LSTM behaviour is in this case comparable with the BLSTM one and the corresponding F-Measure value is superior to those obtained by using the state-of-art methods. Note that, the LSTM network is causal by nature, and thus, can be implemented in real-time.

Concluding, the obtained results showed that the employment of the nonlinear prediction paradigm in combination with the different (B)LSTM autoencoder-based learning schemes is effective and a significant performance improvement with respect to the state-of-the-art approaches was registered.

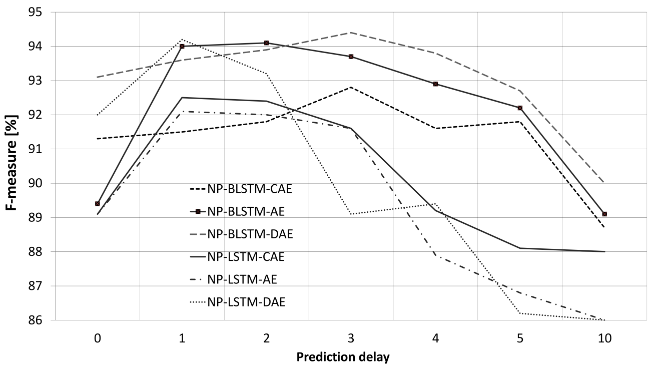

Fig. 6. Comparison between the different types of NP-Autoencoders varying the prediction delay.

TABLE III. Comparison over existing methods by percentage of Precision, Recall, and F-Measure. Indicated layout and prediction delay. Reported approaches are: GMM, HMM, compression autoencoder with BLSTM (BLSTM-CAE) or LSTM (LSTM-CAE), denoising autoencoder with BLSTM (BLSTM-DAE) or LSTM(LSTM-DAE), and related versions of non-linear predictive autoencoders NP-(B)LSTM-CAE/AE/DAE.

Method	**Layout**	**Delay** (k)	**P** (%)	**R** (%)	$\mathbf{F_1}$ (%)
GMM	$g = 128$	-	91.1	87.8	89.4
HMM	$s = 5, g = 64$	-	94.1	88.9	91.4
LSTM-CAE	54-30-54	0	91.7	86.6	89.1
BLSTM-CAE	54-30-54	0	93.6	89.2	91.3
LSTM-AE	54-54-54	0	91.0	87.4	89.1
BLSTM-AE	54-54-54	0	91.1	87.8	89.4
LSTM-DAE	156-256-156	0	94.2	90.6	92.4
BLSTM-DAE	216-216-216	0	94.7	92.0	93.4
NP-LSTM-CAE	54-30-54	1	93.7	91.3	92.5
NP-BLSTM-CAE	54-30-54	3	94.6	91.1	92.8
NP-LSTM-AE	54-54-54	1	92.5	91.6	92.1
NP-BLSTM-AE	54-54-54	2	95.7	92.5	94.1
NP-LSTM-DAE	**216-216-216**	**1**	**95.2**	**93.2**	**94.2**
NP-BLSTM-DAE	**156-256-156**	**3**	**94.9**	**93.9**	**94.4**

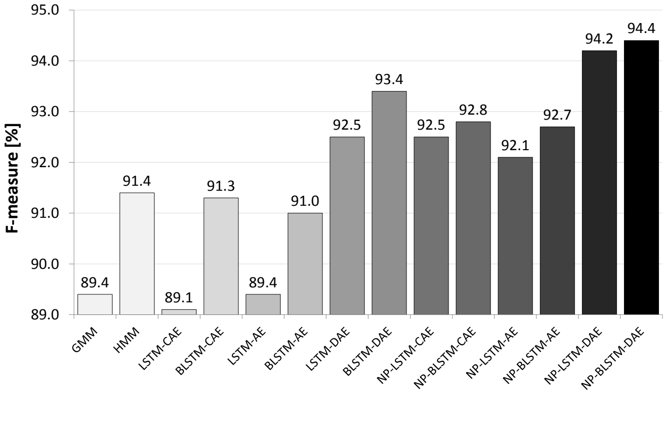

Fig. 7. Comparison with existing percentage of F-Measure.

V. Conclusions and Outlook

We presented a novel, purely unsupervised approach to acoustic novelty detection. It relies on auditory spectral features and non-linear prediction autoencoders with Long Short-Term Memory acting as a one-class classifier. Our approach exploits the reconstruction error of the autoencoder when trying to predict and denoise a novel sound which the network has never seen in the training phase. The strength of a NP-DAE is due the combination of two learning processes: encoding the input by preserving the information about on its variations on the following frames, simultaneously and removing the corruption process applied to the input. Additionally using the LSTM and BLSTM architectures enables the system to use and learn more context. Results are compared with state-of-the-art methods and we conclude that our novel approach significantly outperforms existing methods by achieving up to 94.4 % F-Measure and with an absolute improvement of 3 % over HMM system and of 1 % over ordinary BLSTM-DAE. Future works will focus on the effectiveness of the approach with real-life databases. Moreover further improvements could be obtained to use different type of features, likely more suitable to deal with non-stationary events, like already done by some of the authors in the musical onset case study [36].

Acknowledgment

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 338164 (ERC Starting Grant iHEARu), and No. 645378 (RIA ARIA VALUSPA).

References

[1] L. Tarassenko, P. Hayton, N. Cerneaz, and M. Brady, "Novelty detection for the identification of masses in mammograms," in *Proceedings Fourth International Conference on Artificial Neural Networks, 1995.* IET, 1995, pp. 442–447.

[2] K. Worden, G. Manson, and D. Allman, "Experimental validation of a structural health monitoring methodology: Part i. novelty detection on a laboratory structure," *Journal of Sound and Vibration*, vol. 259, no. 2, pp. 323–343, 2003.

[3] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *Proceedings IEEE International Conference on Multimedia and Expo, ICME 2000.*, vol. 1. IEEE, 2000, pp. 452–455.

[4] C. M. Bishop, "Novelty detection and neural network validation," *IEEE Vision, Image and Signal Processing*, vol. 141, no. 4, pp. 217–222, Aug 1994.

[5] N. Japkowicz, C. Myers, M. Gluck *et al.*, "A novelty detection approach to classification," in *Proceedings International Joint Conference on Artificial Intelligence, IJCAI 1995*, 1995, pp. 518–523.

[6] B. B. Thompson, R. J. Marks, J. J. Choi, M. A. El-Sharkawi, M.-Y. Huang, and C. Bunje, "Implicit learning in autoencoder novelty assessment," in *Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on*, vol. 3. IEEE, 2002, pp. 2878–2883.

[7] S. Hawkins, H. He, G. Williams, and R. Baxter, "Outlier detection using replicator neural networks," in *Data Warehousing and Knowledge Discovery*. Springer, 2002, pp. 170–180.

[8] N. Japkowicz, "Supervised versus unsupervised binary-learning by feedforward neural networks," *Machine Learning*, vol. 42, no. 1-2, pp. 97–122, 2001.

[9] L. Manevitz and M. Yousef, "One-class document classification via neural networks," *Neurocomputing*, vol. 70, no. 79, pp. 1466 – 1481, 2007.

[10] G. Williams, R. Baxter, H. He, S. Hawkins, and L. Gu, "A comparative study of rnn for outlier detection in data mining," in *Proceedings IEEE 13th International Conference on Data Mining, 2002*. IEEE Computer Society, 2002, pp. 709–709.

[11] H. Sohn, K. Worden, and C. R. Farrar, "Statistical damage classification under changing environmental and operational conditions," *Journal of Intelligent Material Systems and Structures*, vol. 13, no. 9, pp. 561–574, 2002.

[12] G. Linares, P. Nocera, and H. Meloni, "Mixed acoustic events classification using ica and subspace classifier," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 1997.*, vol. 4, Apr 1997, pp. 3365–3368 vol.4.

[13] P. Atrey, M. Maddage, and M. Kankanhalli, "Audio based event detection for multimedia surveillance," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2006.*, vol. 5, May 2006, pp. V–V.

[14] A. Harma, M. F. McKinney, and J. Skowronek, "Automatic surveillance of the acoustic activity in our living environment," in *IEEE International Conference on Multimedia and Expo, ICME 2005.* IEEE, 2005, p. 4.

[15] C. Clavel, T. Ehrette, and G. Richard, "Events detection for an audio-based surveillance system," in *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, July 2005, pp. 1306–1309.

[16] S. Ntalampiras, I. Potamitis, and N. Fakotakis, "Probabilistic novelty detection for acoustic surveillance under real-world conditions," *IEEE Transactions on Multimedia*, vol. 13, no. 4, pp. 713–719, 2011.

[17] E. Principi, S. Squartini, R. Bonfigli, G. Ferroni, and F. Piazza, "An integrated system for voice command recognition and emergency detection based on audio signals," *Expert Systems with Applications*, 2015.

[18] S. Ntalampiras, I. Potamitis, and N. Fakotakis, "On acoustic surveillance of hazardous situations," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2009.* IEEE, 2009, pp. 165–168.

[19] E. Marchi, F. Vesperini, F. Eyben, S. Squartini, and B. Schuller, "A Novel Approach for Automatic Acoustic Novelty Detection Using a Denoising Autoencoder with Bidirectional LSTM Neural Networks," in *Proceedings 40th IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2015*, IEEE. Brisbane, Australia: IEEE, April 2015, (5 pages), to appear.

[20] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with lstm recurrent networks," *The Journal of Machine Learning Research*, vol. 3, pp. 115–143, 2003.

[21] A. Graves, "Generating sequences with recurrent neural networks," *CoRR*, vol. abs/1308.0850, 2013.

[22] D. Eck and J. Schmidhuber, "Finding temporal structure in music: Blues improvisation with lstm recurrent networks," in *12th IEEE Workshop on Neural Networks for Signal Processing, 2002.* IEEE, 2002, pp. 747–756.

[23] I. Goodfellow, H. Lee, Q. V. Le, A. Saxe, and A. Y. Ng, "Measuring invariances in deep networks," in *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, Eds. Curran Associates, Inc., 2009, pp. 646–654.

[24] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. Platt, and T. Hoffman, Eds. MIT Press, 2007, pp. 153–160.

[25] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, Dec. 2010.

[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[27] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, Nov 1997.

[28] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional {LSTM} and other neural network architectures," *Neural Networks*, vol. 18, no. 56, pp. 602 – 610, 2005, {IJCNN} 2005.

[29] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural ynetworks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[30] F. Eyben, S. Böck, B. Schuller, and A. Graves, "Universal onset detection with bidirectional long short-term memory neural networks." in *ISMIR*, 2010, pp. 589–594.

[31] F. Eyben, M. Wöllmer, and B. Schuller, "openSMILE – The Munich Versatile and Fast Open-Source Audio Feature Extractor," in *Proceedings of the 18th ACM International Conference on Multimedia, MM 2010*, ACM. Florence, Italy: ACM, October 2010, pp. 1459–1462.

[32] J. Barker, E. Vincent, N. Ma, H. Christensen, and P. Green, "The pascal chime speech separation and recognition challenge," *Computer Speech & Language*, vol. 27, no. 3, pp. 621–633, 2013.

[33] F. Weninger, J. Bergmann, and B. Schuller, "Introducing CURRENNT, the Munich open-source CUDA RecurREnt Neural Network Toolkit," *Journal of Machine Learning Research*, 2014, in press.

[34] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," in *BigLearn, NIPS Workshop*, no. EPFL-CONF-192376, 2011.

[35] M. Smucker, J. Allan, and B. Carterette, "A comparison of statistical significance tests for information retrieval evaluation," in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. ACM, 2007, pp. 623–632.

[36] E. Marchi, G. Ferroni, F. Eyben, L. Gabrielli, S. Squartini, and B. Schuller, "Multi-resolution linear prediction based features for audio onset detection with bidirectional lstm neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 2164–2168.