MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Duration-Controlled LSTM for Polyphonic Sound Event Detection

Hayashi, T.; Watanabe, S.; Toda, T.; Hori, T.; Le Roux, J.; Takeda, K.

TR2017-150 August 2017

Abstract

This paper presents a new hybrid approach called duration-controlled long short-term memory (LSTM) for polyphonic Sound Event Detection (SED). It builds upon a state-ofthe-art SED method which performs frame-by-frame detection using a bidirectional LSTM recurrent neural network (BLSTM), and incorporates a duration-controlled modeling technique based on a hidden semi-Markov model (HSMM). The proposed approach makes it possible to model the duration of each sound event precisely and to perform sequence-by-sequence detection without having to resort to thresholding, as in conventional frame-by-frame methods. Furthermore, to effectively reduce sound event insertion errors, which often occur under noisy conditions, we also introduce a binary-mask-based post-processing which relies on a sound activity detection (SAD) network to identify segments with any sound event activity, an approach inspired by the well-known benefits of voice activity detection in speech recognition systems. We conduct an experiment using the DCASE2016 task 2 dataset to compare our proposed method with typical conventional methods, such as non-negative matrix factorization (NMF) and standard BLSTM. Our proposed method outperforms the conventional methods both in an event-based evaluation, achieving a 75.3% F1 score and a 44.2% error rate, and in a segment-based evaluation, achieving an 81.1% F1 score and a 32.9% error rate, outperforming the best results reported in the DCASE2016 task 2 Challenge.

IEEE/ACM Transactions on Audio, Speech, and Language Processing

Duration-Controlled LSTM for Polyphonic Sound Event Detection

Tomoki Hayashi, *Student Member, IEEE,* Shinji Watanabe, *Senior Member, IEEE,* Tomoki Toda, *Member, IEEE,* Takaaki Hori, *Senior Member, IEEE,*, Jonathan Le Roux, *Senior Member, IEEE,* Kazuya Takeda, *Senior Member, IEEE,*

***Abstract*—This paper presents a new hybrid approach called duration-controlled long short-term memory (LSTM) for polyphonic Sound Event Detection (SED). It builds upon a state-of-the-art SED method which performs frame-by-frame detection using a bidirectional LSTM recurrent neural network (BLSTM), and incorporates a duration-controlled modeling technique based on a hidden semi-Markov model (HSMM). The proposed approach makes it possible to model the duration of each sound event precisely and to perform sequence-by-sequence detection without having to resort to thresholding, as in conventional frame-by-frame methods. Furthermore, to effectively reduce sound event insertion errors, which often occur under noisy conditions, we also introduce a binary-mask-based post-processing which relies on a sound activity detection (SAD) network to identify segments with any sound event activity, an approach inspired by the well-known benefits of voice activity detection in speech recognition systems. We conduct an experiment using the DCASE2016 task 2 dataset to compare our proposed method with typical conventional methods, such as non-negative matrix factorization (NMF) and standard BLSTM. Our proposed method outperforms the conventional methods both in an event-based evaluation, achieving a 75.3% F1 score and a 44.2% error rate, and in a segment-based evaluation, achieving an 81.1% F1 score and a 32.9% error rate, outperforming the best results reported in the DCASE2016 task 2 Challenge.**

***Index Terms*—recurrent neural network, long short-term memory, hidden semi-Markov model, duration control, hybrid model, polyphonic sound event detection.**

I. Introduction

THE goal of sound event detection (SED) is to detect the beginning and end of sound events and to identify and label these sounds. SED has great potential for use in many applications, such as retrieval in multimedia databases [1], life-logging [2], activity monitoring [3]–[5], environmental context understanding [6], automatic control of devices in smart homes [7], analysis of noise pollution [8], and so on. Improvements in machine learning techniques have opened new opportunities for progress in these challenging tasks. As a result, SED has been attracting more and more attention, and research in the field is becoming more active. It is notable that several SED challenges have recently been held, such as the CLEAR AED [9], the TRECVID MED [10], and the DCASE Challenge [11], [12].

SED can be divided into two main categories, monophonic and polyphonic, which are used in different scenarios. In monophonic SED, the maximum number of simultaneously active sound events is assumed to be one. Because real-world sound events include a wide range of sounds, which vary in their acoustic characteristics, duration, and volume, such as the sound of glass breaking, typing on keyboards, knocking on doors, the turning of book pages, human speech, and so on, correctly detecting and identifying such sound events is already difficult, even in the monophonic case. On the other hand, in polyphonic SED, there can be any number of simultaneously active sound events. Polyphonic SED is a more realistic task than monophonic SED because it is likely that several sound events will occur simultaneously in real-world situations. But it is also even more difficult, due to the overlapping of multiple sound events.

The most typical SED method is to use a hidden Markov model (HMM), where an emission probability distribution is represented by Gaussian mixture models (GMM-HMM), with mel frequency cepstral coefficients (MFCCs) as features [13], [14]. In the GMM-HMM approach, each sound event, as well as the silence between events, is modeled by an HMM, and the maximum likelihood path is determined using the Viterbi algorithm. However, this approach typically achieves only limited performance, and requires heuristics, such as the number of simultaneously active events, to perform polyphonic SED.

Another approach is to use non-negative matrix factorization (NMF) [15]–[18]. In NMF approaches, a dictionary of basis vectors is learned by decomposing the spectrum of each single-sound event into the product of a basis matrix and an activation matrix, and then combining the basis matrices of all of the sound events. The activation matrix at test time is estimated using the combined basis vector dictionary, and then used either for estimating sound event activations or as a feature vector which is passed on to a classifier. These NMF-based methods can achieve good performance, but they do not take correlation in the time direction into account, and perform frame-by-frame processing. As a result, the prediction results lack temporal stability, and therefore, extensive post-processing is required. It is also necessary to find the optimal number of bases for each sound event.

More recently, methods based on neural networks have been developed, which have also achieved good SED performance [19]–[26]. A single network is typically trained to solve a multi-label classification problem involving polyphonic sound event detection. Some studies [20], [22], [23], [25] have also utilized recurrent neural networks (RNN), which are able to

T. Hayashi, T. Toda, and K. Takeda are with Nagoya University, JAPAN.
S. Watanabe, T. Hori, and J. Le Roux are with Mitsubishi Electric Research Laboratories (MERL), USA.

take into account correlations in the time direction. Although these approaches achieve good performance, they still must perform frame-by-frame detection and do not have an explicit duration model for the output label sequence. Additionally, threshold values for the actual outputs need to be carefully determined in order to achieve the best performance.

In this study, we propose a duration-controlled LSTM system which is a hybrid system of a hidden semi-Markov model and a bidirectional long short-term memory RNN (BLSTM-HSMM), where output duration is explicitly modeled by an HSMM on top of a BLSTM network. The proposed hybrid system was inspired by the BLSTM-HMM hybrid system used in speech recognition systems [27]–[30]. In this study, we extend the use of the hybrid system to polyphonic SED and, more generally, to multi-label classification problems. Our approach not only allows the implicit capture of various sound event characteristics and correlation information in the time axis, through the use of deep recurrent neural networks with low-level features, but also allows to explicitly model the duration of each sound event through the use of an HSMM. This makes it possible to perform sequence-by-sequence detection without having to resort to the thresholding used in conventional frame-by-frame methods. Furthermore, in order to effectively reduce the sound event insertion errors which are often observed under noisy conditions, we additionally propose using a binary-mask-based post-processing. Specifically, we employ a sound activity detection (SAD) network which determines whether a segment contains only silence or an active sound event of any type, based on an idea inspired by the well-known benefits of voice activity detection in speech recognition systems [31]–[33].

The rest of this paper is organized as follows: Section II discusses various types of recurrent neural networks and the concept of long short-term memory. Section III explains the basics of hidden semi-Markov models. Section IV describes our proposed method in detail. In Section V, we describe the design of our experiment and compare the performance of the proposed method with conventional methods. Finally, we conclude the paper and discuss future work in Section VI.

II. Overview of Recurrent Neural Network Architectures

A. Bidirectional Long Short-Term Memory

A bidirectional long short-term memory recurrent neural network (BLSTM) is a layered network with feedback structures from both the previous time step and the following time step, whose layers consist of long short-term memory (LSTM) [34], [35]. Compared with unidirectional structures, the bidirectional structure makes it possible to propagate information not only from the past but also from the future, giving bidirectional networks the ability to exploit the full context of an input sequence. LSTM architectures prevent the so-called *vanishing gradient* problem [36] and allow the memorization of long-term context information. The structure of a BLSTM is illustrated in Fig. 1 (for simplicity of presentation, Fig. 1 and the following formulations only consider a single hidden layer). As shown in Fig. 1, LSTM layers are characterized by a *memory cell* $\mathbf{s}_t$, and three gates: 1) an *input gate* $\mathbf{g}_t^I$, 2) a *forget gate* $\mathbf{g}_t^F$, and 3) an *output gate* $\mathbf{g}_t^O$. Each gate $\mathbf{g}^*$ has a value between 0 and 1. The value 0 means the gate is closed, while the value 1 means the gate is open. The memory cell memorizes information about the past, the input gate decides whether to pass on the input, and the output gate decides whether to pass on the output. In other words, these gates prevent the propagation of unnecessary signals. The forget gate decides whether to forget the information memorized in the memory cell.

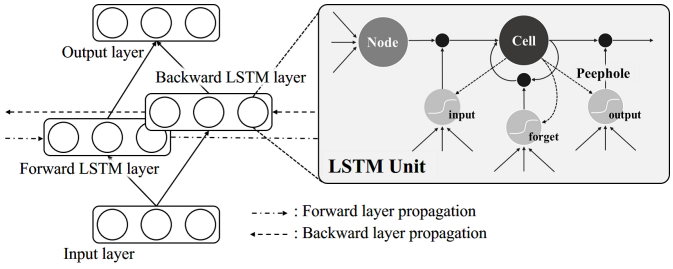


Fig. 1: BLSTM

Let us denote a sequence of feature vectors as $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$. In an LSTM layer, the output vector of the LSTM layer $\mathbf{h}_t$ is calculated as follows:

$$\mathbf{g}_t^I = \sigma(\mathbf{W}^I\mathbf{x}_t + \mathbf{W}_r^I\mathbf{h}_{t-1} + \mathbf{p}^I \odot \mathbf{s}_{t-1} + \mathbf{b}^I), \quad (1)$$

$$\mathbf{g}_t^F = \sigma(\mathbf{W}^F\mathbf{x}_t + \mathbf{W}_r^F\mathbf{h}_{t-1} + \mathbf{p}^F \odot \mathbf{s}_{t-1} + \mathbf{b}^F), \quad (2)$$

$$\mathbf{s}_t = \mathbf{g}_t^I \odot f(\mathbf{W}^C\mathbf{x}_t + \mathbf{W}_r^C\mathbf{h}_{t-1} + \mathbf{b}^C) + \mathbf{g}_t^F \odot \mathbf{s}_{t-1}, \quad (3)$$

$$\mathbf{g}_t^O = \sigma(\mathbf{W}^O\mathbf{x}_t + \mathbf{W}_r^O\mathbf{h}_{t-1} + \mathbf{p}^O \odot \mathbf{s}_t + \mathbf{b}^O), \quad (4)$$

$$\mathbf{h}_t = \mathbf{g}_t^O \odot \tanh(\mathbf{s}_t), \quad (5)$$

where superscripts I, F, O, and C indicate the input, forget, output gates, and memory cell, respectively, $\odot$ represents point-wise multiplication, σ represents a logistic sigmoid function, f represents an activation function, $\mathbf{W}^*$ and $\mathbf{W}_r^*$ denote the input weight matrices and recurrent weight matrices, respectively, $\mathbf{b}^*$ are bias vectors, and $\mathbf{p}^*$ are peephole connection weights. A peephole connection is a connection between a memory cell and a gate, which enables to control the behavior of a gate depending on the state of the memory cell. In a BLSTM layer, the output vector of the forward LSTM layer $\overrightarrow{\mathbf{h}}_t$ and that of the backward LSTM layer $\overleftarrow{\mathbf{h}}_t$ (defined similarly to the forward layer but with all sequences time-reversed) are both calculated using these equations. Finally, the output vector of the output layer $\mathbf{y}_t$ is calculated as follows:

$$\mathbf{y}_t = g\left(\overrightarrow{\mathbf{W}}\overrightarrow{\mathbf{h}}_t + \overleftarrow{\mathbf{W}}\overleftarrow{\mathbf{h}}_t + \mathbf{b}\right), \quad (6)$$

where g represents an activation function, $\overrightarrow{\mathbf{W}}$ and $\overleftarrow{\mathbf{W}}$ represent the weight matrix between the output layer and the forward LSTM layer, and between the output layer and the backward LSTM layer, respectively, while $\mathbf{b}$ denotes the bias vector of the output layer.

B. Projection Layer

In general, it is known that the deep structure of neural networks is what gives them their impressive generalization

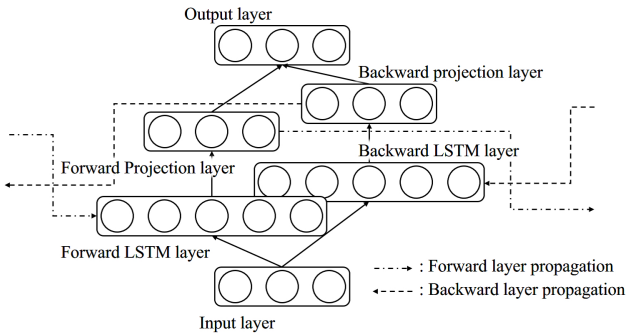


Fig. 2: BLSTM with a projection layer

power. However, building a deep LSTM network with a lot of parameters requires huge memory resources and involves high computational costs. Projection layers have been proposed as a way to address this issue and allow the creation of very deep LSTM networks [27], [28]. The use of projection layers can reduce not only the computational cost but also the effect of overfitting, improving the generalization power. The architecture of a BLSTM with a projection layer is shown in Fig. 2. The projection layer, which is a linear transformation layer, is inserted after an LSTM layer, and it outputs feedback to the LSTM layer. With the insertion of a projection layer, the hidden layer output $\mathbf{h}_{t-1}$ in Eqs. (1)-(4) is replaced with the projection layer output $\mathbf{p}_{t-1}$, and the following equation is added:

$$\mathbf{p}_t = \mathbf{W}^P \mathbf{h}_t, \tag{7}$$

where $\mathbf{W}^P$ denotes the projection weight matrix.

III. Hidden Semi-Markov Model

A. Hidden Markov Model

A hidden Markov model (HMM) is a well-known generative model which can deal with variable-length sequential data. To make the extension to the hidden semi-Markov model (HSMM) easier to understand, we first give a brief overview of HMMs. The structure of a typical left-to-right HMM is shown in Fig. 3(a). Let us assume that we have sequential observations $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$. Each HMM state $i \in \mathcal{S} = \{1, \ldots, N\}$ has an emission probability distribution $e_i(\mathbf{x})$ and the transition from state i to state j is represented by the transition probability a_{ij}. While the emission probability distribution $e_i(\mathbf{x})$ is typically modeled with a Gaussian mixture model (GMM), in a hybrid neural network/HMM model, it is calculated using a pseudo likelihood trick (see Section IV-C). The maximum likelihood estimate of a state sequence for an HMM is commonly obtained using the Viterbi algorithm. To perform the Viterbi algorithm, we introduce two variables: the forward variable $\delta_t(i)$, which represents the maximum likelihood that the partial state sequence ends at time t in state i, and the back pointer $\psi_t(i)$, which records the corresponding likelihood-maximizing pre-transition state. The forward variable $\delta_0(i)$ is initialized using an initial state probability π_i, and the back pointer $\psi_0(i)$ is initialized as 0. These are calculated recursively as follows:

Algorithm 1 HMM Viterbi algorithm

Initialization:
$\delta_0(i) = \pi_i, \ i \in \mathcal{S}$
$\psi_0(i) = 0, \ i \in \mathcal{S}$
Inference:
for $t = 1$ to T **do**
  for $j = 1$ to N **do**
    $\delta_t(j) = \max_{i \in \mathcal{S}} \{\delta_{t-1}(i) a_{ij} e_j(\mathbf{x}_t)\}$
    $\psi_t(j) = \arg\max_{i \in \mathcal{S}} \{\delta_{t-1}(i) a_{ij} e_j(\mathbf{x}_t)\}$
  end for
end for
Termination:
$P^* = \max_{i \in \mathcal{S}} \{\delta_T(i)\}$
$s_T^* = \arg\max_{i \in \mathcal{S}} \{\delta_T(i)\}$
Traceback:
for $t = T - 1$ to 1 **do**
  $s_t^* = \psi_{t+1}(s_{t+1}^*)$
end for

$$\delta_t(j) = \max_{i \in \mathcal{S}} \{\delta_{t-1}(i) a_{ij} e_j(\mathbf{x}_t)\}, \tag{8}$$

$$\psi_t(j) = \arg\max_{i \in \mathcal{S}} \{\delta_{t-1}(i) a_{ij} e_j(\mathbf{x}_t)\}. \tag{9}$$

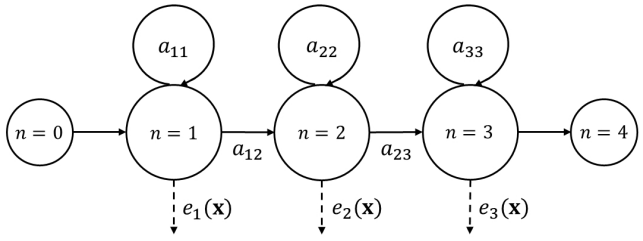
After the recursive calculation, the maximum likelihood P^* and maximum likelihood state s_T^* are calculated as follows:

$$P^* = \max_{i \in \mathcal{S}} \{\delta_T(i)\}, \tag{10}$$

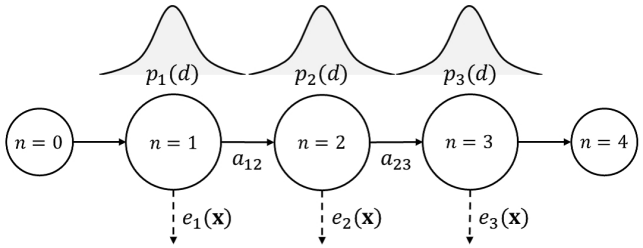
$$s_T^* = \arg\max_{i \in \mathcal{S}} \{\delta_T(i)\}. \tag{11}$$

Now, we can calculate the maximum likelihood path $\{s_1^*, s_2^*, \ldots, s_T^*\}$ using the following equation:

$$s_t^* = \psi_{t+1}(s_{t+1}^*), \tag{12}$$



(a) Hidden Markov model



(b) Hidden semi-Markov model

Fig. 3: Structural difference between an HMM and HSMM

starting at the maximum likelihood state s_T^* at time T and moving backwards. The entire process of maximum likelihood estimation in HMMs using the Viterbi algorithm is shown in Algorithm 1.

B. Hidden Semi-Markov Model

One major problem with HMMs is that they are limited in their ability to represent state duration. In HMMs, state duration probabilities are implicitly represented by state transition probabilities, therefore, HMM state duration probabilities inherently decrease exponentially with time. However, duration probabilities in real data do not necessarily follow an exponentially decreasing distribution. Consequently, the representation of duration in HMMs may be inappropriate and cause a discrepancy between the model and the data. This will be especially apparent in SED, where we need to deal with various types of sounds with various durations.

One solution to this problem is to use hidden semi-Markov models [37], [38]. The structure of an HSMM is shown in Fig. 3(b), and the difference in state duration probability between an HMM and an HSMM is shown in Fig. 4. In HSMMs, duration $d \in \mathcal{D} = \{1, 2, \ldots, D\}$ at state j is explicitly modeled by a probability distribution $p_j(d)$. The parameters of $p_j(d)$ are estimated using maximum likelihood estimation, and the maximum duration D is decided based on the mean and variance of $p_j(d)$. The Viterbi algorithm can be extended to the case of HSMMs by modifying the definitions and recurrence formulas of the forward variable δ and the back pointer ψ as follows:

$$\delta_t(j,d) = \max_{i \in \mathcal{S}, d' \in \mathcal{D}} \{\delta_{t-d}(i,d') a_{ij} p_j(d) e_j(\mathbf{x}_{t-d+1:t})\}, \tag{13}$$

$$\boldsymbol{\psi}_t(j,d) = (t-d, s^*, d^*), \tag{14}$$

where s^*, d^* and $t-d$ represent the previous state, its duration, and its end time, respectively, and s^* and d^* are calculated using the following equation:

$$(s^*, d^*) = \arg\max_{i \in \mathcal{S}, d' \in \mathcal{D}} \{\delta_{t-d}(i,d') a_{ij} p_j(d) e_j(\mathbf{x}_{t-d+1:t})\}, \tag{15}$$

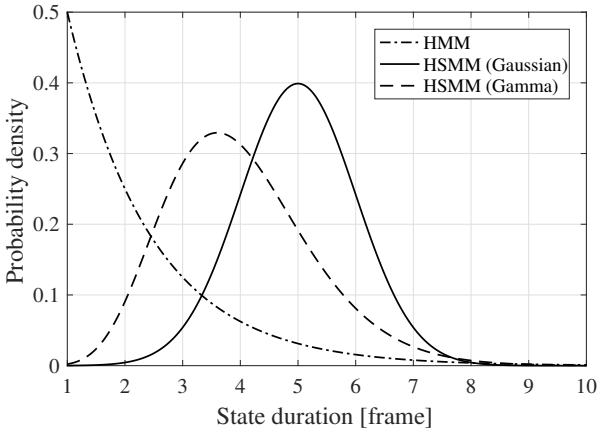


Fig. 4: Difference in duration probability

Algorithm 2 HSMM Viterbi algorithm

Initialization:

$\delta_d(i,d) = \pi_i p_i(d) e_i(\mathbf{x}_{1:d}),\ i \in \mathcal{S}, d \in \mathcal{D}$

$\boldsymbol{\psi}_d(i,d) = (0,0,0)\ i \in \mathcal{S}, d \in \mathcal{D}$

Inference:

for $t = 1$ to T **do**
  for $j = 1$ to N **do**
    for $d = 1$ to D **do**
      $\delta_t(j,d) = \max_{i \in \mathcal{S}, d' \in \mathcal{D}} \{\delta_{t-d}(i,d') a_{ij} p_j(d) e_j(\mathbf{x}_{t-d+1:t})\}$
      $(s^*, d^*) = \arg\max_{i \in \mathcal{S}, d' \in \mathcal{D}} \{\delta_{t-d}(i,d') a_{ij} p_j(d) e_j(\mathbf{x}_{t-d+1:t})\}$
      $\boldsymbol{\psi}_t(j,d) = (t-d, s^*, d^*)$
    end for
  end for
end for

Termination:

$P^* = \max_{i \in \mathcal{S}, d \in \mathcal{D}} \{\delta_T(i,d)\}$

$(s_1^*, d_1^*) = \arg\max_{i \in \mathcal{S}, d \in \mathcal{D}} \{\delta_T(i,d)\}$

Traceback:

$t_1 = T$, $n = 1$

while $t_n > 1$ **do**
  $n \leftarrow n + 1$
  $(t_n, s_n^*, d_n^*) = \boldsymbol{\psi}_{t_{n-1}}(s_{n-1}^*, d_{n-1}^*)$
end while

where the value of $e_j(\mathbf{x}_{t-d+1:t})$ is computed as follows:

$$e_j(\mathbf{x}_{t-d+1:t}) = \prod_{\tau=t-d+1}^{t} e_j(x_\tau). \tag{16}$$

After the recursive calculation, the maximum likelihood P^*, maximum likelihood state s_1^* and its duration d_1^* are calculated as follows:

$$P^* = \max_{i \in \mathcal{S}, d \in \mathcal{D}} \{\delta_T(i,d)\}, \tag{17}$$

$$(s_1^*, d_1^*) = \arg\max_{i \in \mathcal{S}, d \in \mathcal{D}} \{\delta_T(i,d)\}. \tag{18}$$

Finally, we can obtain the maximum likelihood path $\{s_N^*, s_{N-1}^*, \ldots, s_1^*\}$ and its duration $\{d_N^*, d_{N-1}^*, \ldots, d_1^*\}$ by applying the following equation recursively:

$$(t_n, s_n^*, d_n^*) = \boldsymbol{\psi}_{t_{n-1}}(s_{n-1}^*, d_{n-1}^*) \tag{19}$$

The entire HSMM Viterbi algorithm procedure is shown in Algorithm 2. Note that the ability of the HSMM to explicitly model the duration of each state by introducing the duration probability distribution $p_j(d)$ comes at the expense of an increase in the computational cost of the Viterbi algorithm from $O(NT)$ to $O(NTD)$, as compared with an HMM.

IV. Proposed Method

A. System Overview

An overview of our proposed system, separated into training and test phases, is shown in Fig. 5. In the training phase, we extract the feature vectors and perform cepstral mean normalization (CMN) for each training audio sample (see Section IV-B), and divide each active event into three segments

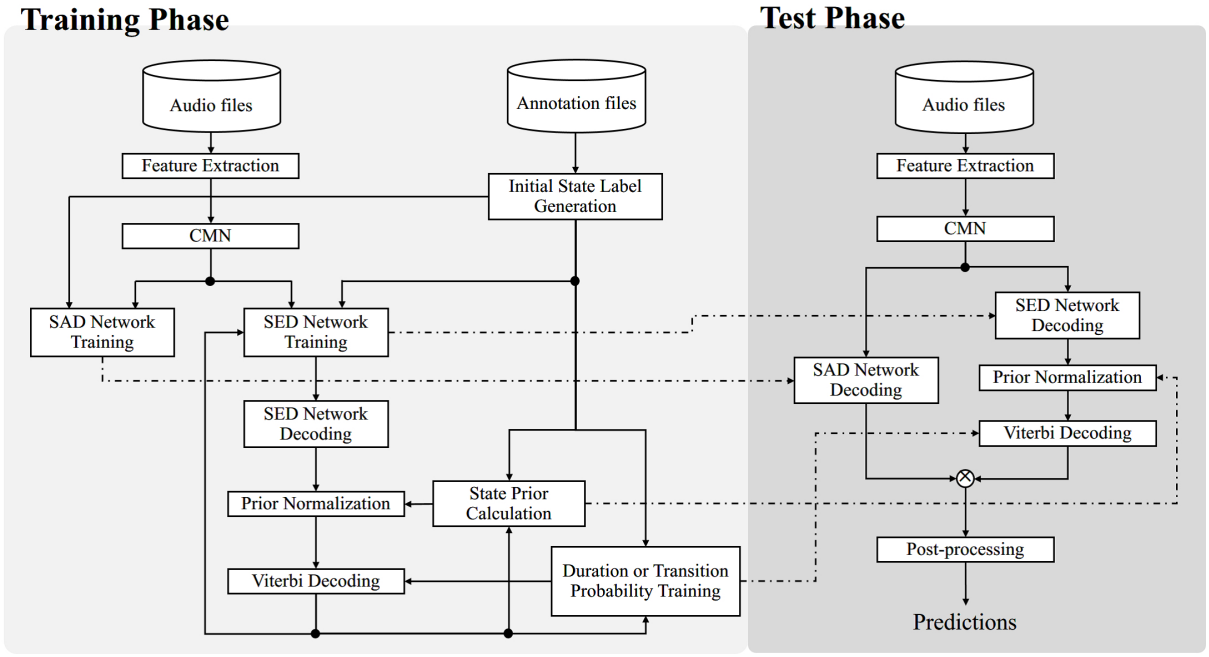
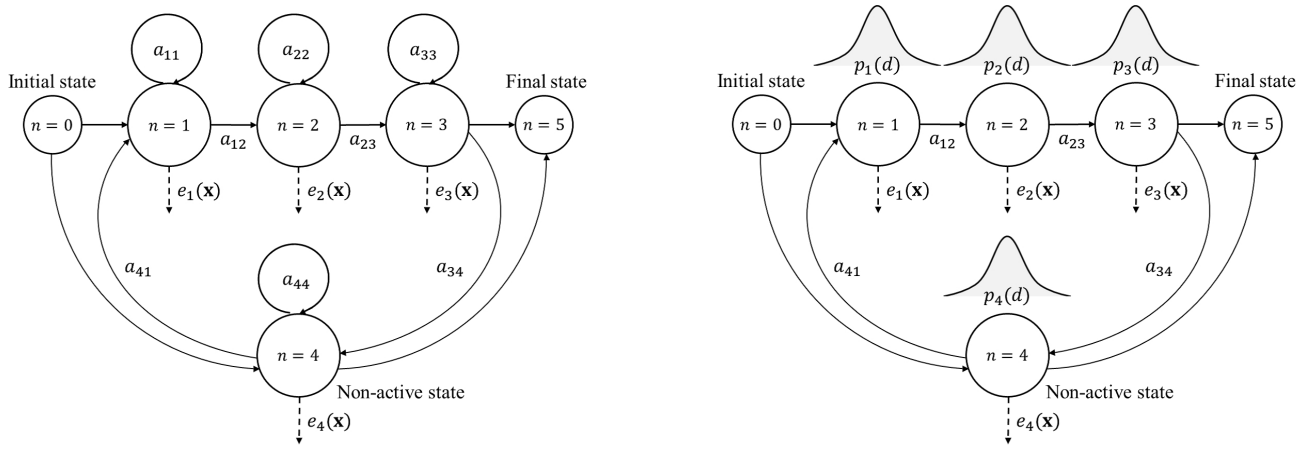


Fig. 5: System overview



(a) Proposed hidden Markov model

(b) Proposed hidden semi-Markov model

Fig. 6: Difference between proposed HMM and HSMM

of equal length in order to assign left-to-right states, thus obtaining initial state labels for the HMM (or HSMM). Using the feature vectors and state labels, we train a sound event detection (SED) network (see Section IV-C), and a sound activity detection (SAD) network (see Section IV-D). The labels are also utilized for calculating the priors of HMM and HSMM states, as well as for training the transition probability of the HMM or the duration probability distribution of the HSMM (see Section IV-C). After training, we perform the Viterbi decoding to update the state labels for the training data, and then repeat the training of the SED network and the transition (or duration) probabilities several times.

In the test phase, we extract the feature vectors from an input audio sample and perform CMN. The feature vectors are used as inputs for both the SED network and the SAD network. The SED network estimates each state posterior, while the SAD network estimates a binary mask which indicates global sound event activity, i.e., whether one or more sound events of any types are active in a given segment. Finally, we apply this binary mask to the activation of each sound event obtained using Viterbi decoding (see Section IV-D), and perform more post-processing (see Section IV-E).

B. Feature Extraction

The input signal is divided into 25 ms windows with 40% overlap, and 100 log mel-filterbank features are then calculated for each window. We use more bands than usual since the resolution of high frequency components is important for SED. Next, we perform cepstral mean normalization for each piece of training data, obtaining an input feature vector $\mathbf{x}_t$ at each frame t. These operations are performed using HTK [39].

C. Hybrid Model

We utilize two hybrid systems, BLSTM-HMM and BLSTM-HSMM, to capture sound-event-dependent temporal

structures explicitly and also to perform sequence-by-sequence detection without the thresholding that is necessary when using conventional frame-by-frame methods. While the BLSTM-HMM implicitly models the duration of each sound event via its transition probabilities, the BLSTM-HSMM can explicitly do so via its duration probabilities. We also extend the hybrid neural network/HMM framework, which is generally used to handle multi-class classification problems, to handle multi-label classification problems for polyphonic SED. To do this, we build a three-state, left-to-right HMM (or HSMM) with a non-active state for each sound event. The structures of our HMM and HSMM are shown in Figs. 6(a) and 6(b), respectively, where $n = 0$, $n = 5$ and $n = 4$ represent the *initial state*, *final state*, and *non-active state*, respectively. Note that the non-active state only pertains to the absence of activity for a particular sound event, and does not indicate whether other sound events are active or not.

For the HMM, the transition probabilities are learned from the sequences of state labels, where we simply calculate the number of transitions from state i to state j and normalize it to meet the definition of a probability. On the other hand, for the HSMM, the transition probabilities of the left-to-right states are fixed at 0.99 (the remaining 0.01 corresponding to the self-loop), and that of the non-active state is fixed at 0.01 (with the remaining 0.99 corresponding to the self-loop). These transition probabilities were determined through preliminary experiments. We represent duration using the gamma distribution in the three left-to-right states and a uniform distribution in the non-active state, defining the duration probability $p_{c,j}(d)$ for state j of the HSMM for event c as:

$$p_{c,j}(d) = \begin{cases} d^{k_{c,j}-1} \dfrac{\exp(-\theta_{c,j} d)}{\theta_{c,j}^{k_{c,j}} \Gamma(k_{c,j})} & (j = 1, 2, 3) \\ \dfrac{1}{D_c} & (j = 4) \end{cases}, \tag{20}$$

where $k_{c,j}$ and $\theta_{c,j}$ respectively denote the shape and scale parameters of the gamma distribution, obtained through maximum likelihood estimation using the state labels, and D_c is the maximum duration length of the three left-to-right states for event c. In this study, D_c is determined as follows:

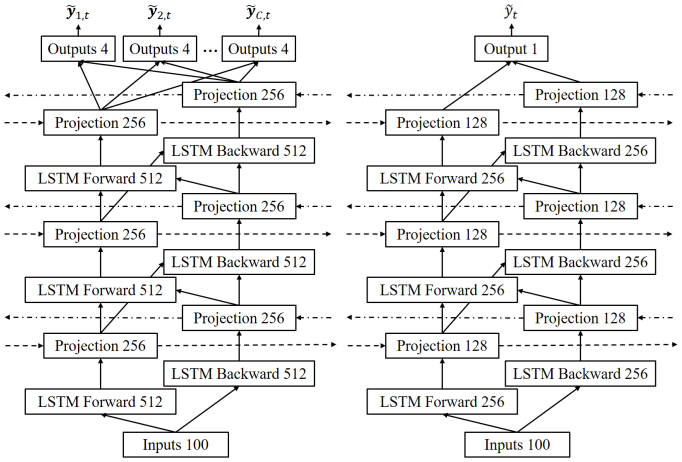
$$D_c = \max_{j \in \{1,2,3\}} \left\{ \mu_{c,j} + 3\sigma_{c,j} \right\}, \tag{21}$$

where $\mu_{c,j}$ and $\sigma_{c,j}$ respectively denote the mean and standard deviations of the duration of state j of event c.

In our hybrid model, the network is used to calculate the state posterior $P(s_{c,t} = j|\mathbf{x}_t)$, where $c \in \{1, 2, \ldots, C\}$ denotes the sound event index, $j \in \{1, 2, \ldots, N\}$ the index of states except for the initial and final states, and $s_{c,t}$ represents the state of event c at time t. The emission probability $e_{c,j}(\mathbf{x}_t)$ for state j of event c can be obtained from the state posterior using Bayes' theorem as follows:

$$e_{c,j}(\mathbf{x}_t) = P(\mathbf{x}_t|s_{c,t} = j) = \frac{P(s_{c,t} = j|\mathbf{x}_t)P(\mathbf{x}_t)}{P(s_{c,t} = j)}, \tag{22}$$

where $P(s_{c,t})$ represents the prior of HMM (or HSMM) states. Note that the factor $P(\mathbf{x}_t)$ is irrelevant in the Viterbi computations. The structure of our proposed network is shown in Fig. 7(a), where $\tilde{\mathbf{y}}_{c,t}$ represents the state posterior of event c at time t. This network has three hidden layers, each consisting



(a) SED network (b) SAD network

Fig. 7: Proposed network structures

of a BLSTM layer with 1,024 nodes and a projection layer with 512 nodes, and an output layer with $C \times N$ nodes. These network structures are determined through preliminary experiments. A softmax operation is used to ensure that the values of posterior $P(s_{c,t}|\mathbf{x}_t)$ sum to one for each sound event c in frame t, as follows:

$$P(s_{c,t} = j|\mathbf{x}_t) = \frac{\exp(a_{c,j,t})}{\sum_{j'=1}^{N} \exp(a_{c,j',t})}, \tag{23}$$

where a represents the activation of the output layer node. The network is optimized using back-propagation through time (BPTT) [40] with Adam [41] and dropout [42] using cross-entropy as shown in the following *multi-class, multi-label* objective function:

$$E(\mathbf{\Theta}) = -\sum_{c=1}^{C} \sum_{j=1}^{N} \sum_{t=1}^{T} y_{c,j,t} \ln(P(s_{c,t} = j|\mathbf{x}_t)), \tag{24}$$

where $\mathbf{\Theta}$ represents the set of network parameters, and $y_{c,j,t}$ is the state label. This objective function can be seen as a kind of multi-task learning [43] of multi-class classification problems. Multi-task learning is a technique to solve a task with additional tasks using a shared input representation, and the effectiveness has been confirmed in various fields such as speech recognition [44], and natural language processing [45]. Note that this objective function is not the same as the multi-class objective function in a conventional DNN-HMM because our SED network estimates state posteriors with respect to each sound event HMM (or HSMM), not the state posteriors of states of all HMMs (or HSMMs). State prior $P(s_{c,t})$ is calculated by counting the number of occurrences of each state using the training data labels. However, in this study, since our synthetic training data does not represent actual sound event occurrences, the prior obtained from occurrences of each state has to be made less sensitive. Therefore, we smooth $P(s_{c,t})$ as follows:

$$\hat{P}(s_{c,t}) = P(s_{c,t})^{\alpha}, \tag{25}$$

where α is a smoothing coefficient. In this study, we set α to 0.01. Finally, we calculate the state emission probability using Eq. (22) and perform the Viterbi algorithm for each HMM (or HSMM) to obtain the maximum likelihood path as shown in Section III.

D. SAD Network

A common problem when performing polyphonic SED under noisy conditions is a decrease in performance due to insertion errors, which occur when background noise is mistaken for sound events, even though there are no actual sound events in that segment. To solve this problem, we propose the use of binary masking with a sound activity detection (SAD) network. The SAD network identifies segments in which there is sound event activity of any type, similarly to voice activity detection (VAD) in the field of speech recognition [31], [32]. In this study, we train the network shown in Fig. 7(b). This network has three hidden layers, each consisting of a BLSTM layer with 512 nodes, a projection layer with 256 nodes, and an output layer with a single node. The structure of the SAD network is almost the same as that of the SED network, however the SAD network does not differentiate between the types of the sound events, and therefore it tends to concentrate on background noise. The SAD network, which performs a simple binary classification, is optimized using BPTT with Adam and dropout under the following sigmoid cross-entropy objective:

$$E(\mathbf{\Theta}) = -\sum_{t=1}^{T} \{y_t \ln(\tilde{y}_t) + (1 - y_t) \ln(1 - \tilde{y}_t)\}, \qquad (26)$$

where y_t represents the reference data indicating the presence or absence of sound events and $\tilde{y}_t$ is the SAD network output. We use a threshold of 0.5 to convert SAD network outputs into a binary mask $\mathbf{M}$, and apply it to the activations $\mathbf{A}_c$ of each sound event c predicted by the BLSTM-HMM (or BLSTM-HSMM), as follows:

$$\tilde{\mathbf{A}}_c = \mathbf{M} \odot \mathbf{A}_c, \qquad (27)$$

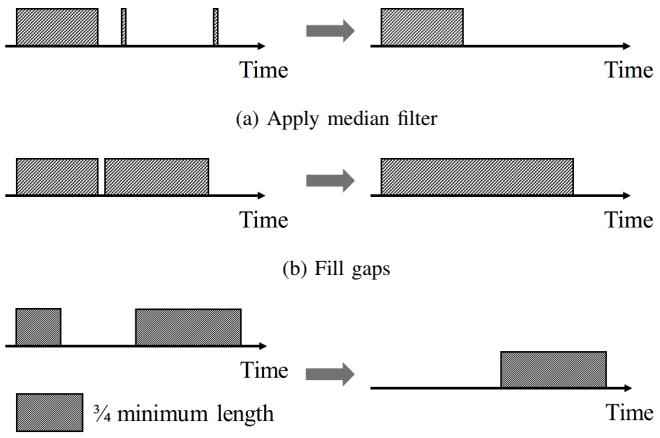
where both $\mathbf{A}_c$ and $\mathbf{M}$ are a binary vector of length T. Note that the same binary mask $\mathbf{M}$ is applied to the activations of each sound event, and that the binary mask only has an effect on the insertion of sound events, not on the substitution or deletion of sound events.

E. Post-processing

In order to smooth the output sequence, we perform three kinds of post-processing:

1) Apply a median filter with a predetermined filter span;
2) Fill gaps which are shorter than a predetermined number;
3) Remove events whose duration is shorter than a predetermined length.

An outline of each post-processing step is illustrated in Fig. 8. Based on preliminary experiments, we set the median filter span to 150 ms (15 frames), the acceptable gap length to 0.1 s (10 frames), and the minimum duration threshold length for each sound event to 3/4 of the minimum duration for that sound event as calculated from the training data.



(a) Apply median filter

(b) Fill gaps

(c) Remove short duration events

Fig. 8: Outline of each post-processing step

V. Experiments

To evaluate our proposed method, we used the DCASE2016 task 2 dataset [12]. The dataset includes a training set consisting of 20 clean audio files per event class, a development set consisting of 18 synthesized audio files of 120 seconds in length, and an evaluation set consisting of 54 synthesized audio files of the same length as the development set files. The 54 files of the evaluation set consist of 18 audio files with different content each synthesized under three different signal-to-noise ratio (SNR) conditions: −6 dB, 0 dB, and 6 dB. Out of 54 files, 27 are monophonic, and the rest are polyphonic. The number of sound event classes in the dataset is 11. The evaluation set is synthesized using unknown samples, but the development set is synthesized using the training set, making it a closed condition development set.

For this study, we chose to further split the training data, keeping 75% of the samples to build our training set, and holding out the rest in order to build an open condition development set, which was lacking in the original dataset. By open condition development set, we refer here to a development set that is based on data not included in the (newly defined) training set. We thus randomly selected 5 samples

TABLE I: Experimental conditions

Sampling rate	44,100 Hz
Window size	25 ms
Shift size	10 ms
# training data	4 s × 100k samples
# development data	120 s × 18 samples
# evaluation data	120 s × 54 samples
# sound event classes	11
Learning rate	0.001
Initial scale	0.001
Gradient clipping norm	5
Batch size	128
Time steps	400
Optimization method	Adam [41]

TABLE II: Experimental results

	Event-based (dev. / eval.)		Segment-based (dev. / eval.)	
Model	F1 score [%]	Error rate [%]	F1 score [%]	Error rate [%]
NMF (Baseline)	31.0 / 24.2	148.0 / 168.5	43.7 / 37.0	77.2 / 89.3
BLSTM	69.9 / 60.1	73.2 / 91.2	87.2 / 77.1	25.8 / 44.4
+ post-processing	81.5 / 71.2	35.7 / 50.9	89.3 / 79.0	20.3 / 36.8
+ SAD binary masking	82.2 / 73.7	34.0 / 45.6	89.5 / 79.9	19.8 / 34.3
BLSTM-HMM	80.0 / 71.0	38.6 / 55.1	88.8 / 79.6	20.4 / 37.4
+ post-processing	81.3 / 71.7	35.2 / 52.3	89.1 / 79.5	19.4 / 36.7
+ SAD binary masking	82.6 / 74.9	32.7 / 44.7	89.7 / 80.5	18.3 / 33.8
BLSTM-HSMM	82.3 / 71.9	34.7 / 51.7	91.3 / 79.8	16.7 / 37.0
+ post-processing	82.3 / 72.1	34.4 / 51.4	91.1 / 79.7	16.7 / 37.0
+ SAD binary masking	**85.0** / **75.3**	**28.9** / **44.2**	**91.5** / **81.1**	**15.8** / **32.9**
I. Choi *et al.* [24]	– / 67.1	– / 61.8	– / 78.7	– / 36.7
T. Komatsu *et al.* [18]	– / 73.8	– / 46.2	– / 80.2	– / 33.1

(out of 20) per event from the training set and generated 18 samples of 120 seconds in length, similar to the original DCASE2016 task 2 development set. These generated samples are used as development data to check performance under open conditions. We used the remaining 15 samples per class to build our own training data. Instead of simply using the original training data, which is too small for training an RNN with sufficient depth, we augmented the training data by synthetically generating our own training data using the clean samples and background noise as follows:

1) Generate a silence signal of a predetermined length;
2) Randomly select a sound event sample;
3) Add the selected sound event to the generated silence signal at a randomly selected location;
4) Repeat steps 2 and 3 a predetermined number of times;
5) Add a background noise signal at a predetermined SNR.

A. Experimental Conditions

We set the signal length to 4 s, the total number of events to a value from 3 to 5, the number of simultaneous events to a value from 1 to 5 (1 corresponding to the monophonic case), and SNR to a value from −9 dB to +9 dB. We then synthesized 100,000 audio samples, for a total length of 111 hours. Both *event-based* (onset-only) and *segment-based* evaluation are conducted, and F1 scores (F1) and error rates (ER) were utilized as the evaluation criteria. Event-based evaluation considers true positives, false positives and false negatives with respect to event instances, while segment-based evaluation is done in a fixed time grid, using segments of one second length to compare the ground truth and the system output (see [46] for more details). We built our proposed hybrid system using the following procedure:

1) Divide an active event into three segments of equal length in order to assign left-to-right state labels;
2) Train the SED network using these state labels as supervised data;
3) Calculate the prior using these state labels;
4) (For HMM) Train the transition probability using these state labels by Viterbi training;
   (For HSMM) Train the duration probability distribution using the maximum likelihood estimation;
5) Calculate the maximum likelihood path with the Viterbi algorithm;
6) Use the maximum likelihood paths as new state labels;
7) Repeat steps 2-6.

In this study, when calculating the maximum likelihood path, we fixed the alignment of non-active states, i.e., we only aligned event-active HMM states because we know the perfect alignment of non-active states due to the use of synthetic data. When training the networks, we monitored the objective functions on the development set at every epoch and stopped training accordingly, using an *early stopping* strategy. All networks were trained using the open source toolkit TensorFlow [47] with a single GPU (Nvidia GTX Titan X)[1]. Details of the experimental conditions are shown in Table I.

B. Experimental Results

To confirm the performance of our proposed method, we compared it with two conventional methods, NMF (DCASE2016 task 2 baseline) [12] and standard BLSTM [20]. The NMF was trained with 15 clean samples per class using the DCASE2016 task 2 baseline script [12]. The BLSTM had the same network structure as the one shown in Fig. 7(a), with the exception that the output layer was replaced with C nodes with sigmoid activation functions, with one node for each sound event. Each node's output $y_c \in [0, 1]$ was binarized to determine event activity. We set the threshold to 0.5, i.e., sound event c is considered to be active if $y_c > 0.5$, and inactive otherwise.

Experimental results are shown in Table II. First, we focus on the differences in the performance of each system. Our proposed system (a BLSTM-HSMM with post-processing and SAD binary masking) achieved the best performance of all of the methods for both event-based and segment-based evaluation criteria, and also outperformed all of the methods submitted to the DCASE2016 task 2 Challenge [12]. Note

[1]Our proposed system implementation is available at: https://github.com/kan-bayashi/dcase2016task2

that the conventional systems [18], [24] were trained using the whole training set, whereas our systems are trained using only part of the training set in order to hold out data for the preparation of an open development set. From these results we can see that it is important for polyphonic SED methods to take the duration of sound events into account, especially by explicitly modeling event duration.

Next, we focus on the effect of post-processing. For the BLSTM method, we can confirm that post-processing was clearly effective. However, this could not always be confirmed for our proposed hybrid models: this could be expected, because the prediction results are already smoothed as a result of the use of HMM or HSMM. In other words, the use of a dynamic modeling technique such as HMM or HSMM can effectively smooth the output sequence, and therefore, it can alleviate the need for post-processing.

Finally, we focus on the effect of SAD network binary masking. Our results confirmed the effectiveness of our proposed SAD masking method when used with all methods, especially for reducing the error rate on the evaluation set. This is because there is more background noise in the evaluation set than in the development set, leading to many insertion errors. (Note that the SAD network by itself achieved an F1 score of 97.7% for the development set and 94.8% for the evaluation set in frame-wise detection.) It is interesting to note that even though we used almost the same network architecture and exactly the same data, we could obtain an improvement in performance by using a slightly different objective function between the SED and SAD networks. This is because using a simple objective function for the SAD network makes it easy to train the network, and combining different tendency models has a performance effect similar to a model ensemble technique. This results also implies that multi-task learning can be used effectively, i.e., the objective function of an SAD network can be utilized as a regularization term in the objective function of an SED network. We will investigate this further in future work.

C. Analysis

In this section, we discuss the details of the performance of the BLSTM-based methods. Class-wise segment-based error rates are shown in Fig. 9, which displays the rates after both post-processing and SAD binary masking have been applied. Focusing first on differences in performance for various sound events, we note that detection performance for *doorslam*, *drawer*, and *pageturn* were very low. This is because the volume of the *doorslam* and *drawer* sound events were very low, making them difficult to detect when there was background noise, resulting in many deletion errors. Even humans have difficulty detecting these sounds under low SNR conditions. The poor results for *pageturn* occurred for a different reason, however. One reason for the low detection performance for *pageturn* was the large number of insertion errors due to the difference in the background noise between the training and evaluation sets. The background noise in the development set is almost the same as in the training set, and in that case we did not observe a large number of insertion errors. However,

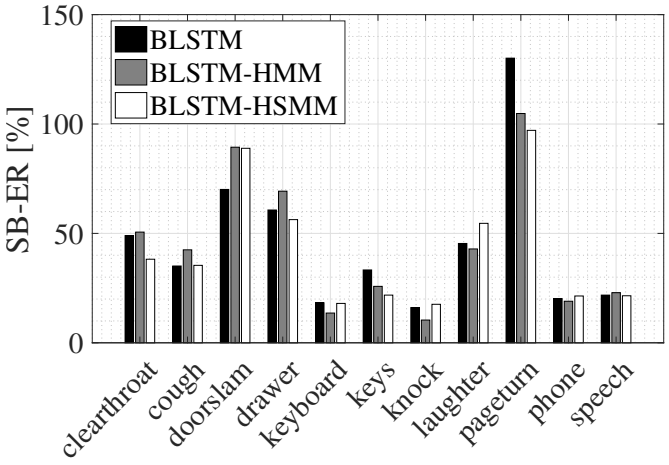


Fig. 9: Class-wise segment-based error rates

background noise in the evaluation set is different from that in the training set, so there is a possibility that the network focused on a specific pattern in the feature vector, similar to one observed in the background noise of the evaluation set.

Focusing now on the differences in performance among BLSTM-based methods, we can see that the overall trends for each model were similar. However, the combination of the BLSTM with an HMM or HSMM was not always effective for all of the sound event classes. To investigate this in detail, performance under different SNR conditions was examined and the results are shown in Table III, which again displays the performance after both post-processing and SAD binary masking have been applied.

TABLE III: Performance under different SNR conditions

	Segment-based (BLSTM / HMM / HSMM)	
SNR [dB]	F1 score [%]	Error rate [%]
6	81.2 / 80.8 / 82.9	32.1 / 32.6 / 28.6
0	80.2 / 80.8 / 81.6	33.7 / 33.4 / 31.7
-6	78.2 / 79.9 / 78.8	37.2 / 35.3 / 38.4

From these results we can see that the performance of all of the models degraded under the low SNR condition, and that the BLSTM-HSMM method was especially susceptible to the effect of background noise. The details of the segment-based error rates without SAD masking are shown in Table IV, and those with SAD masking are shown in Table V, where S, D, and I represent the substitution, deletion, and insertion error rates, respectively.

TABLE IV: Details of segment-based error rates without SAD masking

	Segment-based Error rate (BLSTM / HMM / HSMM)		
SNR [dB]	S [%]	D [%]	I [%]
6	8.2 / 8.0 / 7.1	4.6 / 5.4 / 3.6	22.2 / 22.1 / 20.7
0	8.9 / 7.6 / 7.6	4.9 / 5.2 / 4.5	22.3 / 23.1 / 23.1
-6	9.0 / 7.8 / 8.2	7.6 / 6.6 / 5.3	22.7 / 24.4 / 30.9

TABLE V: Details of segment-based error rates with SAD masking

	Segment-based Error rate (BLSTM / HMM / HSMM)		
SNR [dB]	S [%]	D [%]	I [%]
6	8.2 / 8.2 / 7.3	4.8 / 5.9 / 5.5	19.1 / 18.7 / 15.9
0	8.7 / 7.6 / 7.4	5.4 / 6.0 / 6.0	19.5 / 19.8 / 18.4
-6	9.0 / 7.6 / 7.8	7.9 / 7.1 / 6.2	20.3 / 20.6 / 24.5

Focusing on the substitution error rate, we can confirm that the combination of the BLSTM with an HMM or HSMM was always effective for the reduction of substitution errors. This is because each sound event has a different duration, and therefore substitution errors can be reduced by modeling the duration of each sound event. However, regarding the insertion error rate, while the BLSTM and BLSTM-HMM maintained their performance with only a slight degradation under the low SNR condition, the performance of BLSTM-HSMM fell drastically. This is because the BLSTM-HSMM method models the duration of sound events, forcing them in particular to sustain a certain length, and the mistakenly inserted sound events were thus of significant duration. This caused long-term frame mistakes, and consequently, the performance decreased drastically. The use of SAD network binary masking improved performance by reducing the number of insertion errors, however, there was still a gap between the performance of the BLSTM-HSMM and the other methods under the low SNR condition. As a result, although the BLSTM-HSMM method achieved the best results on average, by examining details of the performance of each method we can see that each model has advantages and disadvantages. Hence, we hypothesized that it would be advantageous to develop a model which combines the feature of each of the BLSTM-based methods. To confirm this, we devised a system combination in which the majority result of the outputs of the three methods was selected as the output, after which post-processing and SAD masking were applied. Performance results for the system combination are shown in Table VI, where the numbers in parentheses represent the amount of improvement from the best results in Table II.

TABLE VI: Performance of system combination

Criteria	F1 score [%]	Error rate [%]
Event-based	76.3 (+1.0)	42.0 (-2.2)
Segment-based	82.4 (+1.3)	30.2 (-2.7)

The combination of the three methods achieved the best performance, which supports our above hypothesis.

Finally, we focus on differences in performance when performing monophonic versus polyphonic SED task. The results are shown in Table VII.

TABLE VII: Performance for different tasks

	Segment-based (BLSTM / HMM / HSMM)	
Task	F1 score [%]	Error rate [%]
polyphonic	79.2 / 79.5 / 80.0	34.7 / 34.0 / 33.0
monophonic	81.1 / 82.4 / 82.9	33.7 / 33.4 / 32.8

Somewhat surprisingly, all of the models achieved similar performance on both tasks, and there was only a slight difference in performance between the monophonic task and the much more challenging polyphonic task.

VI. Conclusion

In this study, we proposed a new hybrid approach for polyphonic SED called duration-controlled LSTM, which incorporates a duration-controlled modeling technique based on an HSMM and a frame-by-frame detection method based on a BLSTM. Our proposed duration-controlled LSTM made it possible to model the duration of each sound event explicitly and also allowed us to perform sequence-by-sequence detection without having to resort to the kind of thresholding necessary in conventional frame-by-frame methods. Furthermore, to reduce the insertion errors which often occur under noisy conditions we also proposed the use of a binary-mask-based post-processing step using an SAD network to identify segments with any kind of sound event activity. Our proposed method outperformed not only conventional methods such as NMF and standard BLSTM, but also the best results submitted for the DCASE2016 task2 Challenge. Furthermore, we confirmed that combining the three BLSTM-based methods brings about further improvement.

In future work, we will develop a model which encompasses advantages of each BLSTM-based method. We also would like to investigate multi-task training in which the objective function of an SAD network is utilized as a regularization term in the objective function of the SED network. Additionally, we plan to apply our proposed method to a larger real-world recording dataset, and to investigate the use of sequential discriminative training for both the BLSTM-HMM and BLSTM-HSMM methods [29], [48].

References

[1] M. Xu, C. Xu, L. Duan, J. S. Jin, and S. Luo, "Audio keywords generation for sports video analysis," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 4, no. 2, p. 11, 2008.

[2] J. A. Stork, L. Spinello, J. Silva, and K. O. Arras, "Audio-based human activity recognition using non-Markovian ensemble voting," in *IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, 2012, pp. 509–514.

[3] Y.-T. Peng, C.-Y. Lin, M.-T. Sun, and K.-C. Tsai, "Healthcare audio event classification using hidden Markov models and hierarchical hidden Markov models," in *IEEE International Conference on Multimedia and Expo*. IEEE, 2009, pp. 1218–1221.

[4] 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*. IEEE, 2005, pp. 4–8.

[5] T. Hayashi, M. Nishida, N. Kitaoka, and K. Takeda, "Daily activity recognition based on DNN using environmental sound and acceleration signals," in *European Signal Processing Conference*. IEEE, 2015, pp. 2306–2310.

[6] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, "Audio context recognition using audio event histograms," in *European Signal Processing Conference*. IEEE, 2010, pp. 1272–1276.

[7] D. Valtchev and I. Frankov, "Service gateway architecture for a smart home," *IEEE Communications Magazine*, vol. 40, no. 4, pp. 126–132, 2002.

[8] L. M. Aiello, R. Schifanella, D. Quercia, and F. Aletta, "Chatty maps: constructing sound maps of urban areas from social media data," *Open Science*, vol. 3, no. 3, p. 150690, 2016.

[9] R. Stiefelhagen, K. Bernardin, R. Bowers, R. T. Rose, M. Michel, and J. Garofolo, "The CLEAR 2007 evaluation," in *Multimodal Technologies for Perception of Humans*. Springer, 2008, pp. 3–34.

[10] "TREC Video Retrieval Evaluation: TRECVID," http://www-nlpir.nist.gov/projects/trecvid/.

[11] "Detection and Classification of Acoustic Scenes and Events 2013 - DCASE 2013," http://c4dm.eecs.qmul.ac.uk/sceneseventschallenge/.

[12] "Detection and Classification of Acoustic Scenes and Events 2016 - DCASE 2016," http://www.cs.tut.fi/sgn/arg/dcase2016/.

[13] J. Schröder, B. Cauchi, M. R. Schädler, N. Moritz, K. Adiloglu, J. Anemüller, S. Doclo, B. Kollmeier, and S. Goetze, "Acoustic event detection using signal enhancement and spectro-temporal feature extraction," in *IEEE Workshop Applications of Signal Processing to Audio and Acoustics*, 2013.

[14] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, "Context-dependent sound event detection," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2013, no. 1, pp. 1–13, 2013.

[15] T. Heittola, A. Mesaros, T. Virtanen, and A. Eronen, "Sound event detection in multisource environments using source separation," in *Workshop on Machine Listening in Multisource Environments*, 2011, pp. 36–40.

[16] S. Innami and H. Kasai, "NMF-based environmental sound source separation using time-variant gain features," *Computers & Mathematics with Applications*, vol. 64, no. 5, pp. 1333–1342, 2012.

[17] A. Dessein, A. Cont, and G. Lemaitre, "Real-time detection of overlapping sound events with non-negative matrix factorization," in *Matrix Information Geometry*. Springer, 2013, pp. 341–371.

[18] T. Komatsu, T. Toizumi, R. Kondo, and Y. Senda, "Acoustic event detection method using semi-supervised non-negative matrix factorization with mixtures of local dictionaries," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop*, September 2016, pp. 45–49.

[19] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multi label deep neural networks," in *International Joint Conference on Neural Networks*. IEEE, 2015, pp. 1–7.

[20] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2016, pp. 6440–6444.

[21] M. Espi, M. Fujimoto, K. Kinoshita, and T. Nakatani, "Exploiting spectro-temporal locality in deep learning based acoustic event detection," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2015, no. 1, p. 1, 2015.

[22] Y. Wang, L. Neves, and F. Metze, "Audio-based multimedia event detection using deep recurrent neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2016, pp. 2742–2746.

[23] F. Eyben, S. Böck, B. Schuller, A. Graves *et al.*, "Universal onset detection with bidirectional long short-term memory neural networks." in *International Society for Music Information Retrieval Conference*, 2010, pp. 589–594.

[24] I. Choi, K. Kwon, S. H. Bae, and N. S. Kim, "DNN-based sound event detection with exemplar-based approach for noise reduction," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop*, September 2016, pp. 16–19.

[25] S. Adavanne, G. Parascandolo, P. Pertila, T. Heittola, and T. Virtanen, "Sound event detection in multichannel audio using spatial and harmonic features," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop*, September 2016, pp. 6–10.

[26] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold *et al.*, "CNN architectures for large-scale audio classification," *arXiv preprint arXiv:1609.09430*, 2016.

[27] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," *arXiv preprint arXiv:1402.1128*, 2014.

[28] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling." in *Interspeech*, 2014, pp. 338–342.

[29] Z. Chen, S. Watanabe, H. Erdogan, and J. Hershey, "Integration of speech enhancement and recognition using long-short term memory recurrent neural network," in *Interspeech*, 2015.

[30] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 6645–6649.

[31] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection," *IEEE Signal Processing Letters*, vol. 6, no. 1, pp. 1–3, 1999.

[32] J. Ramırez, J. C. Segura, C. Benıtez, A. De La Torre, and A. Rubio, "Efficient voice activity detection algorithms using long-term speech information," *Speech Communication*, vol. 42, no. 3, pp. 271–287, 2004.

[33] M. Fujimoto, K. Ishizuka, and T. Nakatani, "A voice activity detection based on the adaptive integration of multiple speech features and a signal decision scheme," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2008, pp. 4441–4444.

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

[35] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[36] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[37] S. Z. Yu, "Hidden semi-Markov models," *Artificial Intelligence*, vol. 174, no. 2, pp. 215–243, 2010.

[38] Z. Heiga, K. Tokuda, T. Masuko, T. Kobayasih, and T. Kitamura, "A hidden semi-Markov model-based speech synthesis system," *IEICE Transactions on Information and Systems*, vol. 90, no. 5, pp. 825–834, 2007.

[39] "HTK Speech Recognition Toolkit," http://htk.eng.cam.ac.uk.

[40] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[41] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[42] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[43] R. Caruana, "Multitask learning," in *Learning to learn*. Springer, 1998, pp. 95–133.

[44] M. L. Seltzer and J. Droppo, "Multi-task learning in deep neural networks for improved phoneme recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 6965–6969.

[45] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *International Conference on Machine Learning*. ACM, 2008, pp. 160–167.

[46] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016.

[47] "TensorFlow - An open source software library for machine intelligence," https://www.tensorflow.org.

[48] K. Veselỳ, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks." in *Interspeech*, 2013, pp. 2345–2349.

Tomoki Hayashi received his B.E. degree in engineering and M.E. degree in information science from Nagoya University, Japan, in 2013 and 2015, respectively. He is currently a Ph.D. student at the Nagoya University. His research interests include statistical speech and audio signal processing. He received the Acoustical Society of Japan 2014 Student Presentation Award. He is a student member of the Acoustical Society of Japan, and a student member of the IEEE.

Shinji Watanabe is a Senior Principal Research Scientist at Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA. He received his Ph.D. from Waseda University, Tokyo, Japan, in 2006. From 2001 to 2011, he was a research scientist at NTT Communication Science Laboratories, Kyoto, Japan. From January to March in 2009, he was a visiting scholar in Georgia institute of technology, Atlanta, GA. His research interests include Bayesian machine learning and speech and spoken language processing. He has been published more than 100 papers in journals and conferences, and received several awards including the best paper award from the IEICE in 2003. He served an Associate Editor of the IEEE Transactions on Audio Speech and Language Processing, and is a member of several committees including the IEEE Signal Processing Society Speech and Language Technical Committee.

Tomoki Toda received his B.E. degree from Nagoya University, Japan, in 1999 and his M.E. and D.E. degrees from Nara Institute of Science and Technology (NAIST), Japan, in 2001 and 2003, respectively. He was a Research Fellow of the Japan Society for the Promotion of Science from 2003 to 2005. He was then an Assistant Professor (2005-2011) and an Associate Professor (2011-2015) at NAIST. From 2015, he has been a Professor in the Information Technology Center at Nagoya University. His research interests include statistical approaches to speech and audio processing. He received more than 10 paper/achievement awards including the IEEE SPS 2009 Young Author Best Paper Award and the 2013 EURASIP-ISCA Best Paper Award (Speech Communication Journal).

Takaaki Hori received the B.E. and M.E. degrees in electrical and information engineering from Yamagata University, Yonezawa, Japan, in 1994 and 1996, respectively, and the Ph.D. degree in system and information engineering from Yamagata University in 1999. From 1999 to 2015, he had been engaged in researches on speech recognition and spoken language understanding at Cyber Space Laboratories and Communication Science Laboratories in Nippon Telegraph and Telephone (NTT) Corporation, Japan. He was a visiting scientist at the Massachusetts Institute of Technology (MIT) from 2006 to 2007. Since 2015, he has been a senior principal research scientist at Mitsubishi Electric Research Laboratories (MERL), Cambridge, Massachusetts, USA. He has authored more than 90 peer-reviewed papers in speech and language research fields. He received the 24th TELECOM System Technology Award from the Telecommunications Advancement Foundation in 2009, the IPSJ Kiyasu Special Industrial Achievement Award from the Information Processing Society of Japan in 2012, and the 58th Maejima Hisoka Award from Tsushinbunka Association in 2013.

Jonathan Le Roux is a principal research scientist at Mitsubishi Electric Research Laboratories (MERL) in Cambridge, Massachusetts. He completed his B.Sc. and M.Sc. degrees in Mathematics at the Ecole Normale Suprieure (Paris, France), and his Ph.D. degree at the University of Tokyo (Japan) and the Université Pierre et Marie Curie (Paris, France), and worked as a postdoctoral researcher at NTT's Communication Science Laboratories from 2009 to 2011. His research interests are in signal processing and machine learning applied to speech and audio. He has authored more than 60 peer-reviewed papers in these fields. He is a founder and chair of the Speech and Audio in the Northeast (SANE) series of workshops, a Senior Member of the IEEE, and a member of the IEEE Audio and Acoustic Signal Processing Technical Committee.

Kazuya Takeda received his B.E.E., M.E.E. and Ph.D. from Nagoya University. After graduating from Nagoya University in 1985, he worked at Advanced Telecommunication Research Laboratories and at KDD R&D Laboratories, Japan, mostly in the area of speech signal processing. He was a Visiting Scientist at the Massachusetts Institute of Technology from October 1988 to April 1989. In 1995, Dr. Takeda moved to Nagoya University, where he started a research group for signal processing applications. Since then he has been working on a wide range of research topics, including acoustics and speech, as well as driving behavior. He is the co-author of more than 100 journal articles and five books. Dr. Takeda is currently a Professor at the Graduate School of Informatics and the Green Mobility Collaborative Research Center, Nagoya University, Japan.