FROM LABELED TO UNLABELED DATA – ON THE DATA CHALLENGE IN AUTOMATIC DRUM TRANSCRIPTION

Chih-Wei Wu, Alexander Lerch
Georgia Institute of Technology, Center for Music Technology
{ cwu307, alexander.lerch}@gatech.edu

ABSTRACT

Automatic Drum Transcription (ADT), like many other music information retrieval tasks, has made progress in the past years through the integration of machine learning and audio signal processing techniques. However, with the increasing popularity of data-hungry approaches such as deep learning, the insufficient amount of data becomes more and more a challenge that concerns the generality of the resulting models and the validity of the evaluation. To address this challenge in ADT, this paper first examines the existing labeled datasets and how representative they are of the research problem. Next, possibilities of using unlabeled data to improve general ADT systems are explored. Specifically, two paradigms that harness information from unlabeled data, namely feature learning and student-teacher learning, are applied to two major types of ADT systems. All systems are evaluated on four different drum datasets. The results highlight the necessity of more and larger annotated datasets and indicate the feasibility of exploiting unlabeled data for improving ADT systems.

1. INTRODUCTION

Automatic drum transcription (ADT), a sub-task of Automatic Music Transcription (AMT) [2] that concerns the extraction of drum events from music signals, witnesses a growth in data-driven approaches such as deep learning in recent years [24, 25, 31–33]. The majority of these ADT studies use the popular ENST-Drums dataset [11] for development by splitting the dataset into different subsets for training, validation, and testing purposes. Nevertheless, the limited amount of labeled data and its potential impact on ADT systems are rarely discussed. The heavy reliance on one dataset raises two major concerns: (i) the model could easily overfit the data, which questions its generality, and (ii) the evaluation results could be overly optimistic due to the small sample size of the split. To avoid these pitfalls, larger datasets and cross-dataset evaluation are necessary. This need has been identified by researchers and has been addressed with newly released annotated datasets such as MDB-Drums [26] and RBMA [33]. These new data enable us to revisit ENST-Drums and re-examine the representativeness of this widely-used dataset through a unified comparison.

Motivated by the above mentioned issues concerning the data in ADT, this paper aims to address the challenge from two different angles, (i) examining the effectiveness of the existing datasets and (ii) investigating additional resources (e.g., unlabeled data) and techniques for supporting the development of general ADT systems. The contributions of this work include: first, the examination of four different datasets, highlighting the importance of data diversity. Second, the evaluation of two paradigms for integrating unlabeled data to two major types of ADT systems. Third, the demonstration of potential improvements of both types of ADT systems on different drum instruments using unlabeled data.

2. RELATED WORK

2.1 Automatic Drum Transcription

The task of automatic drum transcription can be described as converting drum related audio events into music notation. Most of the early ADT systems, as summarized by FitzGerald and Paulus [9], detect onsets of HiHat (HH), Bass Drum (BD), and Snare Drum (SD) in drum only recordings. Recently, this focus has shifted towards transcribing drums in polyphonic mixtures comprised of both percussive and melodic instruments. Following these conventions, this paper defines the ADT task as detecting HH, BD, and SD in polyphonic mixtures.

Generally speaking, the existing ADT systems can be categorized into four types according to the literature [12, 19]. These are (i) *Segment and Classify*: following the standard pattern recognition pipeline, these approaches extract audio features from detected onset locations and classifies them with pre-trained models; this is a popular approach with many proposed systems using different combinations of classifiers and features [10, 12, 27, 28], (ii) *Separate and Detect*: deriving activation functions from recordings to represent the activities of each drum, these systems subsequentially perform onset detection on these activation functions to locate drum hits; approaches include matrix factorization methods such as Non-negative Matrix Factorization (NMF) [6, 23, 35] and deep-learning-based methods such as Recurrent Neural Networks (RNNs) [24, 31, 32] and Convolutional Neural Networks (CNNs) [25, 33], (iii) *Match and Adapt*: identifying drum events by comparing with a set of

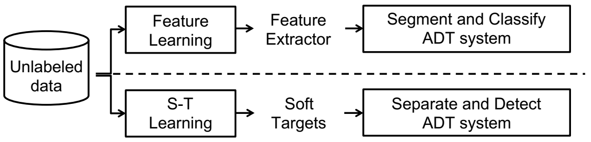

Figure 1. The overview of the evaluated paradigms for integrating unlabeled data to two major ADT approaches

pre-defined templates, these systems often iteratively update the templates [38], and (iv) *HMM-based Recognition*: modeling the temporal connections between drum events using probabilistic models such as Hidden Markov Models (HMMs), these models try to identify the underlying drum sequence by using the Viterbi algorithm [7, 20].

To date, the majority of the existing ADT systems fall into the categories of *Segment and Classify* and *Separate and Detect*. Both these types of systems, despite having fundamental differences, use data-driven methods and face the challenge described in Sect. 1. Therefore, in this paper, we considered both types of systems in our experiments.

2.2 Learning from Unlabeled Data

To address the data challenge in MIR tasks, techniques that build upon the existing labeled data have been proposed. For example, in *transfer learning* [4], a deep neural network trained on a task that has sufficient data can be used to derive features for another task with limited data. This method alleviates the data insufficiency by re-using the effective models in the similar domains. *Data augmentation*, a technique to increase diversity of training data through music-related deformations (e.g., time-stretching, pitch shifting, or distortion) and synthesis, has been successfully applied to MIR tasks [18] and in ADT specifically [32, 36]. However, these techniques still require a reasonably sized correctly annotated dataset as a starting point, which remains a challenge in certain scenarios.

Another direction for addressing the data scarcity is to use unlabeled data. Intuitively, a large collection of unlabeled data can be helpful in deriving more generalized features. This is the main concept of unsupervised *feature learning*, and it can be implemented with algorithms such as Sparse Coding [22], Deep Belief Networks [13], and Auto-encoders [17]. More recently, the *student-teacher learning* paradigm has also emerged as an interesting concept to incorporate unlabeled data. Referred by Hinton et al. as “knowledge distillation” [14], this paradigm transfers the knowledge of a teacher model to a student model using the soft-targets generated by the teacher. As opposed to learning from the hard targets (i.e., ground truth), the student learns from the “dark knowledge” residing in the soft-targets, which can be created using either labeled or unlabeled data [15]. A successful student model can reduce the complexity of the original teacher model without significant performance loss. Several studies also report superior performance of the student models [5, 34, 37]. Overall, methods that work directly with unlabeled data obviously have less dependency on existing labeled data and have higher potential to be applicable to more tasks.

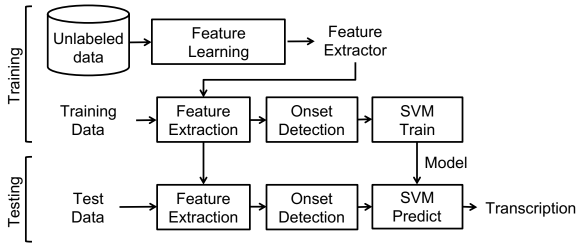

Figure 2. The flowchart of the feature learning paradigm for ADT

3. METHOD

3.1 Overview

To connect general ADT systems to the abundant resources of unlabeled data, this paper investigates the application of *feature learning* and *student-teacher learning* to *Segment and Classify*-based and *Separate and Detect*-based ADT systems, respectively. Figure 1 shows the two paradigms for integrating unlabeled data to ADT systems as investigated in this paper. The feature learning paradigm is designed for *Segment and Classify*-based ADT systems. In this paradigm, the unlabeled data is used to derive a feature extractor using an unsupervised feature learning algorithm. The resulting feature extractor is then integrated into a generic *Segment and Classify* ADT framework. The student-teacher learning paradigm is suitable for *Separate and Detect*-based ADT systems. This paradigm uses teacher models and unlabeled data to generate soft-targets; these soft-targets play the important role of connecting any *Separate and Detect*-based system with unlabeled data and enable the training of the student model. In the following sections, more details of both paradigms are presented.

3.2 Feature Learning

The flowchart in Fig. 2 shows the feature learning paradigm for ADT, including both training and testing. The training phase starts with the training of a feature extractor using the unlabeled data. Specifically, we use a Convolutional Auto-encoder (CAE) as the feature extractor. A generic *Segment and Classify*-based ADT system is then constructed with the following steps: first, the features are extracted from the audio signals using the pre-trained feature extractor. Second, the onset locations are determined by using the ground truth annotations while training. Finally, the feature vectors around the onset locations are collected and used to train three binary classifiers for HH, BD, and SD, respectively. The classifiers used in this paper are Support Vector Machines (SVMs). In the testing phase, the same pipeline is followed except for the onset detection step, which uses an onset detector instead of the ground truth locations. Finally, the presence of each drum can be predicted using the pre-trained SVMs.

The architecture of the CAE is shown in Fig. 3. The input X of the CAE is a Mel-spectrogram, and the output

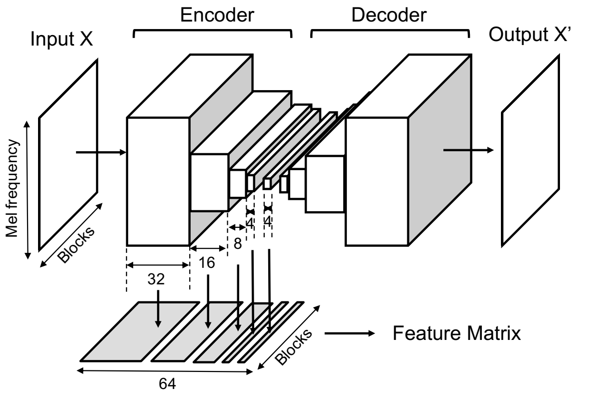

Figure 3. The architecture of the proposed CAE for unsupervised feature learning. The input X is a $128 \times N$ Mel-spectrogram.

X' is the reconstruction of X. The encoder consists of four convolutional layers with $\{32, 16, 8, 4\}$ channels of 3×3 kernels, accordingly. Each convolutional layer is used by a batch normalization layer and a max-pooling layer of (2, 1). This design maintains the temporal resolution, allowing the extraction of block-wise features. The bottleneck layer is also a convolutional layer with 4 channels of 3×3 kernels. All non-linear units are Rectified Linear Units (ReLUs). The structure of the decoder is symmetric to the encoder with the max-pooling layers replaced by the up-sampling layers. The CAE is trained to minimize the Mean Squared Error (MSE) between X and X' using a gradient-descent-based optimization process, and the number of training epochs is 30.

The feature extraction process, as shown in Fig. 3, is inspired by the method proposed by Choi et al. [4]: first, the intermediate activation maps from all the layers in the encoder (including the bottleneck layer) are computed. Next, average pooling is performed on these maps across the Mel-frequency axis. Finally, these outputs are stacked into a $64 \times N$ feature matrix, where N is the number of blocks. To derive the final feature vector at each block, the feature vectors from the current block and the following two blocks are spliced together to capture the temporal variations of the event. This leads to a final feature vector with a dimensionality $d = 3 \times F$, in which F is the number of features (i.e., 64).

In addition to the learned features, a set of baseline features consisting of 20 Mel Frequency Cepstral Coefficients (MFCCs) and their first and second derivatives is also included in this paradigm. As a result, the baseline feature vector has a dimensionality $d = 3 \times 60 = 180$ after the feature splicing.

3.3 Student-Teacher Learning

Figure 4 shows the flowchart of the student-teacher learning paradigm for ADT. In the training phase, the teacher models are used to analyze the unlabeled data and generate the soft-targets. These soft-targets, used as pseudo ground truth to train a student model, contain the activation functions for the different drums. When multiple teachers are present, the student model can be trained by iteratively passing the unlabeled data and its corresponding soft-targets from each teacher. The student model is trained by minimizing the MSE between the soft-targets and the model outputs. In the testing phase, the trained student model processes the test data and generates the corresponding activation functions. The estimated locations of drum hits are identified with a simple peak picking process.

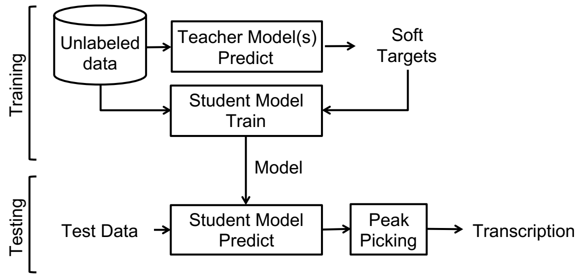

Figure 4. The flowchart of the student-teacher learning paradigm for ADT

The model architecture, configuration, and parametrization of this evaluated paradigm generally follows the setup described in [37]. This includes two teacher models based on Partially-Fixed NMF (PFNMF) [35] and one student model using a fully-connected, feed-forward Deep Neural Network (DNN). The soft-targets are scaled to a numerical range between 0 and 1 using min-max scaling across the training data for each instrument in order to ensure their compatibility with the outputs from the student DNN.

3.4 Implementation

The main input representations for both paradigms are derived from the magnitude spectrogram of the Short Time Fourier Transform (STFT), which is computed using a block size of 2048 and a hop size of 512 samples with Hann window. All of the audio signals are normalized to a range between 1 and -1, down-mixed to mono, and resampled to 44.1 kHz prior to the computation of STFT.

For the feature learning paradigm, both the Mel-spectrogram in dB scale with 128 bins and the MFCCs are computed using librosa, [1] a Python library for audio signal processing. The onset detection is implemented using the *CNNOnsetProcessor* from Madmom. [2] Additionally, the implementation of Linear SVMs from scikit-learn, [3] a Python library for machine learning, is used. A grid search on the penalty parameter C within $\{0.1, 1, 10, 100, 1000\}$ is performed to optimize the performance of the SVMs.

For student-teacher learning paradigm, the teacher models are implemented using the PFNMF function from NmfDrumToolbox. [4] The peak-picking parameters are set to the same as in the original paper [37].

[1] https://librosa.github.io, last access 2018/03/27

[2] https://madmom.readthedocs.io/en/latest/, last access 2018/03/27

[3] http://scikit-learn.org/stable/, last access 2018/03/27

[4] https://github.com/cwu307/NmfDrumToolbox, last access 2018/03/27

The neural networks in both paradigms are implemented using Keras [5] and the Tensorflow [1] backend. The weights are randomly initialized with normal distributions, and the parameters of the ADAM optimizer are set to default. The source code used in this paper is available on Github. [6]

4. EXPERIMENT

4.1 Unlabeled Data

The unlabeled dataset in this paper is built using the source code provided in [37]; this tool allows the compilation of a list of songs from the Billboard Chart [7] and the retrieval of these songs from Youtube. This dataset consists of six musical genres, including R&B/HipHop, Pop, Rock, Latin, Alternative, and Dance/Electronic. Each genre has 1900 songs, which leads to a collection of 11400 songs. All the songs are cross-checked for duplicates and converted to mp3 format with a sampling rate of 44.1 kHz. In our experiments, this dataset is further split into training, validation, and testing set with a percentage of 70%, 15%, and 15%, respectively. To speed up the process while maintaining the diversity, only a 30 s segment is extracted from each song for training. The segment starts in the middle of the song to avoid potential inactivity at the beginning. As a result, the entire training set has a total duration of 66.5 hrs, which is significantly larger than any existing ADT dataset. The list of songs and links are available on Github. [8]

4.2 Labeled Data

In this paper, four different labeled datasets featuring polyphonic mixtures are used: (i) the popular ENST-Drums (referred to as ENST) [11], (ii) the MIREX 2005 (referred to as m2005),(iii) the MDB-Drums (referred to as MDB) [26], and (iv) the RBMA dataset [33]. The latter three public sets have been used in the 2017 Music Information Retrieval Evaluation eXchange (MIREX) [9] drum transcription task.

ENST minus one subset consists of 64 recordings performed by three different drummers on their own drum kits. The average duration of the recordings is 55 s. These recordings feature different musical genres and playing styles, and the multi-track files are available for remixing. In this paper, the accompaniments are mixed with their corresponding drum tracks using a scaling factor of 1/3 and 2/3, respectively. This setup is consistent with several previous studies [24, 31, 35].

m2005 was originally collected for the first MIREX drum transcription task in 2005 and recently made available for MIREX 2017 drum transcription task participants. The public set includes 23 recordings contributed from all the participants of MIREX 2005. While covering a variety of musical genres, J-pop has the highest presence in this dataset with 10 recordings. The average duration of this dataset is 125 s.

MDB consists of 23 recordings of the MusicDelta subset from the MEDLEYDB dataset [3]. These recordings include a variety of musical genres such as Rock, Country, Disco, Reggae, and Jazz. The average duration of the recordings is 54 s. Similar to *ENST*, this dataset contains multi-track files as well as the full mixtures. In this paper, we use the full-mixtures directly without any adjustment of the mixing levels.

RBMA was released as part of the public set for the MIREX 2017 drum transcription task. This public set includes 27 recordings featuring mostly Electronic Dance Music (EDM). The average duration of the tracks is 230 s. Since this dataset focuses on electronic music, it contains electronic drum sounds that can be distinctively different from the other three datasets.

In total, there are 137 files with annotations available for evaluation. All files have a sampling rate of 44.1 kHz.

4.3 Metrics

The evaluation metrics in this paper are Precision (P), Recall (R), and F-measure (F). Only the averaged F-measure is reported due to the limited space. These metrics are implemented using *mir_eval*, a Python library of common MIR metrics [21]. To determine whether an onset is a match with the ground truth, a tolerance window of 50 ms on both sides is used. This setting is consistent with the literature [12, 24, 35], although some authors use smaller tolerance windows such as 30 ms [20] and 20 ms [32].

4.4 Experiment Setup

This paper evaluates 9 ADT systems, comprising 4 systems for the feature learning paradigm and 5 systems for the student-teacher learning paradigm. The configurations of these systems are described as follows:

For the feature learning paradigm, the 4 systems are differentiated by their features. These features are:

(i) MFCC: this set of features has shown its effectiveness in previous ADT studies [20, 27, 29]. Therefore, it is included as a baseline.

(ii) CONV-RANDOM: this set of features is extracted using the proposed CAE architecture with all the weights randomly initialized without further training. This is another baseline inspired by [4] to serve as a sanity check for the effectiveness of the unsupervised training process.

(iii) CONV-AE: this is the set of features extracted from the proposed CAE after training. During the training procedure, the original input is used as the target for optimization. In other words, the CAE is trained to reconstruct the input.

(iv) CONV-DAE: this set of features is similar to CONV-AE except for the optimization target. In this case, a processed input is used as the target. Specifically, the percussive component from the Harmonic Percussive Source Separation (HPSS) [8] algorithm is used, and the CAE is trained to approximate the percussive

[5] https://keras.io, last access 2018/03/27

[6] https://github.com/cwu307/ADT_with_unlabeledData, last access 2018/06/14

[7] https://www.billboard.com/charts, last access 2018/03/27

[8] https://github.com/cwu307/unlabeledDrumDataset, last access 2018/06/14

[9] http://www.music-ir.org/mirex/wiki/2017, last access 2018/03/27

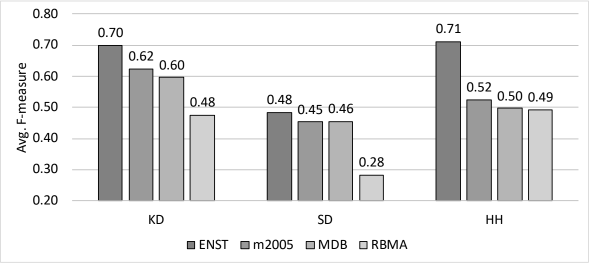

Figure 5. The evaluation results of all labeled datasets with averaged F-measure across all systems.

component. This configuration is inspired by the concept of the Denoising Autoencoder (DAE) [30] and is designed to encourage the extraction of drum-related features.

The teacher models for student-teacher learning paradigm are described in [37]. The 3 student models can be differentiated by their training data. The systems are:

(i) PFNMF (SMT): a teacher PFNMF initialized with the drum templates extracted from the IDMT-SMT-Drum dataset [6].

(ii) PFNMF (200D): a teacher PFNMF initialized with the drum templates extracted from the 200 Drum Machine dataset. [10]

(iii) FC-200: a fully-connected student DNN trained with a subset of the unlabeled dataset, which consists of 200 randomly selected songs from each genre.

(iv) FC-ALL: a fully-connected student DNN trained with all the songs from all genres.

(v) FC-ALL (ALT): a fully connected student DNN trained with all the songs from only the "Alternative" genre. This particular genre is selected for its superior performance in preliminary tests.

Based on these 9 systems, the following experiments are conducted:

E1: **Experiment 1** aims to examine the variance of the labeled datasets. For each dataset, the averaged F-measures across all 9 systems are reported.

E2: **Experiment 2** aims to evaluate the usefulness of unlabeled data for *Segment and Classify*-based ADT systems using the feature learning paradigm. For each system, the averaged F-measures across all the datasets are reported.

E3: **Experiment 3** aims to evaluate the usefulness of unlabeled data for *Separate and Detect*-based ADT systems using the student-teacher learning paradigm. For each system, the averaged F-measures across all the datasets are reported.

Note that for the feature learning paradigm, a cross-dataset validation process is performed (e.g., train on three datasets and test on the remaining one) in order to train the binary classifiers (see Sect. 3.2). For student-teacher learning paradigm, since the student model does not need additional labeled data for training so that a cross-dataset validation is unnecessary.

Experiments		**Averaged F-measure**		
Role	**System**	**HH**	**BD**	**SD**
Baseline	MFCC	0.61	**0.62**	0.40
Baseline	CONV-RANDOM	0.61	0.54	0.39
Evaluated	CONV-AE	0.61	**0.62**	**0.42**
Evaluated	CONV-DAE	0.61	0.61	**0.42**

Table 1. Evaluation results of the feature-learning-paradigm-based systems.

4.5 Results

Figure 5 shows the evaluation result of **E1**. On average, all systems tend to perform the best on *ENST* and the worst on *RBMA*. For some instruments, this gap can be as large as 22% in F-measure. There are two possible reasons for the good performance on *ENST*. First, as many ADT systems, including *Segment and Classify*-based and *Separate and Detect*-based, have been developed and evaluated on *ENST*, there could be potential bias towards this dataset. Second, the *ENST* dataset might be relatively simple compared to the others. A closer examination of the dataset shows a lack of singing voices and the dominance of MIDI synthesized accompaniments, which could potentially over-simplify the ADT problem. The relative poor performance on the *RBMA* dataset might be related to its focus on EDM; the electronic drum sounds with strong audio effects could possibly increase the difficulty for ADT. This seems to be especially true in case of the SD. Overall, the results show that the evaluated systems leave much room for optimization; since many of the parameters in these systems are not extensively tuned, this result is to be expected. However, this also reflects the challenge of building an ADT system that is easily generalizable.

The results of **E2** are shown in Table 1. The following trends can be observed: first, the unlabeled data seems to be helpful in *Segment and Classify*-based ADT systems. A direct comparison between CONV-AE and MFCC shows that the features learned from unlabeled data seem to slightly improve for SD while achieving equal performance on HH and BD. Second, the unsupervised training process is useful for deriving better features. Compared to CONV-RANDOM, both CONV-AE and CONV-DAE show improvements on nearly all instruments, indicating the advantage of the training process. Third, the DAE-inspired training process does not lead to improvements for ADT. This is shown by the almost equivalent results from CONV-AE and CONV-DAE. Since HPSS also introduces artifacts, it might not be the most ideal method for this task; experimentation with other source separation algorithms might provide more insights.

Table 3 shows the results of **E3**. The general trends can be summarized as follows: first, the student-teacher learning seems to be useful for *Separate and Detect*-based ADT systems as all students show a noticeable improvement on HH over the teacher models. This observation

[10] http://www.hexawe.net/mess/200.Drum.Machines/, last access 2018/03/27

Compared Systems		Inst.	Paradigm	Improvement		Deterioration	
Test	**Ref**			**# Files**	**F-measure Gain**	**# Files**	**F-measure Loss**
CONV-AE	MFCC	SD	Feature Learning	70/137	6.5%	40/137	-4.6%
FC-200	PFNMF (SMT)	HH	S-T Learning	78/137	13.8%	44/137	-7.6%

Table 2. Significance check of the most improved pair from each paradigm.

Experiments		Averaged F-measure		
Role	**System**	**HH**	**BD**	**SD**
Teacher	PFNMF (SMT)	0.47	0.61	**0.45**
Teacher	PFNMF (200D)	0.47	**0.67**	0.40
Student	FC-200	**0.56**	0.57	0.44
Student	FC-ALL	0.53	0.59	0.42
Student	FC-ALL (ALT)	0.55	0.58	0.44

Table 3. Evaluation results of the student-teacher-paradigm-based systems. The performance of the teacher models are the baseline.

consolidates the preliminary finding reported in [37]. Second, more unlabeled data do not necessarily lead to better results. For example, FC-200 and FC-ALL (ALT) both outperform FC-ALL on HH and SD. Since the student model is a simple feed-forward DNN, the lack of model capacity and temporal information could limit its potential for further improvement as the data size grows. Experiments using other student models (e.g., CNNs and RNNs) are necessary for confirmation. Third, the student models seem to struggle on BD. A detailed examination on the individual results from each dataset shows that teachers and students are mostly comparable on BD except for *RBMA*. This is possibly due to the challenging nature of *RBMA* as discussed in **E1**. However, further investigation is needed before drawing conclusions.

The results of **E2** and **E3** show that feature learning and student-teacher learning paradigms are able to improve the performance on SD and HH, respectively. In light of these results, an interesting question is: "Are these improvements significant?" In an attempt to answer this question, two pairs of systems are selected for further analysis. Each pair consists of the best baseline and the best evaluated system of each paradigm. A t-test is performed on each pair by comparing their results on all 137 files. Both pairs have shown significant improvements with $p \ll 0.0014$ for both t-tests. Furthermore, the number of improved and deteriorated files is calculated. The results, shown in Table 2, show a positive trend: the number of improved files is, in both cases, greater than the number of deteriorated files. Moreover, the averaged F-measure gains are also higher than the averaged F-measure loss for both pairs.

From Table 2, it can be observed that the improvements on HH from the student-teacher learning paradigm seems to be more substantial. To further investigate the cause of this improvement, one example from the *ENST* dataset, which has the largest F-measure gain among all files, is selected. The HH activation functions generated from both teacher and student model are shown in Fig. 6. Compared to the teacher's activation function, the student's activation function is sharper and less noisy, demonstrating the benefit of this paradigm.

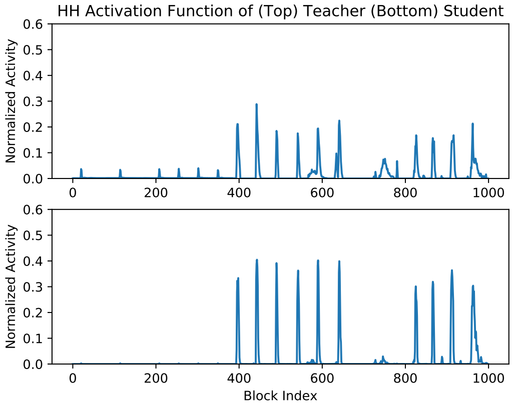

Figure 6. Example of the (top) teacher's and (bottom) student's HH activation function in comparison.

5. CONCLUSION

We discussed the data challenge in ADT and investigated two approaches to address this challenge by considering both labeled and unlabeled data. First, we compared system performance on multiple existing labeled datasets in an unified setting. The results indicate a potential bias of relying on one dataset and highlight the necessity of including more datasets in the future ADT evaluation. Furthermore, we evaluated the usefulness of unlabeled data for two major types of ADT systems via two different learning paradigms, feature learning and the student-teacher learning approach. For both paradigms, we got encouraging (and statistically significant) results demonstrating the potential of achieving better performance than the baseline systems on different drum instruments.

These results, while suggesting the need for additional labeled data in the field of ADT, also encourage the exploration of incorporating unlabeled data in the training. Possible future directions include (i) the evaluation of various methods for unsupervised feature learning such as Sparse Coding [22] and Deep Belief Networks [13], (ii) the evaluation of different combinations of teacher and student models, for example, the combination of different types of DNN either as teachers or students; the identification of suitable architectures for these roles could also be an interesting direction, and (iii) the application of outlier detection [16] approaches to filter out noisy unlabeled data.

6. REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, Xiaoqiang Zheng, and Google Brain. TensorFlow: A System for Large-Scale Machine Learning TensorFlow: A system for large-scale machine learning. In *Proc. of USENIX Symp. on Operating Systems Design and Implementation (OSDI)*, pages 265–284, 2016.

[2] Emmanouil Benetos, Simon Dixon, Dimitrios Giannoulis, Holger Kirchhoff, and Anssi Klapuri. Automatic music transcription: challenges and future directions. *Journal of Intelligent Information Systems*, jul 2013.

[3] Rachel Bittner, Justin Salamon, Mike Tierney, Matthias Mauch, Chris Cannam, and Juan Bello. MedleyDB: a multitrack dataset for annotation-intensive MIR research. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, 2014.

[4] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho. Transfer learning for music classification and regression tasks. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, 2017.

[5] Jia Cui, Brian Kingsbury, Bhuvana Ramabhadran, George Saon, Tom Sercu, Kartik Audhkhasi, Abhinav Sethy, Markus Nussbaum-Thom, and Andrew Rosenberg. Knowledge Distillation Across Ensembles of Multiplingual Models for Low-resource Languages. In *Proc. of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4825–4829, 2017.

[6] Christian Dittmar and Daniel Gärtner. Real-time transcription and separation of drum recordings based on NMF decomposition. In *Proc. of the International Conference on Digital Audio Effects (DAFx)*, pages 187–194, Erlangen, Germany, September 2014.

[7] Georgi Dzhambazov. Towards a drum transcription system aware of bar position. In *Proc. Audio Engineering Society Conference on Semantic Audio (AES)*, London, UK, Jan 2014.

[8] Derry Fitzgerald. Harmonic / Percussive Separation Using Median Filtering. In *Proc. of International Conference on Digital Audio Effects (DAFx)*, 2010.

[9] Derry FitzGerald and Jouni Paulus. Unpitched percussion transcription. In *Signal Processing Methods for Music Transcription*, pages 131–162. Springer, 2006.

[10] Nicolai Gajhede, Oliver Beck, and Hendrik Purwins. Convolutional Neural Networks with Batch Normalization for Classifying Hi-hat, Snare, and Bass Percussion Sound Samples. In *Proc. of the Audio Mostly*, pages 111–115, 2016.

[11] Olivier Gillet and Gaël Richard. Enst-drums: an extensive audio-visual database for drum signals processing. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, 2006.

[12] Olivier Gillet and Gaël Richard. Transcription and separation of drum signals from polyphonic music. *IEEE Transactions on Audio, Speech and Language Processing*, 16(3):529–540, 2008.

[13] Philippe Hamel and Douglas Eck. Learning Features from Music Audio with Deep Belief Networks. In *Proc. of International Society for Music Information Retrieval Conference (ISMIR)*, pages 339–344, 2010.

[14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *arXiv:1503.02531*, pages 1–9, 2015.

[15] Jinyu Li, Rui Zhao, Jui Ting Huang, and Yifan Gong. Learning small-size DNN with output-distribution-based criteria. In *Proc. of the Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1910–1914, 2014.

[16] Yen-Cheng Lu, Chih-Wei Wu, Chang-Tien Lu, and Alexander Lerch. Automatic outlier detection in music genre datasets. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 101–107, 2016.

[17] Jonathan Masci, Ueli Meier, Dan Cirean, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *Proc. of International Conference on Artificial Neural Networks (ICANN)*, 2011.

[18] Brian Mcfee, Eric J Humphrey, and Juan P Bello. A software framework for musical data augmentation. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 248–254, 2015.

[19] Jouni Paulus. *Signal Processing Methods for Drum Transcription and Music Structure Analysis*. PhD thesis, Tampere University of Technology, Tampere, Finland, 2009.

[20] Jouni Paulus and Anssi Klapuri. Drum sound detection in polyphonic music with hidden markov models. *EURASIP Journal on Audio, Speech, and Music Processing*, (14), 2009.

[21] Colin Raffel, Brian Mcfee, Eric J. Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, and Daniel P. W. Ellis. mir_eval: A Transparent Implementation of Common MIR Metrics. *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 367–372, 2014.

[22] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. Self-taught learning: transfer learning from unlabeled data. In *Proc. of the International Conference on Machine Learning (ICML)*, pages 759–766, 2007.

[23] Axel Roebel, Jordi Pons, Marco Liuni, and Mathieu Lagrange. On Automatic Drum Transcription Using Non-Negative Matrix Deconvolution and Itakura Saito Divergence. In *Proc. of the International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2015.

[24] Carl Southall, Ryan Stables, and Jason Hockman. Automatic Drum Transcription Using Bi-Directional Recurrent Neural Networks. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, 2016.

[25] Carl Southall, Ryan Stables, and Jason Hockman. Automatic drum transcription for polyphonic recordings using soft attention mechanisms and convolutional neural networks. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 606–612, 2017.

[26] Carl Southall, Chih-Wei Wu, Alexander Lerch, and Jason Hockman. MDB DRUMS - an annotated subset of medleydb for automatic drum transcription. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)(Late-breaking Demo)*, 2017.

[27] Vinícius M. A. Souza, Gustavo E. A. P. A. Batista, and Nilson E. Souza-Filho. Automatic classification of drum sounds with indefinite pitch. In *Proc. of the International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, Killarney, Ireland, Jul 2015.

[28] Dirk Van Steelant, Koen Tanghe, Sven Degroeve, Bernard De Baets, Marc Leman, and Jean-Pierre Martens. Support vector machines for bass and snare drum recognition. In *Classification – the Ubiquitous Challenge*, pages 616–623. Springer, 2005.

[29] Lucas Thompson, Matthias Mauch, and Simon Dixon. Drum Transcription via Classification of Bar-Level Rhythmic Patterns. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, 2014.

[30] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proc. of the International Conference on Machine Learning (ICML)*, 2008.

[31] Richard. Vogl, Matthias Dorfer, and Peter Knees. Recurrent Neural Networks for Drum Transcription. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 730–736, 2016.

[32] Richard Vogl, Matthias Dorfer, and Peter Knees. Drum Transcription From Polyphonic Music With Recurrent Neural Networks. In *Proc. of the International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 201–205, 2017.

[33] Richard Vogl, Matthias Dorfer, Gerhard Widmer, and Peter Knees. Drum Transcription Via Joint Beat and Drum Modeling Using Convolutional Recurrent Neural Networks. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 150–157, 2017.

[34] Shinji Watanabe, Takaaki Hori, Jonathan L. Roux, and John R. Hershey. Student-Teacher Network Learning with Enhanced Features. In *Proc. of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5275–5279, 2017.

[35] Chih-Wei Wu and Alexander Lerch. Drum transcription using partially fixed non-negative matrix factorization with template adaptation. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 257–263, 2015.

[36] Chih-Wei Wu and Alexander Lerch. On drum playing technique detection in polyphonic mixtures. In *Proc. of International Society for Music Information Retrieval Conference (ISMIR)*, pages 218–224, 2016.

[37] Chih-Wei Wu and Alexander Lerch. Automatic drum transcription using the student-teacher learning paradigm with unlabeled music data. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 613–620, 2017.

[38] Kazuyoshi Yoshii, Masataka Goto, and Hiroshi G. Okuno. Drum sound recognition for polyphonic audio signals by adaptation and matching of spectrogram templates with harmonic structure suppression. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(1):333–345, 2007.