IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 9, 2020, accepted May 29, 2020, date of publication June 2, 2020, date of current version June 12, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2999388

A Comprehensive Review of Polyphonic Sound Event Detection

T. K. CHAN[1,2] AND CHENG SIONG CHIN[1], (Senior Member, IEEE)
[1]Faculty of Science, Agriculture, and Engineering, Newcastle University, Singapore 599493
[2]Xylem Water Solutions Singapore Private Ltd., Singapore 609935

Corresponding author: Cheng Siong Chin (cheng.chin@newcastle.ac.uk)

ABSTRACT One of the most amazing functions of the human auditory system is the ability to detect all kinds of sound events in the environment. With the technologies and hardware advances, polyphonic Sound Event Detection (SED) can be developed to mimic the ability of the human auditory system. However, the development of a SED system is no trivial task, and several different factors often hinder accuracy. Although there are several overview papers available, most of them only provide a theoretical overview of algorithms used with little discussion. Thus, to the best of the authors' knowledge, there is no comprehensive review that covers this particular domain. Therefore, this paper aims to provide an in-depth discussion of different methodologies proposed by various authors that include the features used, detection algorithms, and their corresponding accuracy and limitations. Additional information on possible trends is also discussed that can be useful for future development works.

INDEX TERMS Deep learning, detection algorithms, Gaussian mixture model, non-negative matrix factorization, polyphonic sound event detection.

I. INTRODUCTION

The auditory system can be considered as one of the most amazing functional groups in the human body. It allows humans to detect sound and uses acoustics cues to identify and locate sounds in the environment [1], which is made possible through a fundamental skill known as the Auditory Scene Analysis (ASA). ASA can be described in two stages, which begin with the decomposition of an acoustic signal into different sensory components. Followed by combining components from similar sources into a perceptual structure that can be interpreted by higher-level processes [2]. Subsequent research efforts that attempt to replicate ASA using computational means were then termed as Computational Auditory Scene Analysis (CASA). One major research area in this domain is the SED, which may also be referred to as Acoustic Event Detection (AED).

The primary objective of a SED system is to detect the types of events present in an audio clip and return the onset and offset of the identified event. At first thought, one may think that information provided by a SED system is trivial as compared to the information contained in an image or video. However, SED system can have several advantages as compared to a camera. Firstly, sound does not require illumination and is therefore applicable in the dark. Secondly, occluded objects do not affect detection accuracy. Thirdly, some events can only be detected by a SED system such as a car horn. Finally, sound requires lesser computational resources as compared to an image or video [3], [4]. Thus, a SED system has great potential in many other domains besides speech recognition [5], [6] such as medical telemonitoring [7], surveillance [8], [9], equipment monitoring [10], [11] and wildlife monitoring [12], [13].

In general, a SED system can be categorized as a monophonic or a polyphonic SED system. As illustrated in Figure 1 and Figure 2, a monophonic SED system can only detect a single event from an audio recording, whereas a polyphonic SED system can detect any number of events within a single audio recording [14]–[16]. Naturally, a polyphonic SED system is more appropriate in a real-life application because a real-life environment is more likely to contain multiple sound sources [16]–[19]. But this would also indicate that a polyphonic SED system is much more challenging because the different sound event can coincide [15]–[17], [20], [21] and features extracted from the mixture may not match any of the features extracted from sounds in isolation [18], [19].

The associate editor coordinating the review of this manuscript and approving it for publication was Utku Kose.

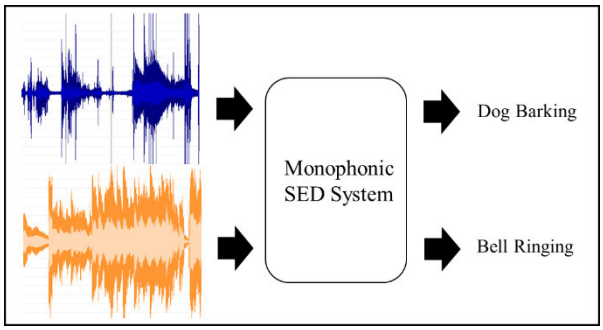


FIGURE 1. Example of a monophonic SED system.

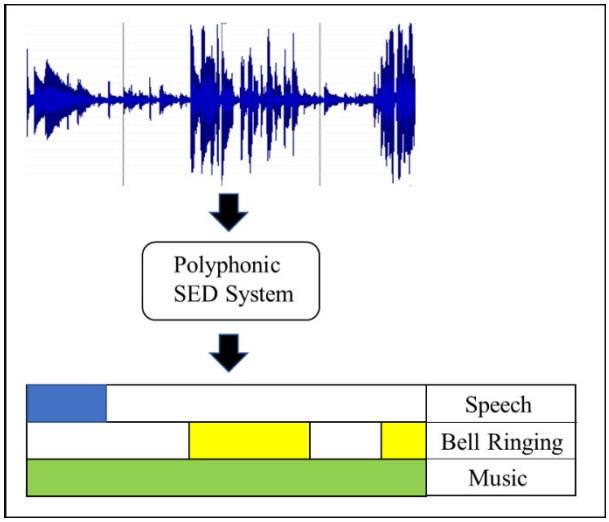


FIGURE 2. Example of a polyphonic SED system where it returns the event tag and the onset and offset highlighted in different colors.

Besides, it is not known a priori how many events can be present in a recording.

A SED system is also associated with other challenges. Firstly, each event class may be made up of several different sources; for example, a dog barking sound event can be produced by dogs of various breeds with varying characteristics of sound [14]. Secondly, the presence of background noise can complicate the identification of sound events within a particular time frame [3] and is further aggravated if the Signal-to-Noise Ratio (SNR) is low. As such, intensive research efforts have been invested in the domain of SED, which attempts to overcome the challenges faced by a SED system and improve the current state of the art. An annual competition known as the Detection and Classification of Acoustic Scenes and Events (DCASE) challenge was also launched to support the development of CASA methods by comparing them using a publicly available dataset.

Although there were discussions and reviews on the DCASE challenge submissions, they were limited to the challenge's submission with an emphasis on submission results [19], [22]–[27]. On the other hand, reviews by Dang *et al.* [20] and Xia *et al.* [28] only covered a brief theoretical aspect of several deep learning models while Bui *et al.* [29] cover Non-negative Matrix Factorization (NMF). Rex [30] provided software recommendations for SED. Although a survey by Chandrakala and Jayalakshmi [31] included discussion of several SED systems, the survey was focused on the audio scene and event classification. The survey by Purwins *et al.* [32] was mainly on the general overview of audio signal processing. Thus, most of the reviews in the literature provided a general overview of the algorithm used without explicit analysis and discussion. As such, to the best of our knowledge, there has not yet been a comprehensive review of SED.

Hence, this paper aims to fill the research gap by providing a comprehensive review of polyphonic SED since such a system is found to be more appropriate in a real-life application and deems to be more challenging as compared to a monophonic SED system. As there is a substantial literature in this area, the systems reviewed in this paper should cover the following points

- Polyphonic SED systems that are applied to real-life audio.
- Systems that can return the onset or offset of a sound event.
- Systems that achieved remarkable results in the annual IEEE AASP DCASE challenges or indexed by EI Compendex or Scopus in reputable databases such as IEEE Xplore.

Although Sound Event Detection and Localization (SEDL), as well as anomalous SED, appear to fit the criteria mentioned above, it is essential to point out that these two areas belong to different domains. SEDL is the combined task of identifying temporal activities of each sound event and the estimation of their respective spatial location trajectories when active [33]–[35].

On the other hand, anomalous SED is the detection of anomalous sounds that are difficult or even impossible to collect [36]. Thus, the detection system has to detect unknown sounds that are not given in the training label [37]. As such, this task cannot be solved as a traditional classification task. Therefore, these two areas will not be covered in this paper under the polyphonic SED system.

By focusing on these points, this paper aims to contribute to the existing literature by providing

- Review of different polyphonic SED system proposed by various authors. It includes an in-depth discussion of their methodology and limitations.
- Several future research directions that may solve existing problems in a SED system.

A SED system can be broadly classified into non-Neural Network (NN) based and NN-based methodologies. As seen in Figure 3, NN-based methods can be further classified into non-hybrid models, hybrid models, and models utilizing weakly labeled data. In this paper, a non-hybrid model refers to a model that has not been modified or stacked with another model, whereas a hybrid model refers to models that are stack

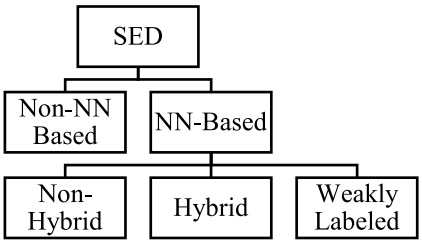


FIGURE 3. SED categories.

together to become one model. As models utilizing weakly label data generally have a big difference with their detection strategy, they are classified as a different subcategory rather than in non-hybrid or hybrid models subcategory.

Thus this paper proposed to discuss these groups of methodologies in the same manner. The discussion of methods in this section will be done in the following order, the non-NN based methods followed by the NN based methods. Therefore, the paper is organized as follows: Section II provides an in-depth discussion of different non-NN based methodologies. Section III presents a detailed discussion of different NN based methods. Section IV presents several public datasets and commonly used evaluation metrics, and Section V provides a discussion on possible future research directions. Finally, the paper ends with a conclusion.

II. NON-NEURAL NETWORK-BASED METHODOLOGIES

In this section, two types of non-NN based methodologies that were commonly used by different authors are discussed in detail. After the discussion of each method, a summary is provided to conclude this section.

A. GMM-HMM

Perhaps one of the earliest states of art for Automatic Speech Recognition (ASR) is the Hidden Markov Model [38], [39], which is a statistical model where each state is not directly observable (i.e., hidden) [40], [41]. With the use of the Expectation-Maximization (EM) algorithm, each state can be categorized by Gaussian Mixture Model (GMM) that models the observation (acoustic vectors) corresponding to the state [40], [42]. GMMs can have several benefits that make them a suitable candidate for this role. Firstly, with enough components, they can model probability distributions to any required level of accuracy, and they are relatively easy to fit data using the EM algorithm [42]. With success in ASR, GMM-HMM becomes a promising and natural choice in modeling other sound events [43], and one such example was by Mesaros *et al.* [44].

The architecture was tested on a large scale audio database that consists of 61 event classes to investigate the effectiveness of GMM-HMM in the real world. In the experiment, Mesaros *et al.* [44] proposed using Mel-Frequency Cepstral Coefficients (MFCCs), delta MFCCs, delta-delta MFCCs as the input features to train a three-state left-to-right HMM with 16 Gaussians per state for each event class. These models were then connected into a network HMM having equal transition probabilities from one event model to another. This would allow an unrestricted sequence of 61 models where any model can follow any other with no limits to the number of events that can be detected in the audio. With the use of the Viterbi algorithm, an optimal sequence of events can then be decoded. Thus, it allows the detection of the most prominent event at each given polyphonic segment together with its timestamp.

However, this system only achieved an F1-score of 30.1% but with an 84.1% Error Rate (ER). Despite the inclusion of event frequency of occurrence as the prior knowledge, the accuracy of this system was not improved. Mesaros *et al.* [44] explained that this might be due to the adding of events from different environments, which averages out the differences in count between events specific to a particular environment. Although there was no limit to the number of events that can be detected in audio, such a system only allows the detection of the most prominent event in each segment. This does not reflect real life-scenario where sound events can coincide. Moreover, the majority of the errors can be caused by temporally overlapping sound events [45].

To improve detection accuracy and allow polyphonic SED on the same dataset, Heittola *et al.* [21] proposed a two stages detection methodology, as illustrated in Figure 4. Heittola *et al.* [21] suggested to first determine the context of the audio before detecting the events in the audio. Context, in this case, refers to the audio background or environment. As explained by Heittola *et al.* [21], this can reduce the search space for sound events as context information can provide rules for selecting a specific set of events.

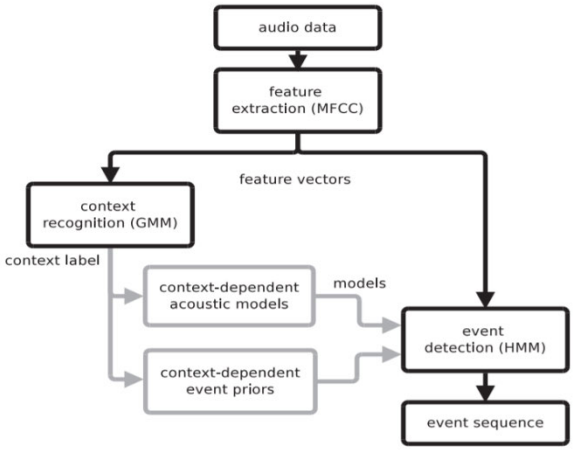


FIGURE 4. Overview of proposed methodology by Heittola *et al.* [21].

In the training stage, each context recognition system was trained using GMM with MFCCs extracted from the different contexts where each GMM consists of 32 gaussian distributions. Whereas a SED system was trained using an HMM with MFCCs, delta MFCCs, and delta-delta MFCCs extracted from sound events belonging to different contexts. Similar to [44], each event class was modeled by an individual three-state left-to-right HMM with 16 Gaussians per state.

Frames with overlapping events were not discarded for training and instead used as training samples for the respectively sound events. Heittola *et al.* [21] suggested that the variability caused by the overlapping sound events classes will average out, and the model will learn a reliable representation of the target sound events. The consecutive passes of the Viterbi algorithm were proposed to allow polyphonic SED, where each pass must be different from the previously decoded one. This could prevent the detection of a similar event, and the number of consecutive passes was fixed at 4.

In the testing stage, test audio was first segmented into 4 seconds segment and was classified by each context recognition model. Context label was then given based on the highest total log-likelihood accumulated over the audio. Based on this context label, the sound present in the audio were then detected using the HMM models trained.

Together with event priors, such a system can achieve a single second segment based F1-score of 19.5 and 30-seconds segment based F1-score of 29.4. However, such a system was not able to win a monophonic HMM-GMM system by a considerable margin and instead performed slightly worse in terms of 30 seconds segment based F1-score. The drawback of this system is the dependency of the context recognition accuracy. A wrongly recognized context can lead to wrong SED model selection and event priors. This complication was not fully reflected in their experiment as different contexts contain similar sound events, and some of the common events were correctly recognized. Also, the number of Viterbi passes was estimated based on average polyphony of the recorded materials, but the number of events in test data is usually not known a priori.

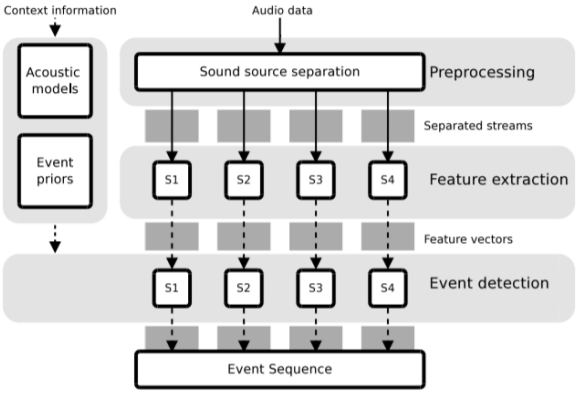


FIGURE 5. Overview of proposed methodology by Heittola *et al.* [46].

Heittola *et al.* [46] then extended his work in [21] by including source separation before the detection of sound events, which is illustrated in Figure 5. However, in this work, the ground truth of the context was known with certainty and given to the SED system, which eliminates the need to train a context recognition system.

In the training stage, which began with the source separation, NMF was utilized to decompose the spectrum into four different components where the number of components was based on the average amount of overlapping events in the evaluation dataset. Since each component may contain one or more sound sources, Heittola *et al.* [46] proposed using Wiener filter on each component to allow the separation of the stream, which contains roughly homogenous spectral content and differs significantly from the others. As the source separation was unsupervised, there was no knowledge of which event was separated into which stream. Thus, Heittola *et al.* [46] suggested two approaches to select the stream that contains the target event using EM algorithm. The first approach was to choose the most prominent stream based on the highest likelihood, while the second approach was to perform iterative elimination of stream that was least possible to contain the target event. However, it was found that results from both schemes were comparable, but stream elimination scheme had the benefit of faster convergence and being straight forward. Separated streams from their respective event class then have features such as MFCCs, delta MFCCs and delta-delta MFCCs extracted where they were used to model a three-state left-to-right HMM with 16 Gaussians per state. In the testing stage, the SED system was applied separately for each stream, and results from each stream were combined into a single set of events.

Based on the proposed methodology and using stream elimination scheme, Heittola *et al.* [46] reported a single second segment based F1-score of 44.9 and average 30 seconds segment F1-score of 60.8, which was approximately doubled of their previous work [46]. Although such a proposal can provide much higher accuracy, it requires the context of testing audio to be known a priori, which may limit its applicability. Similarly, the number of components to be separated is based on the average number of overlapping events in the evaluation dataset and thus cannot be known beforehand. Also, there may be a risk of selecting the wrong stream for event training, even though such a mistake was not mentioned in the text.

Although GMM-HMM had seen its success in ASR, it does not seem to be very effective and accurate for SED. This could be due to the fact that GMM has a serious shortcoming where they cannot effectively exploit information embedded in a large window of frames [42]. On the other hand, the use of MFCCs may not be a suitable input for the model. As suggested by Cakir *et al.* [62], there is a loss of information during the calculation of MFCCs as MFCCs are made up of the first few coefficients after the application of Discrete Cosine Transform. Moreover, the sum of MFCCS of sound sources is not the same as the sum of MFCCs of the mixtures of these sources. Another major drawback of using GMM-HMM is the need to train a large number of models to represent each event class.

B. NMF

Another popular method for SED is the use of NMF. As discussed earlier, NMF was used by Heittola *et al.* [46] as an unsupervised approach for audio source separation. However, NMF can also be used as a supervised method for SED.

NMF is a method popularized by Lee and Seung [47], where the objective is to decompose a non-negative matrix of size $L \times N$ into two non-negative matrices W and H of size $L \times K$ and $K \times N$, respectively. W can be interpreted as the dictionary matrix and H can be interpreted as the activation matrix while K represents the number of components. Thus, the conventional supervised training of an NMF is to extract W from isolated events to form a dictionary [29], [48]. This dictionary would then be applied to the test data to derive H, and post-processing of H is then carried out to identify the activation of events at each time frame. Although such a method is easy to apply, it has difficulty detecting overlapping sounds [48]. Smaragdis and Brown [48] suggested that this problem can be overcome by modeling overlapping sound as an additional class, but the combination of the event would make the training intractable.

Thus, several proposals improved on the conventional NMF method for SED. One of them was the use of coupled NMF by Mesaros *et al.* [49]. As seen in Figure 6, the idea of a coupled NMF is to jointly learn two non-negative dictionary matrices W_1 and W_2 with a common H using two non-negative matrices as input instead of one.

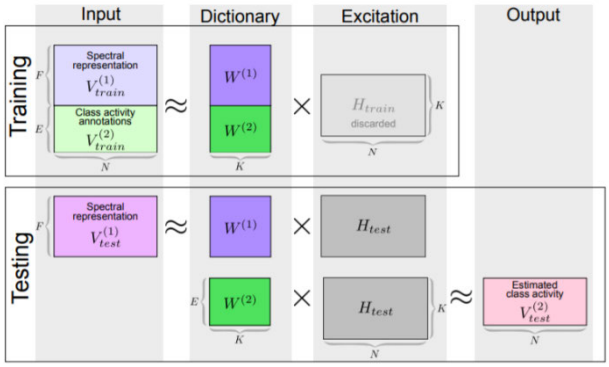


FIGURE 6. Proposed coupled NMF by Mesaros *et al.* [49].

In their experiment, the two-non negative matrices provided were 1) audio frequency spectrum and 2) a frame-level one hot encoding matrix to represent the presence of events based on the annotation of audio. As the size of W_1 is relative to the size of training data, this may not be computationally feasible. Thus, to increase the computation efficiency, Mesaros *et al.* [49] suggested reducing the size of the dictionary through clustering of components and only allowing the centroid of clusters to form the dictionary.

The signal was first decomposed to detect the presence of a sound event in test audio, using NMF with W_1 to obtain H_{test}. H_{test} was then used to reconstruct the frame-level one hot encoding matrix using W_2. As the matrix obtained was not in binary form, it required further processing. The values above the mean of the matrix were set to 1 to indicate the detection of a sound event in that specific time frame. On the other hand, values below the mean were set to 0. To prevent noise and outliers, detected sound events that did not correspond to a total of 200ms duration were also set to 0.

Based on this setup, Mesaros *et al.* [49] achieved average a single second segment based F1 score of 57.8 on TUT-SED 2009. The most interesting result was that the use of cluster centroids as new components could achieve a higher F1-score as compared to using the entire dictionary. Although, using cluster centroids can reduce the computation cost, but the drawback is that there is a need to perform clustering for every addition of new training data. In addition, the optimal cluster number has to be derived through several trials. Finally, the use of mean as a threshold may not be the best value to achieve maximal performance.

On the other hand, Bisot *et al.* [50] proposed a methodology to learn a classifier and the NMF in a joint optimization problem that they referred to as Nonnegative task-driven dictionary learning (TD-NMF).

In essence, this methodology learns an optimized dictionary through NMF with beta divergence by minimizing the classification cost of a regularized linear logistic regression. As explained by Bisot *et al.* [50], this would allow a more discriminative dictionary of spectral templates to be learned. After the dictionary was learned, the test data was projected onto the learned dictionary. The projection from the test data was then used as a feature for classification. To allow multi-label classification, the class probabilities of each test frame were thresholded by a value which was dependent on the acoustic scene (0.3 for the home environment and 0.35 for the residential area). Thus, overlapping events can be detected as long as the probabilities in a given frame exceed the fixed threshold.

Based on this methodology, Bisot *et al.* [50] reported a single second segment based F1-Score of 49.5 with an ER of 69.5 when test on the TUT-SED 2016 development dataset. While this methodology was able to achieve the lowest ER as compared to methods such as GMM, Random Forest (RF), Gated RNN, RNN, and NMF, it was not able to achieve the highest F1-score (RNN made a slightly higher F1-score of 49.8). Although such methodology shows competitive results, it was only tested on a small dataset with a total duration of approximately 80 minutes. Moreover, the number of overlapping events is also minimal. Finally, there is a need to fine-tune the threshold to achieve maximal performance.

Another interesting proposal was by Ohishi *et al.* [51], who suggested the integration of NMF with Bayesian nonparametric for polyphonic SED. A Bayesian nonparametric model is a Bayesian model on an infinite-dimensional parameter space [52]. Thus, the Bayesian nonparametric approach allows a model to adapt its complexity according to the data without the selection of model parameters in advance (i.e., the number of components for NMF or the number of events that should be in an audio clip) [53]. Such a combination would then allow the assumption of possibly an infinite number of events in an audio clip.

Ohishi *et al.* [51] modeled the overlapping sound event using NMF, where Markov Indian Buffet Process (mIBP) and Chinese Restaurant Process (CRP) were integrated and proposed the Bayesian logistic regression to estimate the

audio event labels from the activation matrix. This technique was subsequently tested on an English learning podcast, and the authors [51] reported an accuracy (in terms of Area Under Curve (AUC)) of 0.79 winning a baseline GMM and three other variants of proposed methods.

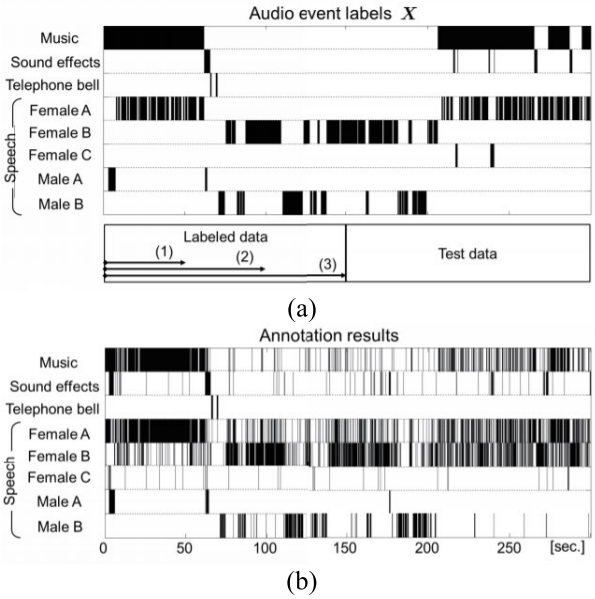


FIGURE 7. Ground Truth (a) and Predicted Labels (b) [51].

However, the performance of this methodology appears to be inconclusive and biased. Firstly, it was only tested on an English learning podcast with a duration of 150s. Secondly, this podcast was prepared in a controlled environment [54]. Thirdly, the GMM may not be effective using only 100s of training data, and thus, the comparison does not appear to be fair. Fourthly, as seen from Fig. 7, there was no training data for Female C, it is unclear how the classification of Female C was achieved. The new event detected should not be assumed to be Female C, and accuracy for Female C should be 0. If that was the case, the accuracy of this proposed model is lower than the baseline GMM. In addition, using AUC as the accuracy metric is not appropriate, as seen in the following figure, almost the entire block of the testing signal was annotated as Female A and Female B, but the AUC calculated was 0.769 and 0.744 respectively. Lastly, it was mentioned by the authors [51] that the Poisson likelihood model used in their study suffers from theoretical issues.

C. SUMMARY OF NON-NEURAL NETWORK-BASED METHODOLOGIES

The previous subsections showcased the different non-NN architectures proposed by various authors, and discussion on their methodologies and limitations were done.

In this section, details on the features used, architectures, performances, and limitations are summarized in the following tables. Table 1 provides the information on features used and their respective processing method. For non-NN based methodologies, there is a wide variety of features used with MFCC forming the majority. However, as mentioned earlier, the use of MFCC may result in the loss of information. Table 2 provides the information on the architectures used and their respective limitations. Among the different methodologies, some of them utilized the context information to provide rules for selecting a specific set of events. However, detection accuracy is highly dependent on the context recognition accuracy, which may limit such strategy. Table 3 then showcase the results of their proposed methodology on different types of datasets. Although non-NN based methods may not require a large amount of strongly labeled data, as seen in Table 3, they do not perform very well.

III. NEURAL NETWORK-BASED METHODOLOGIES

In this section, NN-based methodologies are discussed in detail. The section first began with the discussion of non-hybrid NN methods followed by hybrid NN methods and methods that utilized the weakly labeled data. A summary section is provided at the end of each subsection to conclude the findings of each method.

A. NON-HYBRID MODELS

With the advances in machine learning algorithms and computer hardware, Deep Neural Network (DNN) that contains many layers of linear units, and a large output layer can be trained efficiently [42]. This allows DNN to jointly learn much better feature representations and appropriate classifiers [55] where successes can be seen in sound-related domains such as ASR [42], [56], [57], sound event or environmental classification [58], [59], source separation [60], [61].

Thus, making DNN a viable and attractive choice for SED. Moreover, their architecture allows multi-label classification directly. In contrast, model-based HMMs require additional effort to train individual models for each class or allowing multiple passing of the Viterbi algorithm for polyphonic SED [25].

However, conventional NN-based methodologies only allow single-label classification. Thus the easiest and the most common way to accommodate multi-label classification is to threshold the event class probabilities given by the NNs. This would allow multiple sound events to be detected as long as the probabilities are above the predefined threshold.

Driven by the pros of NN, Cakir *et al.* [62] proposed using an FNN with maxout activation function for polyphonic SED. (Note: Such network may also be referred to as a Maxout Networks [63]). The idea of a maxout unit is to facilitate optimization by dropout and improve the accuracy of dropout's fast approximate model averaging technique [63]. Given an input, x, a maxout unit implements the following function [63],

$$h(x) = \max(xW + b) \tag{1}$$

where W and b are learned parameters. This function simply retrieves the max of input and can be interpreted as making a

TABLE 1. Features used in non-NN based methodologies.

References	Features	Parameters used for Spectrum Calculation				Additional Information
		Window	Window Length (ms)	Overlap (%)	Mel-Filter Banks	
[44]	• MFCCs • Delta MFCCs • Delta-delta MFCCs	Hamming	20	50	40	-
[21]	• MFCCs • Delta MFCCs • Delta-delta MFCCs	-	20	50	40	Frequency range to compute MFCCs is within 30Hz to 22050Hz.
[46]	• MFCCs • Delta MFCCs • Delta-delta MFCCs	-	20	50	-	-
[49]	• Spectrogram	-	100	50	-	1024 FFT bins
[50]	• Mel Spectrum	-	40	50	40	Min-max normalization before STFT
[51]	• Mel Spectrum	Hanning	100	-	25	-

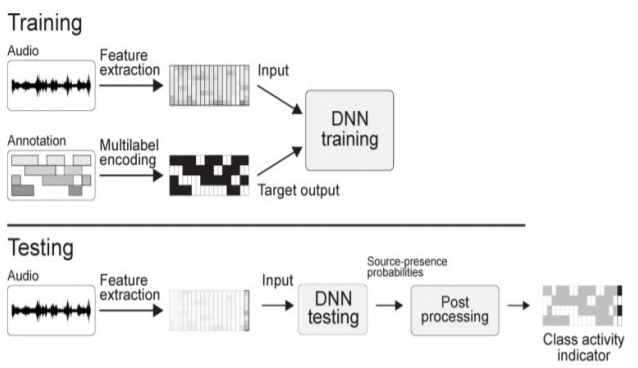


FIGURE 8. Proposed Methodology by Cakir *et al.* [62].

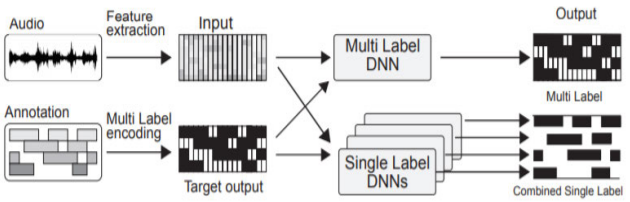


FIGURE 9. Difference between multi-label and combined single label DNN [66].

piecewise linear approximation to an arbitrary convex function [63]. As explained by Cakir *et al.* [62], such function is not bounded, easy to optimize, and does not suffer from vanishing gradients problem. Most importantly, maxout function shows superior results as compared to sigmoid function in speech-related task [64], [65].

To allow the modeling of the dynamic properties of sounds, Cakir *et al.* [62] proposed the use of context windowing (window length of 5) where the audio frame of extracted feature vectors was concatenated with adjacent time frames to form a single training instance. In addition, the output from the output layer was smoothed by a median filter to remove noise. Such a framework is illustrated in Figure 8.

Using two layers FNN, Cakir *et al.* [62] reported a single second segment based F1-score of 63.8. Based on the results using different input features, Cakir *et al.* [62] also concluded that using log-mel band energies as a feature was much better than using MFCC and mel-band energies.

Cakir *et al.* [66] then extended his work by decomposing the multi-label classifier into an ensemble of single-label classifiers. Cakir *et al.* [66] explained that such a classifier has the benefit of allowing dynamic inclusion of new labels by training classifiers only for the new sound events instead of retraining the entire FNN. To enable polyphonic SED, the test audio must be classified by every single model. The difference between the two methodologies can be seen in Figure 9.

Based on such a scheme, Cakir *et al.* [66] reported a single second segment based F1 score of 61.9% which was slightly lower than a multi-label DNN accuracy.

Although there is no need to retrain the entire model with the inclusion of new events, a single label classifier requires training of multiple classifiers that may need to be tuned separated to achieve maximal performance. In addition, the testing time for such architecture maybe a few folds higher than a multi-label classifier.

Results have shown that both architectures achieved much higher accuracy than the non-NN based methodologies, but they are not without any drawbacks. Firstly, using maxout function doubles the number of parameters for every single unit, which can lead to a high parameter number [67]. In addition, maxout network is prone to overfitting [68]. This is because the max function propagates the gradient only to the unit that gave the largest activation, so the remaining units do not get updated [69]. An FNN with maxout function may not

TABLE 2. Proposed architectures and their respective limitations.

References	Proposed Algorithm For SED	Additional Information	Individual Limitations
[44]	GMM-HMM	• EM algorithm for training. • Viterbi algorithm for optimal sequence decoding. • Each event class model represented by three state left to right HMMs with 16 Gaussians per state.	• Each event class has to be modeled by an individual HMM. • GMM cannot effectively exploit information embedded in large window of frames • MFCC may not be a good feature • Requires a large number of model to be trained to represent each event class • Low accuracy with large error rate • It can only detect the most prominent event at a time.
[21]	GMM-HMM	• GMM for context recognition. • EM algorithm for training. • Viterbi algorithm for optimal sequence decoding. • Each event class model represented by three state left to right HMMs with 16 Gaussians per state.	• Each event class has to be modeled by an individual HMM. • GMM cannot effectively exploit information embedded in large window of frames • MFCC may not be a good feature • Requires a large number of model to be trained to represent each event class • Accuracy of this system did not win a monophonic GMM-HMM system by a huge margin • Accuracy is dependent on the context recognition accuracy. • Appropriate number of Viterbi passes was estimated based on data.
[46]	GMM-HMM	• NMF for source separation. • EM algorithm for training. • Viterbi algorithm for optimal sequence decoding. • Each event class model represented by three state left to right HMMs with 16 Gaussians per state.	• Each event class has to be modeled by an individual HMM. • GMM cannot effectively exploit information embedded in a large window of frames • MFCC may not be a good feature • Requires a large number of model to be trained to represent each event class • Context of testing audio must be known a priori. • Appropriate number of components in each test stream is estimated based on evaluation dataset. • May have risk of selecting the wrong stream for event training.
[49]	Coupled NMF	• NMF for matrix decomposition. • K-means clustering for components clustering.	• Need to perform clustering for every addition of new training data. • Optimal cluster number need to be derive through several trials. • Requires careful tuning of threshold value.
[50]	NMF-Logistic Regression	• NMF for matrix decomposition. • Multinomial logistic regression for event detection.	• Training data cannot contain overlapping events. • Requires careful tuning of threshold value.
[51]	NMF-Bayesian Nonparametric	• NMF with mIBP and CRP for modeling overlapping audio events. • Bayesian logistic regression for event annotations.	• Only tested on a small dataset collected in a controlled environment. • Comparison and results are not conclusive and appear bias. • Poisson likelihood model used in the study suffers from theoretical issues.

TABLE 3. Accuracy of non-NN based methodologies.

References	Dataset	Segment Based F1-Score (%)	Other Metric
[44]	TUT-SED 2009		F1-Score: 30.1% ER: 84.1%
[21]	TUT-SED 2009	1 Second: 19.5 30 Seconds: 29.4	
[46]	TUT-SED 2009	1 Second: 44.9 30 Seconds: 60.8	
[49]	TUT-SED 2009	1 Second: 57.8	
[50]	TUT-SED 2016 Development Dataset	1 Second: 49.5	1 Second Segment Based ER: 69.5
[51]	English Learning Podcast		AUC: 0.79

perform as well as a Convolutional Network Network (CNN) with maxout function based on the result shown in [70].

Parascandolo *et al.* [18] then proposed Bidirectional Long Short Term Memory (BLSTM) for polyphonic SED. Bidirectional LSTM allows the processing of data in both directions by utilizing two separate hidden layers. They are subsequently concatenated and fed to the same output layer [71]. It allows access to long-range context in both input directions, which can helps in classification and avoid tailored postprocessing or smoothing step, thus making it a suitable choice for SED [18].

The audio recordings were normalized to a scale of -1 to 1 before the calculation of mel energies to account for audio recorded in different conditions. As an additional measure to reduce overfitting, Parascandolo *et al.* [18] increased the dataset by 16 times using several data augmentation techniques such as time-stretching, sub-frame time-shifting, and blocks mixing. Parascandolo *et al.* [18] also proposed the addition of Gaussian noise to the network weights as it was found to 'simplify' RNN by reducing the amount of information required to transmit the parameters which improve generalization [72].

In their paper, Parascandolo *et al.* [18] compared this architecture with an FNN, vanilla LSTM and a BLSTM without data augmentation on TUT-SED 2009. The final results showed that their method with data augmentation was the best performing architecture with a single frame segment based F1-score of 64.7 and a single second segment based F1-score of 65.5. However, it was only marginally better than a BLSTM without data augmentation, which achieved a single frame segment F1-score of 64.0 and a single second F1-score of 64.6. This shows that simple data augmentation may not be useful, given the considerable increase in computational cost and negligible impact on model accuracy.

As mentioned earlier, Parascandolo *et al.* [18] did not apply any post-processing, as it was hypothesized that output from an RNN was already smoothed. However, empirical results shown by Hayashi *et al.* [73] proved that post-processing is still necessary, especially for event-based evaluation. Although the addition of Gaussian noise can improve generalization, it was also found that such optimization can increase the training time and can affect the performance of an LSTM [74]. In addition, the LSTM has a relatively high model complexity, and parameter tuning for LSTMs is not always simple [75], [76]. Finally, the sequential nature of LSTM prohibits parallelization [77].

Adavanne *et al.* [78] also proposed the use of LSTM for SED but with additional input features such as the pitch and its periodicity as well as the Time Difference of Arrival (TDOA) in sub-bands. Pitch and periodicity were estimated using Librosa implementation of pitch tracking on thresholded parabolically interpolated STFT. On the other hand, TDOA was calculated using the generalized cross-correlation with phase-based weighting (GCC-PHAT). A median filter was then applied to the estimated TDOA to remove noise.

Similar to [18], block mixing was also applied to increase the training data as a measure to reduce overfitting. Besides comparing this architecture with a GMM, Adavanne *et al.* [78] also examined the performance difference between using mono-channel and stereo-channel features. Based on the ER, Adavanne *et al.* [78] concluded that LSTM trained using log mel band energies and TDOA from the stereo channel was the best classifier with an accuracy of single second segment based F1-score of 35.4 with ER of 0.91. If the results were based on the highest F1-score, LSTM trained using mel energies and pitch would be the best classifier which achieved a single second segment based F1-score of 35.7 with a single second Error Rate (ER) of 0.92.

Although the results shown features extracted from the stereo channel are beneficial, but additional features such as pitch and TDOA did not seem to provide many benefits. It is because a LSTM trained using only log mel energies calculated from the stereo channel can already achieve a single second segment F1-score of 35.6 with ER of 0.93. As Adavanne *et al.* [78] also participated in the DCASE 2016 challenge, the only system that won the baseline system was trained using only mel energies from both channels and not with the proposed additional features. Besides the redundancy of extra features, LSTM has a high model complexity, not easy to tune and prohibit parallelization [75]–[77].

On the other hand, Xia *et al.* [79] proposed a regression-based Convolutional Neural Network (CNN) for SED. As explained by Xia *et al.* [79], multi-label classification using frame-wise labeling may not be accurate due to the annotation errors caused by humans. Thus, Greff *et al.* [74] proposed the soft labeling of events in each recording based on a confidence measure.

Xia *et al.* [79] suggested using a parabolic function to estimate the confidence measure where the peak of the parabola was positioned at the center frame of the manually labeled event (i.e., the center frame has the highest confidence). It would then allow a continuous representation for each acoustic event and training outputs would be real-valued variables instead of discrete-valued labels.

Based on such architecture, Xia *et al.* [79] achieved a segment F1-score of 61.02 with an ER of 0.63 on TUT-SED 2017 development dataset and a segment F1-score of 45.3 with an ER of 0.84 on evaluation dataset. However, such results can at best ranked fourth in terms of ER in the DCASE 2017 task 3 challenge. The poor outcome may be due to two reasons. Firstly, the small number of layers and filters may not be sufficient to learn the complex structure of polyphonic music. Secondly, CNN is unable to extract long temporal context information [14].

Although such a network can mitigate the wrongly annotated events by using a confidence measure, hyperparameter for the parabolic function may require careful tuning to achieve maximal performance. Secondly, using a parabola as a confidence function may not be appropriate because events with continuous output without too much of a fade in fade out effect such as vacuum cleaner or blender should have the maximal confidence throughout most of the annotated frame instead of just the center frame.

Vesperini *et al.* [80] proposed Capsule Neural Network (CapsNet) for polyphonic SED. The introduction of CapsNet is to overcome some limitations of CNN, in particular, the loss of information due to max-pooling operator [81].

A capsule can be thought of as a group of neurons whose output represent different properties of the same entities [82]. A CapsNet starts with a convolutional layer for feature extraction while the rest of the layers are capsule layers starting with a primary capsule layer and ending with a class capsule layer. A primary capsule layer represents the lowest level of multi-dimensional entities and contains reshaping and squashing functions. Outputs are then passed to the class capsule layer that has one capsule per output class through a dynamic routing procedure. Event probabilities were then obtained by computing the Euclidean norm of the output of each capsule.

Using CapsNet, Vesperini *et al.* [80] achieved an ER of 0.36 on TUT-SED 2016 and TUT-SED 2017 development dataset using a binaural spectrogram as the input. On the other hand, results for TUT-SED 2017 evaluation dataset show that CapsNet using log mel energies achieved the lowest ER of 0.58 instead of using a binaural spectrogram as input. Regardless of the inputs, CapsNet remains the best classifier (in terms of ER) as compared to CNN, CRNN, and GMM. However, the results did not include the F1-score. Therefore, it is unclear how well such a classifier performs in terms of F1-score.

The drawback for CapsNet is that even for simple architecture, training CapsNet requires significant computational resources [83], and training time can be much longer as compared to a CNN [84]. As mentioned by Vesperini *et al.* [80], there can be a significant performance drop if hyperparameters are not tuned correctly. Moreover, CapsNet can have a larger fluctuation in performance during training and if the fixed number of training epochs is suboptimal, CapsNet can be more prone to significant errors as compared to CNNs. In addition, CapsNet also appears to shows lesser generalization ability as compared to CNNs [80]. Finally, AdaDelta which was used as the gradient optimizer in their architecture takes a longer time to converge as compared to Adam due to its iterative operation [85], [86].

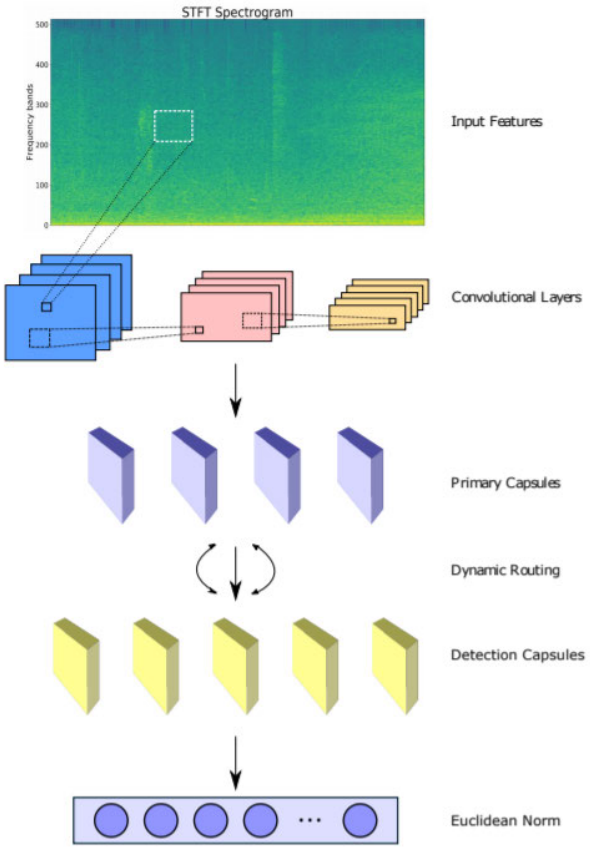


FIGURE 10. Flowchart of SED using CapsNet [80].

B. SUMMARY OF NON-HYBRID MODELS

The previous section showcased the different NN architectures proposed by various authors, and discussion on their limitations was done. In this section, details on the features used, architectures, accuracy, and limitations are summarized in the following tables.

Table 4 provides the information on features used and their respective processing method. It is evident that mel energies are the most popular input features and there is not much difference in the way they are calculated. The MFCC could cause loss of information and empirical results shown in [62] highlighted that using mel energies as a feature can increase the detection accuracy. The key difference in the input lies in

TABLE 4. Features used by different authors.

References	Features	Parameters used for Spectrum Calculation			Additional Information
		Window Length (ms)	Overlap (%)	Mel-Filter Banks	
[62]	• Log mel energies	50	50	40	-
[66]	• Mel energies	50	50	40	• Min-max normalization before STFT
[18]	• Log mel energies	50	50	40	• Min-max normalization before STFT • Z-score normalization of energy band • Data augmentation using time-stretching, subframe time-shifting and block mixing
[78]	• Log mel energies • Pitch and its periodicity • TDOA	50	50	40	• Z-score normalization of feature vectors • Data augmentation using block mixing • Periodicity extracted in 100Hz-4000Hz • TDOA calculated using a window length of 120ms, 240ms, 480ms with 20ms hop • Post-processed of TDOA with Median filter (kernel of length three)
[79]	• Log mel energies	-	-	-	-
[80]	• Log mel energies	40	50	40	• Min-max normalization before STFT

TABLE 5. Accuracy of non-hybrid NN methodologies.

References	Dataset	Segment Based F1-Score (%)	Segment Based ER	Other Metric
[62]	TUT-SED 2009	1 Second: 63.8		
[66]	TUT-SED 2009	1 Second: 61.9		Hamming Loss: ~ 0.0325
[18]	TUT-SED 2009	1 Frame: 64.7 1 Second: 65.5		
[78]	TUT-SED 2016 Development Dataset	1 Second: 35.4	1 Second: 0.91	
	TUT-SED 2016 Evaluation Dataset	1 Second: 47.8	1 Second: 0.80	
[79]	TUT-SED 2017 Development Dataset	0.1 Second: 61.02	0.1 Second: 0.63	
	TUT-SED 2017 Evaluation Dataset	0.1 Second: 45.3	0.1 Second: 0.84	
[80]	TUT-SED 2016 Development Dataset		1 Second: 0.36	
	TUT-SED 2016 Evaluation Dataset		1 Second: 0.69	
	TUT-SED 2017 Development Dataset		1 Second: 0.36	
	TUT-SED 2017 Evaluation Dataset		1 Second: 0.58	

the post-processing, where several authors proposed normalizing the energy bands.

Table 5 presents the results of their proposed methodology tested on different datasets. Although results generally have shown that NN-based methodologies can perform better than a non-NN based methodologies, there is still a large room for improvement that can be made.

Table 6 provides information on the architectures used. The different columns showcase what the key configurations used for SED are. In the Additional Information column, it provides additional details that the authors proposed in their methodology. In the table, three aspects may seem confusing. Firstly, there is a column named as No of Hidden Layers, {No of Units / Filters} with value given in the following format, 2, {800, 800}. It means that there are 2 hidden layers with 800 units in the first layer and 800 units in the second layer. Secondly, in the Additional Information column, there is an entry known as the Early stop criterion. This is the number of epochs that the authors used to evaluate if there is any further performance improvement. Thus, for the Early stop criterion: 20 epochs, this actually means that the author proposed to stop the training of the model if accuracy did not increase or did not have a significant increase after 20 epochs. Thirdly, for Filter size or Pooling size, it refers to the size of the filter and pooling operator respectively. For example, if Pooling size: (1, 3) means the pooling operator is using a pooling size of 1 by 3 and if Pooling size: (1, 4), (1, 3), (1, 2) means the pooling operator has a pooling size of 1 by 4 in the first layer and 1 by 3 in the second layer and 1 by 2 in the third.

Finally, Table 7 summarized the different limitations of each proposed methodology. It is important to point out that

TABLE 6. Proposed architecture by different authors.

Ref	Proposed Algorithm	No of Hidden Layers, {No of Units / Filters}	Activation Function	Loss Function	Gradient Descent Optimizer	Additional Information
[62]	FNN	2, {800, 800}	• Maxout at hidden layer • Sigmoid for output layer	KL Divergence	SGD	• Initial weight range: +-0.001 • Learning rate: 0.02 • Mini-batch: 50 • Pooling operator: Max • Pooling size: (1, 3) • Context window: 5 frames • Median filter: 10 frames window
[66]	FNN	2, {400, 400}	• Maxout at hidden layer • Sigmoid for output layer	BCE	SGD	• Learning rate: 0.02 • Context window: 2 frames • Median filter: 10 frames window
[18]	BLSTM	4, {200, 200, 200, 200}	• Hyperbolic tangent for memory cell • Sigmoid for input, forget, output gates and output layer	RMSE	RMS-Prop	• Initial weight range: +-0.1 • Learning rate: 0.005 • Decay rate: 0.9 • Gaussian input noise: 0.2 • Early stop criterion: 20 epochs
[78]	LSTM	2, {32, 32}	• Sigmoid for output layer	BCE	Adam	• Early stop criterion: 100 epochs
[79]	CNN	2, {16, 32}	• ReLu for convolution layers • Sigmoid for output layer	MSE	Adam	• CNN filter size: (3, 3) • Batch normalization after each convolution layer • Pooling operator: Max • Pooling applied after every convolution layer
[80]	CapsNet for TUT-SED 2016 Home	3, {32,32,8}	• ReLu for convolution layers and capsule layer • Sigmoid for output layer	Margin Loss	AdaDelta	• CNN filter size: (6, 6) • Pooling Operator: Max • Pooling size: (1, 4), (1, 3), (1, 2) • Primary capsule convolution capsule: 8 • Primary capsule filter size: (4 , 4) • Routing iteration: 3 • Decay rate: 0.95 • Dropout rate: 0.2 • Mini-batch: 20 • Early stop criterion: 20
	CapsNet for TUT-SED 2016 Residential and TUT-SED 2017	4, {4, 16, 32, 4}				• CNN filter size: (4 , 4) • Pooling Operator: Max • Pooling size: (1, 2), (1, 2), (1, 2), (1,2) • Pooling applied after every convolution layer • Primary capsule convolution capsule: 7 • Primary capsule filter size: 3 by 3 • Routing iteration: 4 • Decay rate: 0.95 • Dropout rate: 0.2 • Mini-batch: 20 • Early stop criterion: 20

all the NN-based methodologies require a large amount of strongly labeled training data to learn the mapping between features and event class.

Another common limitation is the use of a global threshold on the output layer (with 0.5 as the most commonly used value). However, this threshold value may need to be tuned to

TABLE 7. Limitations of proposed non-hybrid NN methodologies.

References	Common Limitations	Individual Limitations
[62]	• Requires large amount of strongly label training data. • Requires careful tuning of threshold value to indicate presence of event.	• Maxout function doubles the number of parameters for every single unit. • Maxout network is prone to overfitting. • FNN with maxout function may not perform as well as a CNN CNN with maxout function.
[66]		• Single label classifier may need to be tuned separately to achieve maximal performance. • The testing time may be a few folds higher than a multi-label classifier. • Maxout function doubles the number of parameters for every single unit. • Maxout network is prone to overfitting. • FNN with maxout function may not perform as well as a CNN CNN with maxout function.
[18]		• Data augmentation techniques is not useful. • Lack of post processing which may results in a lower event-based evaluation metric. • Use of Gaussian noise as optimization can increase training time and decrease the accuracy. • LSTM has high model complexity and is not easy to tune. • Sequential nature of LSTM also inhibits parallelization.
[78]		• Additional audio features provide little performance improvement. • LSTM has high model complexity and is not easy to tune. • Sequential nature of LSTM also inhibits parallelization.
[79]		• Simple architecture may be the cause of low accuracy. • CNN lacks long temporal context information. • May require careful tuning of hyperparameters for the parabolic function. • Using a parabola as a confidence function may not be appropriate.
[80]		• Unclear how well this classifier performs in terms of F1-score. • CapsNet requires significant computational resources. • Training of CapsNet is longer, can be more prone to large errors and also lower generalization ability as compared to CNNs. • Hyperparameters must be tuned correctly for maximal performance. • AdaDelta has higher computational time as compared to Adam

achieve maximal performance [16], [17]. As reported in [62], a high threshold value can result in better accuracy for audio with a low polyphonic level, while a low threshold value can result in better accuracy for audio with a high polyphonic level.

C. HYBRID MODELS

Perhaps the most popular hybrid model is the use of Convolution Recurrent Neural Network (CRNN). The reason behind its popularity could be due to the combination of two can supplement each other. As explained by Cakir *et al.* [14] CNN can learn filters that are shifted in both time and frequency. In addition, CNN is capable of capturing energy modulation patterns across time and frequency when applied to spectrogram-like inputs [87]. However, CNN lacks long temporal context information [14]. On the other hand, RNN can overcome this constraint by integrating information from an earlier time window but it cannot capture the invariance in the frequency domain [14]. Thus, the combination of two can overcome the shortcomings while providing the benefits of both approaches.

As seen in Figure 11, the architecture of a CRNN is straightforward; simply stack a CNN over an RNN so that features map extracted by the CNN can be passed directly to

the RNN. Finally, to accommodate multi-label classification, simply apply a global threshold on the output layer.

FIGURE 11. Flowchart of a CRNN.

Cakir *et al.* [14] proposed stacking CNN with a Gated Recurrent Unit (GRU) for polyphonic SED. The idea of GRU is similar to an LSTM, where its motivation is to overcome the vanishing or exploding gradient problems faced by a conventional RNN.

A gated recurrent unit (GRU) was proposed by Cho *et al.* [88] to make each recurrent unit to capture dependencies of different time scales adaptively. It is similar to an LSTM such that GRU also has gating units that modulate the flow of information inside the unit. However, they do not have separate memory cells [89]. It results in a simpler model with much lesser parameters, and as evaluated by Cakir *et al.* [14], performance between the two models are comparable in their application.

Based on such architecture, Cakir *et al.* [14] reported a single frame segment based F1-score of 69.7 with 0.45 single frame ER and single second segment based F1-score of 69.3 with 0.48 single frame Error Rate (ER) on TUT-SED 2009 dataset which outperforms CNN, RNN, GMM, and FNN. However, such architecture was not the best classifier when tested on TUT-SED 2016 dataset in terms of F1-score even though Cakir *et al.* [14] architecture had the lowest single frame and single second ER. Cakir *et al.* [14] reported a single frame segment based F1-score of 27.5 that is lower than RNN that has an F1-score of 27.6 and a single second segment based F1-score of 30.3 which is lower than GMM which has an F1-score of 32.5. This architecture was also not the best architecture when tested on CHIME-Home evaluation dataset. CNN achieved the lowest Equal Error Rate (EER) of 10.7, while Cakir *et al.* [14] CRNN achieve an EER of 11.3.

On the other hand, Jung *et al.* [90] proposed using BLSTM to stack with CNN. The architecture and training scheme remains largely similar as compared to [14]. For Jung *et al.* [90] architecture, they had more recurrent layers and instead of using batch normalization on the recurrent layers, they applied layer normalization. Since BLSTM produces two outputs that differ due to the input order, the outputs were concatenated together and used as input to the subsequent layer. As an additional measure to increase the training accuracy, Jung *et al.* [90] proposed the application of transfer learning. It was achieved by training a Convolutional Bidirectional LSTM (CBLSTM) using a set of synthetic data. Then transferred the adjusted weights to a new CBRNN training using a new set of training data.

Based on this architecture, Jung *et al.* [90] achieved a single frame segment F1-score of 49.9, which is much higher as compared to Cakir *et al.* [14] architecture producing a single frame segment F1-score of 27.5. With the application of transfer learning, Jung *et al.* [90] achieved an even higher single frame segment F1-score of 55.9. Jung *et al.* [90] system also have a lower single frame ER of 0.56 as compared to Cakir *et al.* [14] architecture, which had a single frame ER of 0.98. However, due to the application of transfer learning, there is a need to generate synthetic data that requires additional effort. Moreover, as mentioned earlier, LSTM has high model complexity and is not easy to tune.

Adavanne *et al.* [91] also proposed a CBLSTM but with the inclusion of spatial features extracted from the different channels which were the extension of his work in [78]. Adavanne *et al.* [91] suggested using log mel energies, TDOA estimated from GCC-PHAT, GCC-PHAT, dominant frequency and their amplitudes (dom-freq) as well as Autocorrelation (ACR) which was used to estimate pitch. Adavanne *et al.* [91] idea were to stack each feature from each channel over the other to form a volume. By slicing the volume along with a particular time frame, all multichannel features corresponding to the time frame can be extracted. Since there was a difference in features dimension, Adavanne *et al.* [91] proposed to use separate CNN to learn local shift-invariant features. Features were then concatenated and passed to the BLSTM.

Based on different feature combinations, Adavanne *et al.* [91] concluded that using log mel energies with dom-freq was the best combination for TUT-SED 2009 which achieved a single second segment based F1 score of 71.7 with a single second ER of 0.43. Whereas, the best feature combination for TUT-SED 2016, was log mel energies with TDOA which achieved a single second segment F1 score of 35.8 with a single second ER of 0.95.

However, results were not impressive and shown that features extracted from the stereo channel were beneficial, but not with additional features. For TUT-SED 2009, a classifier trained using only log mel energies can already achieve a single second F1-score of 71.1 with 0.43 ER which means that the best classifier only performed marginally better. Whereas for TUT-SED 2016, LSTM trained using log mel energies already achieved a single second segment F1-score of 35.6 with Error Rate (ER) of 0.93 in their earlier work [78]. With such a complex network architecture and additional features, accuracy was only improved by 0.2 but at the expense of ER.

Adavanne *et al.* [92] further extended his work with the use of a 3D CNN and only using log mel energies with GCC-PHAT. As compared to the earlier work [91], the main difference in [92] is 1) the first layer of CNN use to extract features from log mel energies and GCC-PHAT is a 3D CNN, 2) the bidirectional LSTM is replaced with a bidirectional GRU, 3) early stop is based on accuracy improvement over 100 epochs instead of 50.

Adavanne *et al.* [92] then tested the architecture using the different combinations of features on a synthetic dataset and the results reiterated that GCC-PHAT as an additional feature was not helpful and cannot increase the accuracy

of a classifier. With this conclusion, Adavanne *et al.* [92] then tested architecture on TUT-SED 2017 and achieved a single second F1-score of 67.5 with a single second ER of 0.35 which won a similar architecture using 2D CNN which achieved a single second F1-score of 64.8 with a single second ER of 0.37.

Xia *et al.* [93] also extended his work in [79] by introducing Auxiliary Classifier Generative Adversarial Network (AC-GAN) to balance the dataset between event class and the use of CRNN instead of a CNN.

Xia *et al.* [93] idea was to use AC-GAN produce virtual sound samples that are close to the real sound samples with additional conditions such as the sound event class information and the sound event localization information. Note that localization in this context refers to the location of the activated frames in the audio; it does not refer to the location of the sound source. Thus, generated samples would be more convincing than using simple data augmentation methods proposed in [18] that has limited effectiveness. Generated samples were then used together with the real samples to train a CRNN. Based on such a proposal, Xia *et al.* [93] reported a single second segment based F1-score of 31.1 with ER of 0.58 on the TUT-SED 2016 development dataset (Home) but scoring only a single second segment based F1-score of 19.6 with ER of 0.84 on the evaluation dataset. As results were only reported on the part of the dataset, it is unclear how well it performs on the entire dataset and thus cannot be compared with other methodologies that tested their methods over the whole dataset.

The methodology was also tested on TUT-SED 2017 dataset and reported a single second segment based F1-score of 52.7 with ER of 0.34 on the development dataset. On the other hand, Xia *et al.* [93] achieved a single second segment based F1-score of 48.3 with ER of 0.59 on the evaluation dataset. As compared to [92] result on the development dataset, Xia *et al.* [93] did not perform better than [92] and performance has an approximate 15% gap. However, based on the evaluation results, Xia *et al.* [93] can easily clinch the top place in the DCASE 2017 task 3 challenge.

As seen from the results, the performance of such methodology applied to different datasets has a large fluctuation. It could be due to the fact that AC-GAN tends to generate near-identical samples for most classes as the number of labels increases [94]. In the experiment, for SED-2016, AC-GAN was used to balance 5 minority classes for SED-2016 while AC-GAN was used to balance only 3 minority classes for SED-2017.

Secondly, AC-GAN imposes perfect separability, which is disadvantageous when the supports of the class distributions have significant overlap [94]. Since the method is using a soft label instead of a hard label, the use of parabolic function may not be appropriate. It is due to the events with continuous output that do not have much fade in fade out effect such as vacuum cleaner or blender. As a result, it should have the maximal confidence throughout most of the annotated frame instead of just the center frame. Moreover, the hyperparameter of the parabolic function may need to be adequately tuned for maximal performance.

Ding and He [95] proposed an adaptive multi-scale detection method that combined the idea of an hourglass network with Bidirectional GRU (BGR). It is a CRNN but with much higher sophistication. The hourglass network comprises of a series of convolutional and max-pooling layers to process features down to a very low resolution. At each max pooling step, the network branches off and applies more convolutions at the original pre-pooled resolution. After reaching the lowest resolution, the network begins the top-down sequence of upsampling and a combination of features across scales [96]. Since this was a hybrid model, the combination of features at each scale was sent to a bidirectional GRU. The outputs of the GRUs were then upsampled according to the scales and multiplied with a set of weights for each scale to balance the contribution of each branch. The resulting values were then added up and sent to the output layer. In their study, Ding and He [95] proposed using a 4 layers hourglass network with 3 layers bidirectional GRU at each scale. Based on such architecture, Ding and He [95] achieved a single second F1-score of 48.7% with an ER of 0.7821 on TUT-SED 2016 evaluation dataset and a single second F1-score of 43.6% with an ER of 0.7723 on TUT-SED 2017 evaluation dataset.

With these results, Ding and He architecture [95] easily won the top three contestants' architecture in each challenge. However, such architecture requires much higher computational resources as compared to a CRNN due to the combination of features at different scales. In addition, such a network can easily overfit with a small number of samples. Moreover, as compared to [93], such a complex network cannot win a conventional CRNN with data augmentation using GAN.

D. SUMMARY OF HYBRID MODELS

In the previous section, different types of CRNN were reviewed in detail and discussed. Although there are various ways to construct a CRNN, the main difference lies in the choice of RNN. Although GRU and LSTM may be on par in terms of accuracy, GRU has much lesser parameters as compared to an LSTM. It should also be noted that although CRNN can further increase the detection accuracy, it still requires a large amount of strongly labeled training data to learn the mapping between features and event class. Moreover, stacking CNN with RNN will increase the computational complexity by several folds and this would result in much longer training time. In addition, such architecture does not allow parallel computing due to the sequential nature [97]. Lastly, CRNN also requires the use of a global threshold to determine the activation of sound events, which can affect the detection accuracy if this is not tuned properly.

In this section, features used by different authors are presented in Table 8. Similarly, mel energies remain as the most popular feature. Whereas, the details of their architecture can be seen in Table 9. For the Filters and Units in Table 9, it should be noted that the number represents the number of filters or units in each layer. Example in 256, 256, 256 in the

TABLE 8. Features used by different authors.

References	Features	Parameters used for Spectrum Calculation				Additional Information
		Window	Window Length (ms)	Overlap (%)	Mel-Filter Banks	
[14]	• Log mel energies	Hanning	40	50	40	Z-score normalization of energy band.
[90]	• Log mel energies	-	50	50	40	Clipped below -100db and Min-Max normalized of energy band.
[91]	• Log mel energies • TDOA • GCC-PHAT • dom-freq • ACR	Hamming	40	-	40	dom-freq picked from 100 to 4000Hz. ACR calculated in 40ms window and in the range of 107.5Hz to 4410 Hz.
[92]	• Log mel energies • GCC-PHAT	-	40	50	40	GCC extracted in 120, 240, 480 ms window
[93]	• Log mel energies	-	40	50	40	Data augmentation using AC-GAN
[95]	• Mel energies	-	40	50	128	Z-score normalization of energy band.

Filter column refers to 256 filters in the first layer, 256 filters in the second layer, and 256 filters in the third layer. Whereas, for Pooling Size in Table 9, the value indicates the pooling size at each layer.

Thus, a pooling size of (1, 5), (1, 4), (1, 2) represent that the T-F input only has its frequency dimension reduced by 5 times in the first layer, further reduced by another 4 times in the second layer and another 2 times in the last layer. Using a 40 mel bands T-F input as an example, the 40 bands become 1 band in 3 stages: $40 \rightarrow 8 \rightarrow 2 \rightarrow 1$.

Finally, limitations and results of all the CRNN architectures are presented in Table 10 and Table 11 respectively.

E. MODELS UTILIZING WEAKLY LABELED DATA

As seen in the earlier subsections, methodologies proposed requires the use of strongly labeled data for model training. However, collecting this type of data is often time-consuming as it often requires repeated listening and adjusting of label time boundaries on a visual interface [98]. Furthermore, the sizes of such datasets are often limited to minutes or a few hours [3], [98], [99]. In certain scenarios such as an approaching vehicle, the onset and offset time is ambiguous due to the fade in and fade out effect [100] and is subjective to the person labeling the event [101]. If audio frames are erroneously annotated, it will create unnecessary noise for any classifier [62].

However, there exists a substantial amount of data that is weakly labeled (i.e., sound events is annotated without the onset and offset information) and it is possible to learn a high-quality and dynamic predictor [99] using only such data. In this section, different methodologies utilizing weakly labeled data are reviewed and discussed.

Lee *et al.* [102] proposed an ensemble of CNNs for SED utilizing only weakly labeled data. Lee *et al.* [102] idea was to have two different CNN architectures trained with different inputs. The first CNN architecture which was referred to as the Global Input Model, took the entire T-F representation as the training input. Whereas the second CNN architecture which was referred to as the Separated Input Model, took a smaller segment of T-F representation as training input. These smaller segments are broken up from the original T-F representation using a non-overlapping sliding window with a window size of 1 to 5 seconds and were considered to have the same label as the original T-F representation.

The final prediction was then given by combining the models' probabilities. The framework can be seen in Figure 12.

In their approach, Lee *et al.* [102] proposed an iterative approach to form the ensemble based on accuracy measures. Thus, a model will only be included in the ensemble if it can raise the accuracy of the ensemble. Based on such scheme, an ensemble was formed from a total of 12 models trained using the Global Input Model and Separated Input Model. Using the ensemble, Lee *et al.* [102] reported an F1-score of 52.1 with ER of 0.66 on the DCASE task 4 challenge development dataset and an F1-score of 55.5 with an ER of 0.66 on the evaluation dataset. Based on such a result, Lee *et al.* [102] was able to clinch the first place in the 2017 DCASE Task 4 challenge

The main drawback of such methodology is the computational resource and time required to train a large amount of model and perform the iterative selection to form an ensemble. In their methodology, Lee *et al.* [102] proposed a background subtraction method to eliminate the background noise which is carried out simply by subtracting the median from each mel band. However, this method can significantly degrade model performance that used the entire or half the T-F representation. By treating the smaller segments to have the same label as the original T-F representation can induce noise to the training sample because the smaller samples may not even contain any information regarding the label.

TABLE 9. Proposed hybrid NN architecture by different authors *Note that the first layer is a 3D CNN and the last dimension of the filter size is the number of channels. **Note that this is a 4 layer hourglass network.

References	Model Based On	CNN Details	RNN Details	Loss Function	Gradient Descent Optimizer	Additional Information
[14]	TUT-SED 2009	• Layers: 3 • Filters: 256, 256, 256 • Filter size: (5, 5) • Pooling operator: Max • Pooling size: (1, 5), (1, 4), (1, 2) • Activation function: ReLu	• Type: GRU • Layer: 1 • Units: 256	BCE	Adam	• Dropout rate: 0.25 • Early stop criterion: 100 • Batch normalization after each convolutional layer or fully connected layer • Pooling applied after every convolution layer • Activation function for output layer: Sigmoid
	TUT-SED 2016	• Layer: 3 • Filters: 256, 256, 256 • Filter size: (5, 5) • Pooling operator: Max • Pooling size: (1, 2) , (1, 2), (1, 2) • Activation function: ReLu	• Type: GRU • Layers: 3 • Units: 96, 96, 96			
	CHIME-Home	• Layer: 4 • Filters: 256, 256, 256, 256 • Filter size: (5, 5) • Pooling operator: Max • Pooling size: (1, 2), (1, 2), (1, 2), (1, 1) • Activation function: ReLu	• Type: GRU • Layer: 1 • Units: 256			
[90]	-	• Layers: 3 • Filters: 256, 256, 256 • Filter size: 3 • Pooling operator: Max • Pooling size: (1, 5), (1, 4), (1, 2) • Activation function: Sigmoid	• Type: BLSTM • Layers: 3 • Units: 100, 100, 100 • Activation function: Sigmoid	BCE	Adam	• Learning rate: 0.001 • Decay rate: Exponential • Dropout rate: 0.3 • Batch normalization for each convolution layer. • Pooling applied after every convolution layer • Layer normalization for each BLSTM layer • Activation function for output layer: Sigmoid
[91]	Log mel energies	• Layers: 3 • Filters: 100, 100, 100 • Filter size: (3, 3) • Pooling operator: Max • Pooling size: (1, 2), (1, 2), (1, 2) • Activation function: ReLu	• Type: BLSTM • Layers: 2 • Units: 100, 100	BCE	Adam	• Mini-batch: 32 • Dropout rate: 0.5 • Batch normalization for each convolution layer • Early stop criterion: 50 epochs • Activation function for output layer: Sigmoid
	GCC-PHAT	• Layers: 3				

TABLE 9. Proposed hybrid NN architecture by different authors *Note that the first layer is a 3D CNN and the last dimension of the filter size is the number of channels. **Note that this is a 4 layer hourglass network.

		• Filters: 100, 100, 100 • Filter size: (3, 3) • Pooling operator: Max • Pooling size: (1, 3), (1, 2), (1, 2) • Activation function: ReLu				
	ACR	• Layers: 3 • Filters: 100, 100, 100 • Filter size: (3, 3) • Pooling operator: Max • Pooling size: (1, 10), (1, 4), (1, 2) • Activation function: ReLu				
	TDOA, dom-freq	• Layer: 1 • Filters: 100 • Filter size: (3, 3) • Activation function: ReLu				
[92]	Log mel energies	• Layers: 3 • Filters: 64, 64, 64 • Filter size: (3, 3)* • Pooling operator: Max • Pooling size: (1, 5), (1, 2), (1, 2) • Activation function: ReLu	• Type: GRU • Layers: 2 • Units: 64, 64 • Activation function: Hyperbolic Tangent	BCE	Adam	• Learning rate: 0.0001 • Mini-batch: 128 • Dropout rate: 0.2 • Batch normalization for each convolution layer • Pooling applied after every convolution layer • Early stop criterion: 100 epochs • Activation function for output layer: Sigmoid
	GCC-PHAT	• Layers: 3 • Filters: 64, 64, 64 • Filter size: (3, 3)* • Pooling operator: Max • Pooling size: (1, 5), (1, 3), (1, 2) • Activation function: ReLu				
[93]	-	• Layers: 3 • Filters: 128, 128, 128 • Filter size: (3, 3) • Pooling operator: Max • Pooling size: (1, 5), (1, 2) , (1, 2) • Activation function: ReLu	• Type: GRU • Layers: 2 • Units: 32, 32	BCE	Adam	• Learning rate: 0.001 • Batch normalization for each convolution layer • Pooling applied after every convolution layer • Activation function for output layer: Sigmoid
[95]	-	• Layers: 4** • Activation function: ReLu	• Type: GRU • Layers: 3	Weighted Average BCE	Adam	• Learning rate: 0.001 • Mini-batch: 45 • Batch normalization for each convolution layer • Pooling applied after every convolution layer • Activation function for output layer: Sigmoid

TABLE 10. Limitation of proposed hybrid NN methodologies.

References	Common Limitations	Individual Limitations
[14]	• Requires large amount of strongly label training data. • Requires careful tuning of threshold value to indicate presence of event. • Complex architecture which may require high computational time.	• Accuracy was not the best across different dataset.
[90]		• Require additional effort to generate synthetic data for transfer learning. • LSTM has high model complexity and is not easy to tune.
[91]		• Additional audio features provide little performance improvement. • LSTM has high model complexity and is not easy to tune.
[92]		• Additional features did not help to increase accuracy.
[93]		• May require careful tuning of hyperparameters for the parabolic function. • Using a parabola as a confidence function may not be appropriate. • AC-GAN produces sample with less diversity as classes increases. • AC-GAN imposes perfect separability which is disadvantageous.
[95]		• May requires much higher computational resource as compared to a CRNN. • Such network can easily overfit with a small amount of samples. • Cannot perform as well as a CRNN with data augmentation using GAN

TABLE 11. Accuracy of hybrid NN on different dataset.

References	Dataset	Segment Based F1-Score (%)	Segment Based ER	Other Metric
[14]	TUT-SED 2009	1 Frame: 69.7 1 Second: 69.3	1 Frame: 0.45 1 Second: 0.48	
	TUT-SED 2016 Development Dataset	1 Frame: 27.6 1 Second: 30.3	1 Frame: 0.98 1 Second: 0.95	
	CHiME-Home Development Dataset			EER: 13
	CHiME-Home Evaluation Dataset			EER: 11.3
[90]	TUT-SED 2016 Development Dataset	1 Frame: 55.9	1 Frame: 0.56	
[91]	TUT-SED 2009	1 Second: 71.7	1 Second: 0.43	
	TUT-SED 2016 Development Dataset	1 Second: 35.8	1 Second: 0.95	
[92]	TUT-SED 2017 Development Dataset	1 Second: 67.5	1 Second: 0.35	
[93]	TUT-SED 2016 Development Dataset (Home)	1 Second: 37.1	1 Second: 0.58	
	TUT-SED 2016 Evaluation Dataset (Home)	1 Second: 19.6	1 Second: 0.84	
	TUT-SED 2017 Development Dataset	1 Second: 52.7	1 Second: 0.34	
	TUT-SED 2017 Evaluation Dataset	1 Second: 48.3	1 Second: 0.59	
[95]	TUT-SED 2016 Evaluation Dataset	1 Second: 48.7	1 Second: 0.78	
	TUT-SED 2017 Evaluation Dataset	1 Second: 43.6	1 Second: 0.77	

The idea of disregarding the label position was also tested in [101] where results have shown that it can induce noise to the training examples which can result in a lower segment based F1-score.

Xu *et al.* [103] proposed a gated CRNN and a temporal attention based localization method. The system was a joint system that produced the audio labels as well as their respective annotations in the audio. The idea was to make use of a FNN with dual activation function that was connected to the CRNN.

The first activation function in the FNN was the softmax activation function which was used to infer the temporal locations for each occurring class and attended to the most salient frames for each class. Thus, this allowed a localization vector (i.e., onset and offset vector) to be produced.

The second activation function was the sigmoid activation function which performed classification at each frame. Thus, providing the classification output. To obtain the final audio tag prediction, the localization vector was multiplied by the classification output and averaged across the time axis which can be thought of as attention pooling. The workflow is illustrated in Figure 13.

To prevent overfitting due to an imbalanced dataset, Xu *et al.* [103] proposed a new training selection scheme which would include samples from all classes in the batches using for training. The scheme would follow the similar distribution ratio of each class but ensure that there is at least one sample from the minority classes, while the majority class would only be at most 5 times more than the minority class [103] [104].

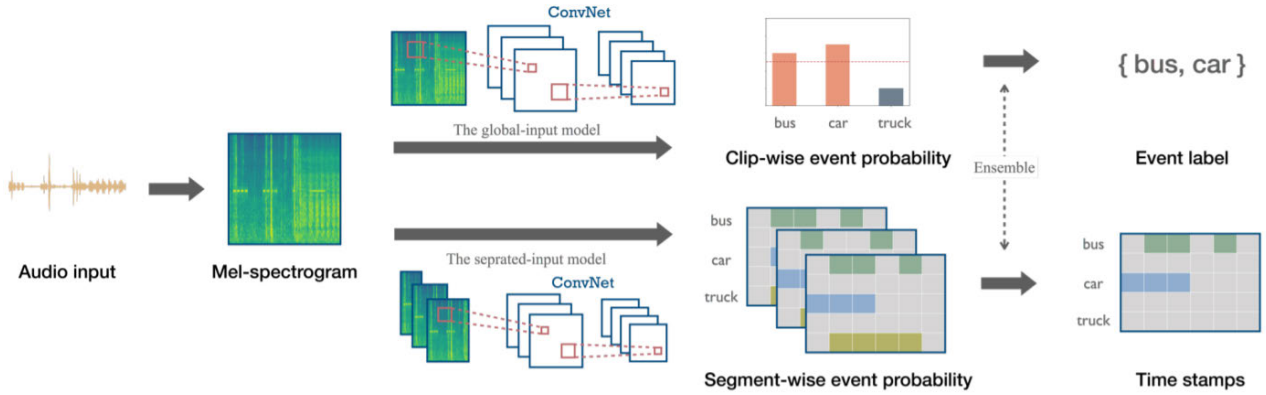


FIGURE 12. Flowchart of Lee *et al.* framework [102].

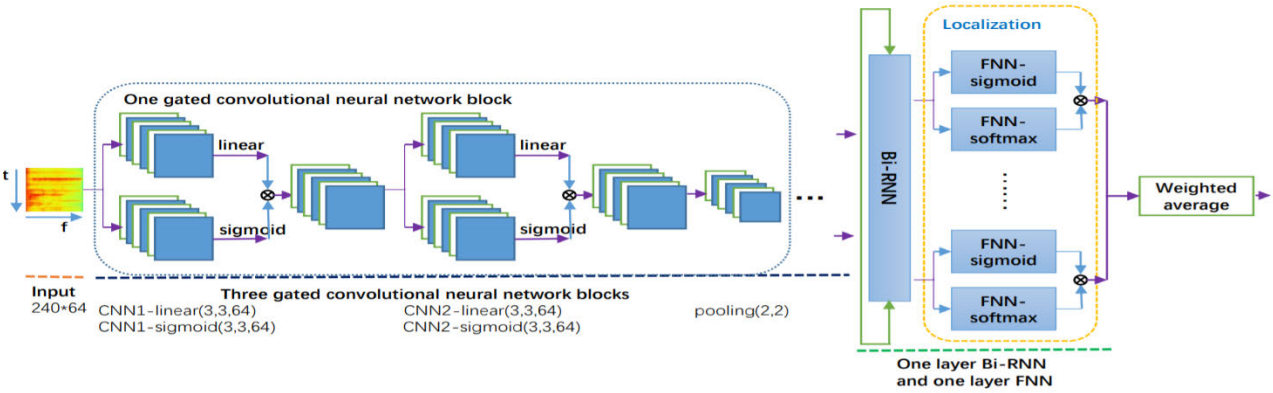


FIGURE 13. Flowchart of Xu *et al.* framework [103].

Xu *et al.* [103] also proposed the use of Gated Linear Unit (GLU) activation function in CNN which can be regarded as a local attention scheme. GLU is similar to other gating mechanisms in LSTM or GRU which control the information flow to the next layer and is defined as [98] [100]

$$Y = (W * X + b) \otimes \sigma\,(V * X + c) \tag{2}$$

where X is the input and σ is the sigmoidal function. W and V are the filters with b and c as the bias. The benefit of GLU is that it can reduce the gradient vanishing problem for the deep network by allowing a linear path for the gradient propagation while keeping nonlinear capabilities through the sigmoid operation [105].

To further improve the detection accuracy, Xu *et al.* [103] proposed fusing results generated during each training epoch and an ensemble system where one system was trained using MFCCs while the other was trained using log mel energies. Outputs from both systems will then be averaged to retrieved the final posterior. The final posterior was then thresholded by an array of class-dependent thresholds to determine the occurrence of the sound event [104].

Based on such setting, Xu *et al.* [103] achieved a single second segment based F1-score of 49.7 with ER of 0.72 on the DCASE 2017 task 4 challenge development dataset and a single second segment based F1-score of 51.8% with an ER of 0.73 on the DCASE 2017 task 4 challenge evaluation dataset.

However, based on the evaluation results, such a system can at best ranked second in the DCASE 2017 challenge.

This system was also found to perform rather poorly on event-based evaluations for a similar task in DCASE 2018 [106]. In addition, experiments also found that such architecture cannot be simplified and will result in poor detection accuracy [107] and thus requiring a large computational cost due to the sophisticated network. The use of GLU in their methodology can increase the total number of parameters to be learned due to the use of 2 different filters. Finally, it was also found that linear softmax pooling function may work better instead of using attention pooling [108].

Kong *et al.* [97] then proposed a CNN-Transformer (CNNT) for SED. The motivation of their work was to reduce the training time of a CRNN by making use of a transformer to replace the RNN. The CRNN has to be calculated sequentially. Thus, making it quite challenging to complete the training process in a short amount of time [97]. The transformer can take a long time dependency into consideration for a

system. But with the benefit of allowing parallel computing that gives a good alternative to RNN[97].

To perform SED, Kong *et al.* [97] made use of 3 different sets of thresholds. The first set would allow the audio to be tagged. The second set was a set of upper bound threshold which would indicate the frames that contain the respective sound event. However, this may result in several false negatives. Thus, the third set of the lower bound threshold was utilized to determine if the neighboring frames contain the same sound event. As there were many thresholds, manual tuning of these values can be difficult and inefficient. Thus, Kong *et al.* [97] proposed an automatic threshold optimizer. The idea was to calculate the gradients over the thresholds. Thus, allow an optimization method such as Adam to be applied.

Based on such implementation, Kong *et al.* [97] achieve a segment based F1-score of 52.4 with ER of 0.75 on the development dataset. The F1-score of 57.3 with an ER of 0.75 on the evaluation dataset was obtained. However, based on the results of both DCASE 2017 development and evaluation data, the CNNT hybrid was not able to achieve better performance than a CRNN with the attention layer. Although the Transformer can be trained faster than an RNN, its weakness is in the decoding process. Due to the auto-regressive architecture and self-attention in the decoder, the decoding process can be slow [109], [110].

Lu [111] then proposed a modified version of Xu *et al.* [103] system. Instead of using GLU, Lu [111] proposed to use Context Gating (CG) as the activation function for CNN which can be given as [112]

$$Y = \sigma\left(W * X + b\right) \otimes X \tag{3}$$

where X is the input and σ is the sigmoidal function. W is the set of filters with b as the bias. Such implementation also allows non-linear interactions among activations of the input representation but has the benefit of improved efficiency as compared to GLU [112]. This is because there is only a set of weights to be learned which reduces the number of parameters.

Lu [111] then adopted the Mean-Teacher semi-supervised method [113] in their methodology. The idea is to have two similar models where one is called the Teacher Model (TM) and the other is called the Student Model (SM) to learn a similar task at the same time. However, the weights of the SM are updated through gradient descent. But the weights of the TM are updated as an exponential moving average of the SM weights [113]. Such a training scheme is based on the fact that using average model weights over training epochs instead of using the final weights directly can produce a more accurate model [113]. In Lu [111] study, he proposed the use of two cost functions to update the model's weights, 1) the classification cost of SM on the labeled data and 2) the consistency cost between SM and TM on both the labeled and the unlabeled data. As mentioned by Lu [111], a trained TM will usually be better than a trained SM.

Similar to [103], Lu [111] used an ensembled system to produce a more accurate model where Lu [111] proposed to combine the results from different systems using different consistency costs for the Mean Teacher model. Based on this system, Lu [111] achieved an event-based F1-score of 34.4% with an ER of 1.16 on the development dataset. An event-based F1-score of 32.4% on the evaluation dataset was also achieved. Based on the score on the evaluation dataset, Lu [111] clinched the 1st place in the 2018 DCASE Task 4 challenge.

The main drawback of this model is the large number of models to be trained to form the ensemble. Due to the nature of the student-teacher model, each model with a different consistency cost requires two models (i.e., SM and TM) to be trained synchronously, this can result in a significant increase in computational cost and resource burdens. Since the model was based on [103], the attention pooling used by Lu [111] may not work as well as linear softmax pooling function. It was also found that such a system performed rather poorly for sound events of short duration [26]. Finally, consistency cost requires careful tuning to achieve maximal performance. Since Lu [111] proposed an ensemble approach, the impact of consistency cost was alleviated.

Lin *et al.* [114] also proposed a teacher-student frame for SED using weakly labeled data. But instead of using similar models, TM was proposed to have higher model complexity and was hypothesized that such a model could integrate the audio contextual information, thus, capable of producing a better clip level prediction. Whereas, the SM was a simpler model was hypothesized to be better at frame-level prediction. Lin *et al.* [114] methodology can be thought of as the distillation approach [115] or the mutual learning [116] framework. The proposed teacher and student model can be seen in Figure 14.

The entire training procedure can be broken down into two different stages where the transition from stage one to stage two is controlled by the training epochs defined by the user. The first stage can be thought of as the student learning stage while the second stage can be thought of as the mutual learning stage. In the first stage, model parameters of both TM and SM are updated based on a combined cost which consists of three different components. The first two components were the classification cost of TM and SM on the labeled data which can be given as

$$C_1 = J\left(y_k, TMy_k\right) \tag{4}$$

$$C_2 = J\left(y_k, SMy_k\right) \tag{5}$$

where J denotes the cost function and y_k as the true label of the k sample of the weakly labeled dataset. TMy_k and SMy_k represents the predicted label of the k sample in the weakly labeled dataset by the TM and SM respectively. The third component was the classification cost of TM and SM on the unlabeled data where predictions by TM were considered as the actual label which can be represented as

$$C_3 = J\left(TMp_i, SMp_i\right) \tag{6}$$

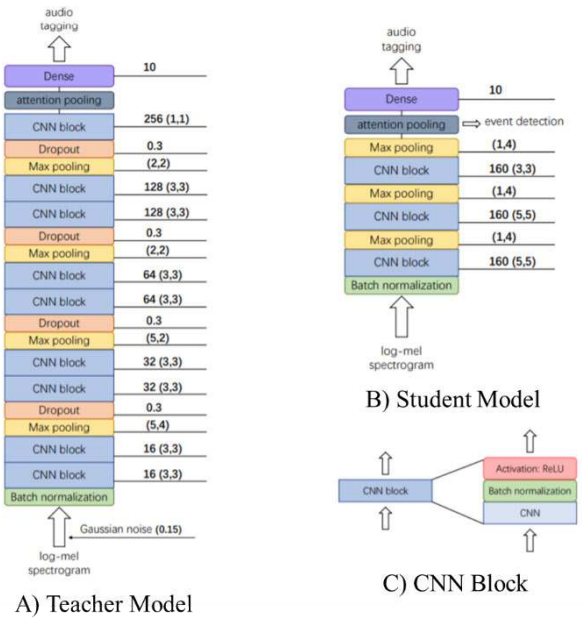


FIGURE 14. Flowchart of Lin *et al.* framework [114].

where p_i denotes the predicted label of the i sample of the unlabeled dataset. TMp_i and SMp_i represents the predicted label of i sample in the unlabeled dataset by the TM and SM respectively. Thus, in the first stage, the learning procedure effectively forces the SM to learn from the TM.

After a specific training epoch, the training procedure transit into the second stage where the combined cost was added with another component, C_4 which can be represented as

$$C_4 = \gamma \times J\,(SMp_i, TMp_i) \tag{7}$$

where γ is a hypermeter, which determines how much should the TM learns from the SM. The models' parameters were then updated using the combined cost. Thus, such a learning scheme will allow the two CNN models to be trained simultaneously and forcing them to learn from each other which Lin *et al.* [114] hypothesized can increase the abilities of the two models.

Based on such architecture, Lin *et al.* [114] reported an event-based F1 score of 39.5 which won Lu [111] system when tested on the DCASE 2018 evaluation dataset.

Lin *et al.* [117] then extended their work by remodeling the feature. As suggested by Lin *et al.* [117], this will create a set of disentangled features and would mitigate the effect of overlapping sound events. Subsequently, Lin *et al.* [117] suggested using median filters with different window sizes based on the event class to smooth the output. Finally, an ensemble of models with different γ was trained to produce the audio prediction.

Based on these improvements, it won the first place in the DCASE 2019 task 4 challenge with an event-based F1-score of 45.4 on the development dataset and an event-based F1-score of 42.7 on the evaluation dataset.

However, the drawback of [114] and [117] is similar to [111], two models (i.e. SM and TM) have to be trained synchronously which can result in a significant increase of computational cost and resource burdens. The use of ensemble in [117] can further aggravate this problem. There are also several hyperparameters to be tuned. Firstly, how can the training epochs be determined effectively to allow proper transition from stage one to stage two? Secondly, how much should the TM learn from the SM?

Unfortunately, both of these hyperparameters have to be tuned manually to achieve maximal performance. In [117], the different window sizes of the median filters used were derived based on the average event duration in the synthetic dataset, which may not be perfect.

Kothinti *et al.* [106] approach was to develop two separate models for different purposes, as illustrated in Figure 15. The first model, which was a combination of Restricted Boltzmann Machine (RBM), conditional RBM (cRBM) and Principle Component Analysis (PCA), was in charge of the event boundary detection. Whereas the second model which was an ensemble of CRNNs, was in charge of providing the audio label.

For the first model, a spectrogram was used as input for the RBM to capture local-spectrotemporal dependencies. Outputs were subsequently passed to an array of cRBMs to generate the final high representation of the acoustic signal. PCA was then applied to reduce the dimensions of this signal representation. The closest preceding sample at 25% of the maximal value in the reduced representation was then determined as the onset of an event. On the other hand, the offset was determined based on the threshold set on the short-term energies of the audio signal.

For the second model, log mel energies were used as the input for the ensemble. Kothinti *et al.* [106] suggested using three different CRNNs. The first two systems had the same architecture but were trained differently. The first system was trained using only weakly labeled data while the second was trained using weakly labeled data and augmented data which was generated by mixing weakly label audios. On the other hand, the third system was the DCASE task 4 baseline system trained using only weakly label audio. The key difference between the three systems besides the type of training inputs used was the number of filters and kernel size used in each convolutional layer. Finally, the predictions of the three systems were combined to give the final prediction through majority voting.

The ensembled system then achieved an event-based F1-score of 30.05 with ER of 1.36 when tested on the development dataset. It also produced an event-based F1-score of 25.4 with ER of 1.19 when tested on the evaluation dataset. However, based on the accuracy, it was ranked 3rd (in terms of teams) in the DCASE 2018 task 4 challenge that suggests a large room for improvement.

Such a system is also computationally expensive due to different models and ensemble used. In addition, it was found that the accuracy of the boundary detection system is lower

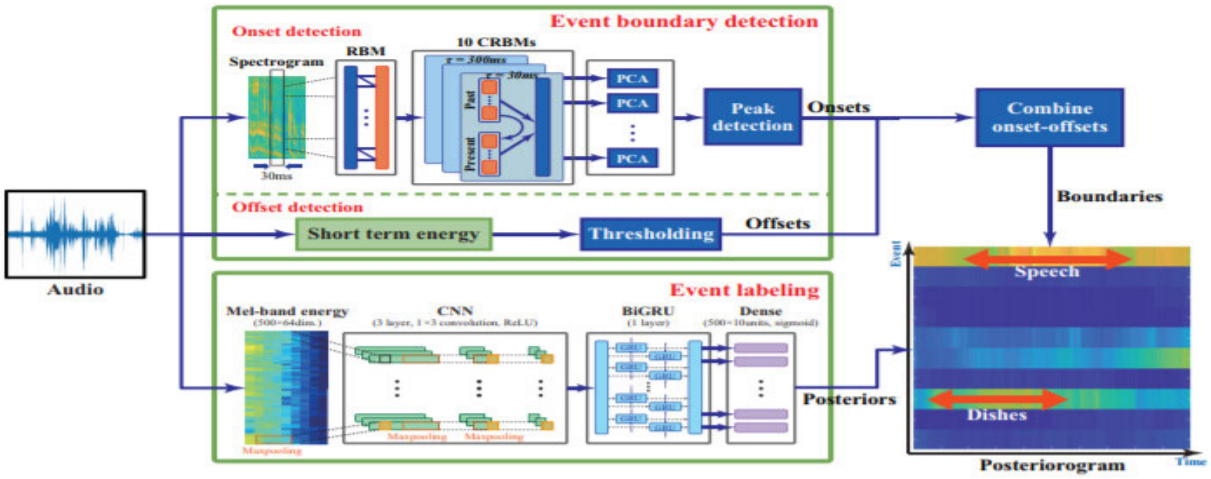


FIGURE 15. Proposed methodology by Kothinti *et al.* [106].

for audio with overlapping events. But the impact was only mitigated due to the error tolerance used for evaluation. However, the cause of this phenomenon is not directly understandable. It was mentioned in [118], it may not be a good choice to use Contrastive Divergence (CD) to train a cRBM for prediction purposes. However, it is unclear if it will also affect the representation learning, which in turn affect the boundary detection accuracy. Theoretical analysis of CD is also difficult [119] that can prohibit further investigation. On the other hand, there is also a possibility that the accuracy of boundary detection is affected by suboptimal thresholds used for boundary detection. Finally, it was also found that the audio tagging system was poor at classifying the detected events that deteriorate the overall performance significantly.

Pellegrini and Cances [107] then proposed using a combinative approach where CNN was used for audio tagging and CRNN was used for SED. For the audio tagging model, the last convolution block was followed by global average pooling and global max pooling, then by a dense layer with 1024 units and finally the output layer. The probabilities from the output layer were then thresholded by a set of class-dependent thresholds that were optimized by the Genetic Algorithm. It would allow the final audio tag to be derived.

For the SED model, the last convolution block was followed by a Bidirectional GRU (BGRU), then a dense layer with 64 units and then the output layer. Another distinct difference would be the cost function where the SED model took Cosine Similarity between classes in consideration. It was proposed as an additional measure to penalize overlapping events. The output from this model was a set of temporal predictions for each class where Pellegrini and Cances [107] only kept the temporal predictions of classes detected as positive by the audio tagging model. These values were subsequently rescaled to 0 and 1 and smoothed with a sliding average window. Finally, the final segments were obtained by a global threshold to detect the onsets and offsets of events.

Based on such implementation, Pellegrini and Cances [107] achieved an event-based F1-score of 34.75 when tested on the DCASE 2018 development dataset and an event-based F1-score of 26.2 when tested on the DCASE 2018 evaluation dataset. However, based on such accuracy, it was ranked 3rd (in terms of teams) in the DCASE 2018 task 4 challenge.

Besides the need to train two different models, several hyperparameters were coarsely determined. For example, the threshold used to detect the final onsets and offsets of events and the regularization weight to determine the amount of contribution by Cosine Similarity to the loss function. A coarsely determined threshold value can have an adverse effect on model accuracy. The overall accuracy is also dependent on the audio tagging model's accuracy that only achieved an accuracy of 77%. It was found that if audio labels are correctly given, the system can have an accuracy improvement of another 10%. Pellegrini and Cances [107] also utilized GLU. But it can increase the total number of parameters to be learned due to the use of two different filters. Although Cosine Similarity was added to penalize overlapping events, it was not always helpful. In specific scenarios, it can help to decorrelate overlapping classes. But in certain situations, it cannot provide decorrelation and can even cause one of the classes to be undetectable.

F. SUMMARY OF MODELS UTILIZING WEAKLY LABELED DATA

In the previous section, different methodologies with their limitations were discussed in detail. To leverage a large amount of weakly labeled data, most of the methods adopted the use of ensemble or training of two different models where one is in charge of audio tagging while the other is in charge of boundary detection. However, this will means that it requires much more computational resources and time to train a different number of models. It is also important to point out that the model using weakly labeled data is not as effective as its counterpart using strongly labeled data. Thus, this remains

an open research area to show how data can be utilized effectively. Finally, the use of thresholds to determine the event activation remains a norm, but the researchers [97], [107] presented some interesting ideas to tune it to an optimal value automatically. As shown in the paper [97], the accuracy did increase based on this optimization.

In this section, the features used by different authors are presented in Table 12. Similarly, mel energies remain the most popular as a feature and will most probably stay as the gold standard for SED.

Table 13 then presents the information of the architecture proposed. Naming convention, writing style remains the same as the previous section to ensure consistency. Finally, results and limitations of all the CRNN architectures are presented in Table 14 and Table 15 respectively.

IV. PUBLIC DATASET AND EVALUATION METRIC USED BY DIFFERENT AUTHORS

In the earlier sections, various methodologies were discussed in detail. In this section, different public datasets and evaluation metrics used by various authors will be described.

A. TUT-SED 2016

TUT-SED 2016 is a dataset that consists of real-life recordings that were recorded at high quality and carefully annotated to provide high-quality audio for polyphonic SED. Each recording was performed in a different location, namely: a residential area and home environment, to satisfy the requirement for high acoustic variability.

Each recording was recorded using binaural Soundman OKM II Klassik/studio A3 electret in-ear microphones and Roland Edirol R09 wave recorder at 44.1 kHz sampling rate and 24-bit resolution [120]. Each recording has an average duration of 3-5 minutes that adds up to a total of 78 minutes of audio. In this dataset, there are a total of seven annotated sound event classes for residential area recordings and a total of 11 annotated sound event classes for home recordings. Table 16 then present the distribution for these events from a different environment.

B. TUT-SED 2017

TUT-SED 2017 is a subset of TUT Acoustic Scenes 2017. It consists of recordings of acoustic street scenes (city center and residential area) with various levels of traffic and other activity and has an average duration of 3-5 minutes. Similarly, each recording was recorded using binaural Soundman OKM II Klassik/studio A3 electret in-ear microphones and Roland Edirol R09 wave recorder at 44.1 kHz sampling rate and 24-bit resolution [121].

The street acoustic scene was selected to represent an environment of interest for the detection of sound events related to human activities and hazard situations. The target sound event classes were chosen to represent common sounds related to human presence and traffic. The selected sound classes for this dataset are brakes squeaking, car, children, large vehicles,

TABLE 12. Features used by different authors.

References	Features	Parameters used for Spectrum Calculation				Additional Information
		Window	Window Length (ms)	Overlap (%)	Mel-Filter Banks	
[102]	• Log mel energies	-	46	78	128	Audio was normalized prior to the calculation of mel energies. All mel bands were then subtracted by the median value.
[103]	• MFCC • Log mel energies	Hamming	64	35	64	
[97]	• Log mel energies	Hanning	32	69	64	Audio resampled to 32000Hz. Frequency range to compute mel band is 50Hz to 14000Hz.
[111]	• Log mel energies	-	93	83	128	Audio resampled to 22050Hz.
[114]	• Log mel energies	-	40	50	64	Normalization of energy bands.
[117]	• Log mel energies	-	40	50	64	
[106]	• Spectrogram		10	-	128	
	• Log mel energies	Hamming	40	50	64	
[107]	• Log mel energies	-	46	50	64	Frequency range to compute mel bands is 20Hz to 11025Hz.

people speaking, and people walking. The events' distribution can be seen in Table 17.

C. DCASE 2017

The DCASE 2017 dataset is a subset of Audio set [122] which comprises an ontology of 632 audio event categories and a collection of 1,789,621 labeled 10-sec excerpts from YouTube videos [122]. To collect such a massive dataset, Google worked with human annotators who listened, analyzed, and verified the sounds they heard within the YouTube 10-second clips. To facilitate the faster accumulation of examples for all classes, Google relied on available YouTube metadata and content-based search to nominate candidate video segments that were likely to contain the target sound [121].

In the DCASE 2017 dataset, there is a total of 17 sound events divided into two categories, namely, a warning sound and vehicle sound. The number of instances for each event is presented in Table 18. However, this dataset is only weakly labeled.

D. DCASE 2018

The DCASE 2018 dataset is also a subset of Audio set [122] and consists of 10 classes of sound events. Similarly, this dataset is also weakly labeled. The total number of recordings is 1578 where the event occurrences are presented in Table 19. In addition to this dataset, there is also another 14412 in the domain but unlabeled recordings.

E. EVALUATION METRIC

As seen in the earlier section, the two most common evaluation metrics used by different authors are the event-based and segment-based F1-score that were described in [123].

Event-based metrics compare system output and corresponding reference events by the event, whereas segment-based metrics compare system output and reference in short time segments [123].

Given the error tolerance, the intermediate statistics for event-based metric are defined as follows [123]:

True Positive (TP): a predicted event with its onset and offset tally with the actual annotation.

False Positive (FP): a predicted event with its onset and offset do not tally the actual annotation.

False Negative (FN): a misdetection, the system did not predict any event even though there is an occurrence of the event.

The intermediate statistics for the segment-based metric follow the same definition but instead of considering the entire event, it considers the accuracy in segments. Based on these definitions, precision, P, and recall, R, are calculated as follow

$$P = \frac{TP}{TP + FP} \tag{8}$$

$$R = \frac{TP}{TP + FN} \tag{9}$$

TABLE 13. Proposed architecture utilizing weakly labeled data.

References	Model 1	Model 2	Loss Function	Gradient Descent Optimizer	Additional Information
[102]	• Type: CNN (Global Input) ○ Layers: 10 ○ Filters: 64, 64, 64, 64, 64, 64, 64, 64, 64, 64 ○ Filter size: (3, 3) for all layers ○ Pooling Operator: Max ○ Pooling after every 2 convolution layers ○ Pooling size: (2, 4), (2, 4), (4, 4), (4, 4) ○ Activation function: ReLu	• Type: CNN (Separated Input) ○ Layers: 10 ○ Filters: 64, 64, 64, 64, 64, 64, 64, 64, 64, 64 ○ Filter size: (3, 3) for all layers ○ Pooling Operator: Max ○ Pooling after every 2 convolution layers ○ Pooling size: (2, 2), (2, 2), (4, 3), (4, 4) ○ Activation function: ReLu		Adam	• Batch normalization for each convolution layer • Global average pooling after the last convolution layer
[103]	• Type: CRNN (Audio Tagging) • CNN ○ Layers: 6 ○ Filters: 64 ○ Filter size: (3, 3) for all layers ○ Pooling operator: Max ○ Pooling after every 2 convolution layers	• Type: CRNN (Temporal Localization) • CNN ○ Layers: 6 ○ Filters: 64 ○ Filter size: (3, 3) for all layers ○ Pooling operator: Max ○ Pooling after every 2 convolution layers ○ Pooling size: (1, 2) through the whole	BCE	Adam	• Minibatch data balancing • Learning rate: 0.001 • Activation function for output layer: Sigmoid and Softmax • Attention pooling

TABLE 13. *(Continued.)* **Proposed architecture utilizing weakly labeled data.**

	○ Pooling size: (2, 2) through the whole model ○ Activation function: GLU • BGRU ○ Layer: 1 ○ Units: 128	model ○ Activation function: GLU • BGRU ○ Layer: 1 ○ Units: 128			after the last layer
[97]	• Type: CNNT • CNN ○ Layers: 8 ○ Filters: 64, 64, 128, 128, 256, 256, 512, 512 ○ Filter size: (3, 3) for all layers ○ Pooling operator: Average ○ Pooling after every 2 convolution layer ○ Pooling size: (4, 1) for the last layer, otherwise (2,2) ○ Activation function: ReLu • Transformer		BCE	Adam	• Learning rate: 0.001 • Mixup with alpha of 1 • Batch normalization after every convolution layer • Mel bins are averaged after the last convolution layer
[111]	• Type: CRNN (TM) • CNN ○ Layers: 7 ○ Filters: 16, 32, 64, 128, 128, 128, 128 ○ Filter size: (3, 3) for all layers ○ Pooling operator: Average ○ Pooling after every layer ○ Pooling size: (2, 2) for the first 2 layers, otherwise (1, 2) • BRNN ○ Layer: 1 ○ Units: 128	• Type: CRNN (SM) • CNN ○ Layers: 7 ○ Filters: 16, 32, 64, 128, 128, 128, 128 ○ Filter size: (3, 3) for all layers ○ Pooling operator: Average ○ Pooling after every layer ○ Pooling size: (2, 2) for the first 2 layers, otherwise (1, 2) • BRNN ○ Layer: 1 ○ Units: 128	-	Adam	• Dropout rate: 0.5 • Activation function for output layer: Sigmoid and Softmax • Attention pooling after the last layer
[114], [117]	• Type: CNN (TM) ○ Gaussian input noise: 0.15 ○ Layers: 9 ○ Filters: 16, 16, 32, 32, 64, 64, 12, 128, 256 ○ Filter size: (1, 1) for the last layer otherwise (3, 3) ○ Pooling Operator: Max ○ Pooling after every 2 convolution layers ○ Pooling size: (5,4), (5,2), (2,2), (2,2) ○ Activation function: ReLu	• Type: CNN (SM) ○ Layers: 3 ○ Filters: 160, 160, 160 ○ Filter size: (3, 3) for the last layer otherwise (5, 5) ○ Pooling Operator: Max ○ Pooling after every convolution layer ○ Pooling size: (1,4), (1,4), (1,4) ○ Activation function: ReLu	Modified BCE	Adam	• Learning rate: 0.0018 • Mini-batch: 64 • Decay rate: 0.8 • Dropout rate: 0.3 • Batch normalization for each convolution layer • Attention pooling after the last convolution layer • Early stop criterion: 20 epochs

TABLE 13. *(Continued.)* **Proposed architecture utilizing weakly labeled data.**

[106]	• RBM+cBRM (Event Boundary Model) ○ Units for RBM: 350 ○ Units for cRBM: 300	• Type: CRNN (Audio Tagging Model) **Variant 1 and 2** • CNN ○ Layers: 3 ○ Filter: 128, 128, 192 ○ Filter size: (1, 3) for all layers ○ Pooling after every layer ○ Pooling size: (1,8), (1, 4), (1,2) ○ Activation function: ReLu • BGRU ○ Layer: 1 ○ Units: 64 **Variant 3** • CNN ○ Layers: 3 ○ Filter: 64, 64, 64 ○ Filter size: (3, 3) for all layers ○ Pooling operator: Max ○ Pooling after every layer ○ Pooling size: (1,4), (1, 4), (1,4) ○ Activation function: ReLu • GRU ○ Layer: 1 ○ Units : 64		CD for RBM and cRBM	• PCA on the output of RBM+cRBM • Global average pooling after the last convolution layer for Variant 3 • Dropout rate: 0.3 for Variant 3 • Early stop criterion: 20 epochs for Variant 3 • Activation function for output layer: Sigmoid
[107]	• CNN (Audio Tagging Model) ○ Layers: 3 ○ Filters: 64 ○ Filter size: (3, 3) for all layers ○ Pooling Operator: Max ○ Pooling applied after every convolution layer ○ Pooling size: (1, 4) for all layers ○ Activation function: GLU	• CRNN (SED Model) • CNN ○ Layers: 3 ○ Filters: 64 ○ Filter size: (3, 3) for all layers ○ Pooling Operator: Max ○ Pooling applied after every convolution layer ○ Pooling size: (1, 4) for all layers ○ Activation function: GLC • BGRU ○ Layer: 1 ○ Units: 64 ○ Activation function: Hyperbolic Tangent	BCE for CNN, BCE with Cosine Similarity for CRNN	Adam	• Dropout rate: 0.2 for CNN layers • Drop rate: 0.5 for audio tagging model • Batch normalization after every convolution layer

TABLE 14. Limitations of proposed architecture utilizing weakly labeled data.

References	Common Limitations	Limitations
[102]	• Large computational resource and time required to train a different number of models.	• Iterative selection method to form an ensemble is time-consuming. • Background subtraction method can significantly degrade model performance that used the entire or half the T-F representation. • By treating the smaller segments to have the same label as the original T-F representation can induce noise to the training sample.
[103]		• Accuracy was low in terms of event-based evaluations for a similar task in DCASE 2018. • Architecture cannot be simplified as it will result in poor detection accuracy • The use of GLU increases the total number of parameters to be learned. • Attention pooling may not work as well as linear softmax pooling function.
[97]		• CNNT was not able to achieve better performance than a CRNN with attention layer. • Decoding process in a Transform is slow.
[111]		• Attention pooling may not work as well as linear softmax pooling function. • Accuracy is low for sound events with a short duration. • Consistency cost requires careful tuning.
[114]		• Several hyperparameters which directly affect the accuracy have to be tuned carefully.
[117]		• Several hyperparameters which directly affect the accuracy have to be tuned carefully. • Window sizes of the median filters were derived based on the average event duration in the synthetic dataset which may not be perfect.
[106]		• Accuracy of boundary detection system is lower for audio with overlapping events which is caused by CD learning or suboptimal threshold values. • Audio tagging system was poor at classifying the detected events.
[107]		• Several hyperparameters which directly affect the accuracy have to be tuned carefully. • The overall accuracy is also dependent on the audio tagging model's accuracy. • The use of GLU increases the total number of parameters to be learned. • Cosine Similarity was not always helpful.

F1-score, F, is then calculated as follow

$$F = \frac{2PR}{P+R} \tag{10}$$

It should be noted that event-based and segment based evaluation measures can place significant emphasis on the onset and offset of the sound events [124]. In addition, the same system with different decision threshold (i.e. audio detection threshold, frame activation threshold) may receive different performance rankings under the same metric. As such, Bilen *et al.* [124] proposed the redefinition of TPs and FPs which allows the calculation of Polyphonic Sound Detection-Receiver Operating Characteristic (PSD-ROC) curve and Polyphonic Sound Detection Score (PSDS) which serves as a more robust alternative to evaluate a SED system.

The idea of a PSD-ROC curve is similar to a conventional ROC curve. The difference in a PSD-ROC curve lies in the axes where the x-axis represents the effective FP Rate (eFPR) while the y-axis represents the effective TP Rate (eTPR). The basic definitions will be given in the following paragraphs. It is recommended for readers to refer to [124] for more details.

The eFPR is calculated as follows [124]

$$eFPR = R_{FP,c} + a_{CT}\frac{1}{|C|-1}\sum_{\substack{\hat{c}\in C\\ \hat{c}\neq c}} R_{CT,c,\hat{c}} \tag{11}$$

where $R_{FP,c}$ is the FP rate, a_{CT} is the weighting parameter, C being a set of sound classes, and $R_{CT,c,\hat{c}}$ represents the cross trigger rate.

On the other hand, eTPR is calculated as follow [124]

$$eTPR = \mu_{TP}(e) - a_{ST} * \sigma_{TP}(e) \tag{12}$$

where $\mu_{TP}(e)$ and $\sigma_{TP}(e)$ are the mean and standard deviation of TP ratios across classes respectively. a_{ST} is a weighting parameter.

Finally, PSDS is defined as the normalized area under the PSD-ROC curve, $r(e)$, given as [124].

$$PSDS = \frac{1}{e_{\max}}\int_0^{e_{\max}} r(e)\,de \tag{13}$$

where $e_{\max}$ is the maximum eFPR value of interest for the SED system under evaluation.

TABLE 15. Accuracy of proposed architecture utilizing weakly labeled data on different dataset.

References	Dataset	Event Based F1-Score (%)	Event Based ER	Segment-Based F1-score	Segment-Based ER
[102]	DCASE 2017 Task 4 Development Dataset			1 Second: 52.1	1 Second: 0.66
	DCASE 2017 Task 4 Evaluation Dataset			1 Second: 55.5	1 Second: 0.66
[103]	DCASE 2017 Task 4 Development Dataset			1 Second: 49.7	1 Second: 0.72
	DCASE 2017 Task 4 Evaluation Dataset			1 Second: 51.8	1 Second: 0.73
[97]	DCASE 2017 Task 4 Development Dataset			1 Second: 52.4	1 Second: 0.75
	DCASE 2017 Task 4 Evaluation Dataset			1 Second: 57.3	1 Second: 0.75
[111]	DCASE 2018 Task 4 Development Dataset	34.4	1.12		
	DCASE 2018 Task 4 Evaluation Dataset	32.4			
[114]	DCASE 2018 Task 4 Evaluation Dataset	39.5			
[117]	DCASE 2019 Task 4 Development Dataset	45.4		1 Second: 69.02	
	DCASE 2019 Task 4 Evaluation Dataset	42.7		1 Second: 71.4	
[106]	DCASE 2018 Task 4 Development Dataset	30.1	1.36		
	DCASE 2018 Task 4 Evaluation Dataset	25.4	1.19		
[107]	DCASE 2018 Task 4 Development Dataset	34.8			
	DCASE 2018 Task 4 Evaluation Dataset	26.2			

TABLE 16. TUT-SED 2016 dataset.

Residential Area		Home	
Event Class	Instances	Event Class	Instances
(Object) Banging	23	(Object) Rustling	60
Bird Singing	271	(Object) Snapping	57
Car Passing By	108	Cupboard	40
Children Shouting	31	Cutlery	76
People Speaking	52	Dishes	151
People Walking	44	Drawer	51
Wind Blowing	30	Glass Jingling	36
		Object Impact	250
		People Walking	54
		Washing Dishes	84
		Water Tap Running	47

TABLE 17. TUT-SED 2017 dataset.

Event Class	Instances	
	Development Dataset	Evaluation Dataset
Brake Squeaking	52	23
Car	304	106
Children	44	15
Large Vehicle	61	24
People Speaking	89	37
People Walking	109	42
Total	659	247

TABLE 18. DCASE 2017 dataset.

Categories	Sound Event	Instances
Warning Sound	Train Horn	441
	Air Horn, Truck Horn	407
	Car Alarm	273
	Reversing Beeps	337
	Ambulance (Siren)	624
	Police Car (Siren)	2399
	Fire Engine, Fire Truck (Siren)	2399
	Civil Defense (Siren)	1506
	Screaming	744
Vehicle Sound	Bicycle	2020
	Skateboard	1617
	Car	25744
	Car Passing By	3724
	Bus	3745
	Truck	7090
	Motorcycle	3291
	Train	2301

F. DISCUSSION

In this section, several openly available datasets are presented. Although TUT-SED 2016 and TUT-SED 2017 are strongly labeled, the sizes of these datasets are considered small and limited. On the other hand, the sizes of DCASE 2017 and DCASE 2018 are much larger but are weakly labeled. Therefore, these few datasets can be used for different tasks. However, as recommended in [123], the event-based metric should be used as the primary measure to evaluate the performance and capabilities of any models proposed. This is because event-based metrics better illustrate the ability to correctly locate and label longer blocks of audio.

TABLE 19. DCASE 2018 dataset.

Sound Event	Instances
Speech	550
Dog	214
Cat	173
Alarm Bell Ringing	205
Dishes	184
Frying	171
Blender	134
Running Water	343
Vacuum Cleaner	167
Electric Shaver/Toothbrush	103
Total	2244

On the other hand, PSD-ROC and PSDS are relatively new evaluation metrics, thus it may not be sufficient or comprehensive enough to report these two metrics without event-based or segment based metrics.

V. POSSIBLE FUTURE RESEARCH DIRECTIONS

As seen in the previous sections, even though CRNN is the new state of the art for SED, there is still a large room for improvement. However, Artificial Intelligence (AI) and Machine Learning (ML) are progressing at a faster rate as compared to the earlier days, possibly due to the hardware advancement or the induction of new and useful learning algorithms. In the following subsections, several concepts that may be integrated into SED are discussed. The purpose of this section is to introduce the general idea of the concepts and their respective limitations and how it may help in the SED domain.

A. CAPSULE NEURAL NETWORK

As mentioned earlier, the introduction of CapsNet is to overcome some limitations of CNN, in particular, the loss of information due to max-pooling operator [81].

As mentioned earlier, inter-capsules are connected through a process known as dynamic routing. This can be viewed as a parallel attention mechanism that allows each capsule at one level to attend to some active capsules at the level below and to ignore others. Such a process is hypothesized to allow the model to recognize multiple objects in the image even if objects overlap [125]. The issue of overlapping can be seen in SED where multiple events occur at the same time [80]. Therefore, the use of CapsNet may very well address the overlapping issues and as demonstrated in [80], the use of CapsNet can allow the model to achieve the lowest ER as compared to different architectures on different datasets. However, research on capsules is still at its infancy stage [83], [125].

For example, the iterative routing procedure is inefficient and if the number of capsules in any layer becomes too large, the routing procedure becomes computationally intractable [126], [127]. To preserve the spatial information, the original Capsnet only uses shallow CNN. However, due to the absence of deep semantic information, it can suffer from limited robustness on the classification task of complex datasets [128]. Finally, there are still open questions about specific aspects of the network implementation mainly due to the lack of thorough recommendations on how to design CapsNet [83].

Therefore, the success shown in [80] may not be at its peak at this moment due to the aforementioned issues. However, with the rising interest in CapsNet, there are various modifications proposed in the literature to improve the original CapsNet.

Zhao *et al.* [129] proposed an adaptive optimizer to enhance the reliability of routing as well as capsule compression and partial routing to improve the scalability of capsule networks. Such modifications were found to yield competitive results on Natural Language Processing tasks. On the other hand, Duarte *et al.* [126] proposed a new voting procedure for convolutional capsule layers to reduce the number of computations used in capsule routing thus alleviating the intractable computational resource issue. Whereas Rawlinson *et al.* [130], found that unsupervised sparsening of latent capsule layer activity appears to generalize better than supervised masking and has the potential to enable deeper capsules networks.

It is believed that with the various modifications and improvements proposed in the literature, CapsNet can reach a new horizon which eventually becomes a new standard for SED. However, the current state of CapsNet research can be said to be on the same level of advancement as CNNs were in 1998 [83]. Therefore, it will probably require a lot more small insights before it can out-perform a highly developed technology [125].

B. GENERATIVE ADVERSARIAL NETWORK

GAN was originally proposed by Goodfellow *et al.* [131]. The idea of GAN is to train two NNs to pit against each other where one is known as the generative model while the other is known as the discriminative model. The task of a generative model is to produce a virtual sample. While the task of a discriminative model is to learn and determine whether a sample is from the model distribution or the data distribution. The training of a GAN usually continues until virtual samples generated cannot be distinguished from the genuine data sample [132]. This idea can be used to increase or rebalance the sample size of the training data and was proven to be beneficial in [93].

A common way of application is to apply GAN on an image like time-frequency representation, i.e. spectrograms [133] to generate new samples. On the other hand, it is also possible to generate new samples based on raw waveforms [133]. In an experiment [133], it was found that human judges preferred samples generated using raw waveforms as opposed to samples made using time-frequency representations based on the criteria of sound quality and speaker diversity. Thus, it may be of interest to investigate the use of raw audio

waveforms to generate new samples or even the use of both audio waveforms and time-frequency representations to create new samples.

However, one key question remains, how does one measure the quality of generated audio samples? Image can be evaluated intuitively but audios are often difficult to evaluate subjectively [134].

The most straightforward evaluation method is the use of human judgment. Besides the additional time and effort to annotate the samples, another downside of using human annotators is that the metric varies depending on the setup of the task [135].

Thus, Salimans *et al.* [135] proposed the use of an inception score where the idea was to use a pre-trained inception model to measure both the diversity and semantic discriminability of generated samples. Although Salimans *et al.* [135] found that the use of the inception score correlates well with human judgment, it has two flaws where a poor generative model can achieve a high inception score [133]. Firstly, a generative model that produces a single example of each class with uniform probability will be assigned a high score [133]. Secondly, a generative model that overfits the training data will achieve a high score simply by producing samples on which the classifier was trained [133].

To mitigate the flaws of the inception score, Donahue *et al.* [133] proposed to use the average Euclidean distance as the secondary measure. Donahue *et al.* [133] suggested calculating the average Euclidean distance of a set of examples to their nearest neighbors within the set (other than itself) where a higher value indicates a higher diversity amongst samples. In addition, Donahue *et al.* [133] also suggested calculating the average Euclidean distance of examples to their nearest neighbor in the real training data. Therefore, if the generative model simply produces examples from the training set, this measure will be 0 that allows easier elimination of poor samples.

Thus, it is recommended that generated samples be evaluated through several different measures before integrating into any model to prevent the inclusion of sub-par samples.

C. BLIND SOURCE SEPARATION

The combination of a SED system with a source separation algorithm may also be an interesting area to investigate. Blind Source Separation (BSS) can be defined as a problem where both the sources and the mixing methodology are unknown and only mixture signals are available for the further separation process [136]. In different scenarios, it may be desirable to recover all individual sources from the mixed-signal, or at least to segregate a particular source [136].

Therefore, source separation can also be used as a pre-processing step where it can either be used to separate the audio signal into the foreground and background sound. Otherwise, it can also be used to separate the audio mixture into roughly homogenous spectral before performing SED. This was demonstrated in [46], where the detection accuracy was almost doubled.

On the other hand, detection accuracy may also be enhanced by aggregating the output of a SED system with a source separation model. In the DCASE 2020 Task 4 challenge, the baseline mean teacher model [137] was integrated with a source separation algorithm [138]–[140] where the idea was to average the outputs from a SED system and a source separation model. Based on such an idea, the event-based F1-score of the baseline system was increased by a total of 0.8%.

However, the extent of how sound separation can improve a SED system is still uncertain at this point. The number of audio sources presented in an audio mixture is often not known. Moreover, without the availability of any directional cues, the extraction of each audio source can be exceptionally difficult [138].

These problems may be mitigated or overcome by applying BSS algorithm as a method to provide an approximated temporal labels for the weakly labeled data. Such an idea was applied in the previous work of the authors [101] where NMF was applied to provide approximated temporal labels to the weakly labeled data rather than using it as a BSS algorithm. Results have shown that it was able to achieve a much higher event-based F1-score as compared to the baseline system.

Therefore, the idea of integrating a SED system with a source separation algorithm may be an interesting area to investigate that may bring unexpected improvement to a SED system.

VI. CONCLUSION

In this paper, different Sound Event Detection (SED) methodologies proposed in the literature were reviewed and discussed in detail. Although the state of the art CRNN can raise the bar to a new level, it should be noted that there is still a large room for improvement. As seen in the previous sections, most of the authors reported their model accuracy using segment-based evaluation. But the proposed methodologies performed poorly in terms of event-based evaluation even though they had a notable performance in terms of segment-based evaluations [106], [127]. Despite several strategies to reduce the reliance on strongly labeled data, it is obvious that the model using weakly labeled data is not as effective as its counterpart using strongly labeled data. Thus, this remains an open research area as to how weakly labeled data can be utilized effectively. Finally, this paper also provided some possible future trends which may be useful for future developments and may lead the development of SED system to a new horizon.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their pointers which greatly improve the quality of this article

REFERENCES

[1] M. C. Brown and J. S. Sacchi, "Audition," in *Fundamental Neuroscience*, L. Squire, D. Berg, F. E. Bloom, Sascha du Lac, A. Ghosh, and N. C. Spitzer, Eds., 4th ed. Cambridge, MA, USA: Academic, 2012, pp. 553–574.

[2] A. Bregman, *Auditory Scene Analysis: The Perceptual Organization of Sound*. London, U.K.: MIT Press, 1990.

[3] Q. Kong, Y. Xu, I. Sobieraj, W. Wang, and M. D. Plumbley, "Sound event detection and time–frequency segmentation from weakly labelled data," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 27, no. 4, pp. 777–787, Apr. 2019.

[4] Q. Zhou, Z. Feng, and E. Benetos, "Adaptive noise reduction for sound event detection using subband-weighted NMF," *Sensors*, vol. 19, no. 3206, pp. 1–19, Jul. 2019.

[5] N. Cho and E.-K. Kim, "Enhanced voice activity detection using acoustic event detection and classification," *IEEE Trans. Consum. Electron.*, vol. 57, no. 1, pp. 196–202, Feb. 2011.

[6] E.-L. Benaroya and G. Peeters, "A generic classification system for multichannel audio indexing: Application to speech and music detection," in *Proc. 14th Int. Workshop Image Anal. Multimedia Interact. Services (WIAMIS)*, Paris, France, Jul. 2013, pp. 1–4.

[7] N. C. Phuong and T. D. Dat, "Sound classification for event detection: Application into medical telemonitoring," in *Proc. Int. Conf. Comput., Manage. Telecommun. (ComManTel)*, Ho Chi Minh City, Vietnam, Jan. 2013, pp. 330–333.

[8] C. Clavel, T. Ehrette, and G. Richard, "Events detection for an audio-based surveillance system," in *Proc. IEEE Int. Conf. Multimedia Expo*, Amsterdam, The Netherlands, Jul. 2005, pp. 1–4.

[9] M. Chaudhary, V. Prakash, and N. Kumari, "Identification vehicle movement detection in forest area using MFCC and KNN," in *Proc. Int. Conf. Syst. Modeling Advancement Res. Trends (SMART)*, Moradabad, India, Nov. 2018, pp. 158–164.

[10] T. K. Chan and C. S. Chin, "Health stages diagnostics of underwater thruster using sound features with imbalanced dataset," *Neural Comput. Appl.*, vol. 31, pp. 5767–5782, Oct. 2019.

[11] S. Grollmisch, J. Abeber, J. Liebetrau, and H. Lukashevich, "Sounding industry: Challenges and datasets for industrial sound analysis," in *Proc. 27th Eur. Signal Process. Conf. (EUSIPCO)*, A Coruna, Spain, Sep. 2019, pp. 1–5.

[12] J. Florentin, T. Dutoit, and O. Verlinden, "Identification of European woodpecker species in audio recordings from their drumming rolls," *Ecol. Informat.*, vol. 35, pp. 61–70, Sep. 2016.

[13] Z. Zhao, S.-H. Zhang, Z.-Y. Xu, K. Bellisario, N.-H. Dai, H. Omrani, and B. C. Pijanowski, "Automated bird acoustic event detection and robust species classification," *Ecol. Informat.*, vol. 39, pp. 99–108, May 2017.

[14] E. Cakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 25, no. 6, pp. 1291–1303, Jun. 2017.

[15] Y. Liu, J. Tang, Y. Song, and L. Dai, "A capsule based approach for polyphonic sound event detection," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Honolulu, HI, USA, Nov. 2018, pp. 1853–1857.

[16] T. Hayashi, S. Watanabe, T. Toda, T. Hori, J. Le Roux, and K. Takeda, "BLSTM-HMM hybrid system combined with sound activity detection network for polyphonic sound event detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, New Orleans, LA, USA, Mar. 2017, pp. 766–770.

[17] T. Hayashi, S. Watanabe, T. Toda, T. Hori, J. Le Roux, and K. Takeda, "Duration-controlled LSTM for polyphonic sound event detection," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 25, no. 11, pp. 2059–2070, Nov. 2017.

[18] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Shanghai, China, Mar. 2016, pp. 6440–6444.

[19] J. Schroder, N. Moritz, J. Anemuller, S. Goetze, and B. Kollmeier, "Classifier architectures for acoustic scenes and events: Implications for DNNs, TDNNs, and perceptual features from DCASE 2016," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 25, no. 6, pp. 1304–1314, Jun. 2017.

[20] A. Dang, T. H. Vu, and J.-C. Wang, "A survey of deep learning for polyphonic sound event detection," in *Proc. Int. Conf. Orange Technol. (ICOT)*, Singapore, Dec. 2017, pp. 75–78.

[21] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, "Context-dependent sound event detection," *EURASIP J. Audio, Speech, Music Process.*, vol. 2013, no. 1, pp. 1–13, Jan. 2013.

[22] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Trans. Multimedia*, vol. 17, no. 10, pp. 1733–1746, Oct. 2015.

[23] G. Lafay, E. Benetos, and M. Lagrange, "Sound event detection in synthetic audio: Analysis of the dcase 2016 task results," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust. (WASPAA)*, New Paltz, NY, USA, Oct. 2017, pp. 11–15.

[24] A. Mesaros, T. Heittola, E. Benetos, P. Foster, M. Lagrange, T. Virtanen, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: Outcome of the DCASE 2016 challenge," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 26, no. 2, pp. 379–393, Feb. 2018.

[25] A. Mesaros, A. Diment, B. Elizalde, T. Heittola, E. Vincent, B. Raj, and T. Virtanen, "Sound event detection in the DCASE 2017 challenge," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 27, no. 6, pp. 992–1006, Jun. 2019.

[26] R. Serizel and N. Turpault, "Sound event detection from partially annotated data: Trends and challenges," in *Proc. IcETRAN Conf.*, Srebrno Jezero, Serbia, Jun. 2019, pp. 1–11.

[27] R. Serizel, N. Turpault, A. Shah, and J. Salamon, "Sound event detection in synthetic domestic environments," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 86–90.

[28] X. Xia, R. Togneri, F. Sohel, Y. Zhao, and D. Huang, "A survey: Neural network-based deep learning for acoustic event detection," *Circuits, Syst., Signal Process.*, vol. 38, no. 8, pp. 3433–3453, Mar. 2019.

[29] M.-Q. Bui, V.-H. Duong, S. Mathulaprangsan, B.-T. Pham, W.-J. Lee, and J.-C. Wang, "A survey of polyphonic sound event detection based on non-negative matrix factorization," in *Proc. Int. Comput. Symp. (ICS)*, Chiayi, Taiwan, Dec. 2016, pp. 351–354.

[30] J. Rex, "Audio event detection and localisation for directing video surveillance–a survey of potential techniques," in *Proc. 4th Int. Conf. Imag. Crime Detection Prevention (ICDP)*, London, U.K., 2011, pp. 1–4.

[31] S. Chandrakala and S. L. Jayalakshmi, "Environmental audio scene and sound event recognition for autonomous surveillance: A survey and comparative studies," *ACM Comput. Surv.*, vol. 52, no. 3, pp. 1–34, Jul. 2019.

[32] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.-Y. Chang, and T. Sainath, "Deep learning for audio signal processing," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 2, pp. 206–219, May 2019.

[33] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen, "Sound event localization and detection of overlapping sources using convolutional recurrent neural networks," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 1, pp. 34–48, Mar. 2019.

[34] A. Mesaros, S. Adavanne, A. Politis, T. Heittola, and T. Virtanen, "Joint measurement of localization and detection of sound events," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust. (WASPAA)*, New York, NY, USA, Oct. 2019, pp. 333–337.

[35] Y. Cao, Q. Kong, T. Iqbal, F. An, W. Wang, and M. D. Plumbley, "Polyphonic sound event detection and localization using a two-stage strategy," in *Proc. Detection Classification Acoust. Scenes Events Workshop*, New York, NY, USA, Oct. 2019, pp. 30–34.

[36] Y. Koizumi, S. Saito, H. Uematsu, N. Harada, and K. Imoto, "ToyADMOS: A dataset of miniature-machine operating sounds for anomalous sound detection," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust. (WASPAA)*, New York, NY, USA, Oct. 2019, pp. 313–317.

[37] H. Purohit, R. Tanabe, T. Ichige, T. Endo, Y. Nikaido, K. Suefusa, and Y. Kawaguchi, "MIMII dataset: Sound dataset for malfunctioning industrial machine investigation and inspection," in *Proc. Detection Classification Acoust. Scenes Events Workshop (DCASE)*, New York, NY, USA, Oct. 2019, pp. 209–213.

[38] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[39] J. M. Baker, L. Deng, J. Glass, S. Khudanpur, C. H. Lee, N. Morgan, and D. O'Shaughnessy, "Research developments and directions in speech recognition and understanding, part 1," *IEEE Signal Process. Mag.*, vol. 26, no. 3, pp. 75–80, May 2009.

[40] P. Mayorga, D. Ibarra, V. Zeljkovic, and C. Druzgalski, "Quartiles and mel frequency cepstral coefficients vectors in hidden Markov-Gaussian mixture models classification of merged heart sounds and lung sounds signals," in *Proc. Int. Conf. High Perform. Comput. Simulation (HPCS)*, Amsterdam, The Netherlands, Jul. 2015, pp. 298–304.

[41] L. Shi, I. Ahmad, Y. He, and K. Chang, "Hidden Markov model based drone sound recognition using MFCC technique in practical noisy environments," *J. Commun. Netw.*, vol. 20, no. 5, pp. 509–518, Oct. 2018.

[42] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[43] M. Rajapakse and L. Wyse, "Generic audio classification using a hybrid model based on GMMs and HMMs," in *Proc. 11th Int. Multimedia Model. Conf.*, Melbourne, VIC, Australia, Jan. 2005, pp. 1–6.

[44] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic event detection in real life recordings," in *Proc. 18th Eur. Signal Process. Conf. (EUSIPCO)*, Aalborg, Denmark, Aug. 2010, pp. 1267–1271.

[45] A. Diment, T. Heittola, and T. Virtanen, "Sound event detection for office live and office synthetic AASP challenge," in *Proc. IEEE AASP Challenge Detection Classification Acoust. Scenes Events*, Mar. 2013, pp. 1–3. [Online]. Available: http://c4dm.eecs.qmul.ac.uk/sceneseventschallenge/abstracts/OL/DHV.pdf

[46] T. Heittola, A. Mesaros, T. Virtanen, and M. Gabbouj, "Supervised model training for overlapping sound events based on unsupervised source separation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Vancouver, BC, Canada, May 2013, pp. 8677–8681.

[47] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, Oct. 1999.

[48] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, New Paltz, NY, USA, Oct. 2003, pp. 177–180.

[49] A. Mesaros, T. Heittola, O. Dikmen, and T. Virtanen, "Sound event detection in real life recordings using coupled matrix factorization of spectral representations and class activity annotations," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Brisbane, QLD, Australia, Apr. 2015, pp. 151–155.

[50] V. Bisot, S. Essid, and G. Richard, "Overlapping sound event detection with supervised nonnegative matrix factorization," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, New Orleans, LA, USA, Mar. 2017, pp. 31–35.

[51] Y. Ohishi, D. Mochihashi, T. Matsui, M. Nakano, H. Kameoka, T. Izumitani, and K. Kashino, "Bayesian semi-supervised audio event transcription based on Markov Indian buffet process," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Vancouver, BC, Canada, May 2013, pp. 3163–3167.

[52] P. Orbanz and Y. W. Teh, "Bayesian nonparametric model," in *Encyclopedia of Machine Learning and Data Mining*, C. Sammut and G. I. Webb, Eds. Berlin, Germany: Springer, 2017, pp. 1–14.

[53] S. J. Gershman and D. M. Blei, "A tutorial on Bayesian nonparametric models," *J. Math. Psychol.*, vol. 56, no. 1, pp. 1–12, Feb. 2012.

[54] K. Nakamura and K. Nakadai, "Robot audition based acoustic event identification using a Bayesian model considering spectral and temporal uncertainties," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Hamburg, Germany, Sep. 2015, pp. 4840–4845.

[55] N. Takahashi, M. Gygli, and L. Van Gool, "AENet: Learning deep audio features for video analysis," *IEEE Trans. Multimedia*, vol. 20, no. 3, pp. 513–524, Mar. 2018.

[56] G. Ferroni, R. Bonfigli, E. Principi, S. Squartini, and F. Piazza, "A deep neural network approach for voice activity detection in multi-room domestic scenarios," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Killarney, Ireland, Jul. 2015, pp. 1–8.

[57] D. Snyder, D. Garcia-Romero, and D. Povey, "Time delay deep neural network-based universal background models for speaker recognition," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand. (ASRU)*, Scottsdale, AZ, USA, Dec. 2015, pp. 92–97.

[58] O. Gencoglu, T. Virtanen, and H. Huttunen, "Recognition of acoustic events using deep neural networks," in *Proc. 22nd Eur. Signal Process. Conf. (EUSIPCO)*, Lisbon, Portugal, Sept. 2014, pp. 506–510.

[59] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *Proc. IEEE 25th Int. Workshop Mach. Learn. Signal Process. (MLSP)*, Boston, MA, USA, Sep. 2015, pp. 1–6.

[60] J. Du, Y. Tu, Y. Xu, L. Dai, and C.-H. Lee, "Speech separation of a target speaker based on deep neural networks," in *Proc. 12th Int. Conf. Signal Process. (ICSP)*, Hangzhou, China, Oct. 2014, pp. 473–477.

[61] X.-L. Zhang and D. Wang, "A deep ensemble learning method for monaural speech separation," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 5, pp. 967–977, May 2016.

[62] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multi label deep neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Killarney, Ireland, Jul. 2015, pp. 1–7.

[63] I. Goodfellow, D. W. Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," in *Proc. 30th Int. Conf. Mach. Learn.*, Atlanta, GA, USA, Jun. 2013, pp. 1319–1327.

[64] P. Swietojanski, J. Li, and J.-T. Huang, "Investigation of maxout networks for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Florence, Italy, May 2014, pp. 7649–7653.

[65] P. Swietojanski and S. Renals, "Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, South Lake Tahoe, NV, USA, Dec. 2014, pp. 171–176.

[66] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Multi-label vs. Combined single-label sound event detection with deep neural networks," in *Proc. 23rd Eur. Signal Process. Conf. (EUSIPCO)*, Nice, France, Aug. 2015, pp. 2551–2555.

[67] G. Castaneda, P. Morris, and T. M. Khoshgoftaar, "Evaluation of maxout activations in deep learning across several big data domains," *J. Big Data*, vol. 6, no. 1, pp. 1–35, Aug. 2019.

[68] M. Cai, Y. Shi, and J. Liu, "Stochastic pooling maxout networks for low-resource speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Florence, Italy, May 2014, pp. 3266–3270.

[69] L. Tóth, "Phone recognition with hierarchical convolutional deep maxout networks," *EURASIP J. Audio, Speech, Music Process.*, vol. 2015, pp. 1–13, Sep. 2015.

[70] S. Renals and P. Swietojanski, "Neural networks for distant speech recognition," in *Proc. 4th Joint Workshop Hands-Free Speech Commun. Microphone Arrays (HSCMA)*, Villers-les-Nancy, France, May 2014, pp. 172–176.

[71] A. Graves, N. Jaitly, and A.-R. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand.*, Olomouc, Czech, Dec. 2013, pp. 273–278.

[72] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Vancouver, BC, Canada, May 2013, pp. 6645–6649.

[73] T. Hayashi, S. Watanabe, T. Toda, T. Hori, J. L. Roux, and K. Takeda, "Bidirectional LSTM-HMM hybrid system for polyphonic sound event detection," in *Proc. Detection Classification Acoust. Scenes Events Challenge*, Sep. 2016, pp. 35–39. [Online]. Available: http://dcase.community/documents/challenge2016/technical_reports/DCASE2016_Hayashi_2006.pdf

[74] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[75] M. Zöhrer and F. Pernkopf, "Virtual adversarial training and data augmentation for acoustic event detection with gated recurrent neural networks," in *Proc. Interspeech*, Stockholm, Sweden, Aug. 2017, pp. 493–497.

[76] A. Zeyer, P. Doetsch, P. Voigtlaender, R. Schluter, and H. Ney, "A comprehensive study of deep bidirectional LSTM RNNS for acoustic modeling in speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, New Orleans, LA, USA, Mar. 2017, pp. 2462–2466.

[77] E. Martin and C. Cundy, "Parallelizing linear recurrent neural nets over sequence length," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, Vancouver, BC, Canada, May 2018, pp. 1–9.

[78] S. Adavanne, G. Parascandolo, P. Pertila, T. Heittola, and T. Virtanen, "Sound event detection in multichannel audio using spatial and harmonic features," in *Proc. Detection Classification Acoust. Scenes Events Workshop*, Budapest, Hungary, Sept. 2016, pp. 1–5.

[79] X. Xia, R. Togneri, F. Sohel, and D. Huang, "Confidence based acoustic event detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Calgary, AB, Canada, Apr. 2018, pp. 306–310.

[80] F. Vesperini, L. Gabrielli, E. Principi, and S. Squartini, "Polyphonic sound event detection by using capsule neural networks," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 2, pp. 310–322, May 2019.

[81] M. K. Patrick, B. Y. Edward, A. F. Adekoya, and A. A. Mighty, "Capsule networks—A survey," *J. King Saud Univ.-Comput. Inf. Sci.*, to be published, doi: 10.1016/j.jksuci.2019.09.014.

[82] G. Hinton, S. Sabour, and N. Frosst, "Matrix capsules with EM routing," in *Proc. Int. Conf. Learning Represent.*, Vancouver, BC, Canada, May 2018, pp. 1–15.

[83] R. Mukhometzianov and J. Carrillo, "CapsNet comparative performance evaluation for image classification," 2018, *arXiv:1805.11195*. [Online]. Available: http://arxiv.org/abs/1805.11195

[84] X. Jiang, Y. Wang, W. Liu, S. Li, and J. Liu, "CapsNet, CNN, FCN: Comparative performance evaluation for image classification," *Int. J. Mach. Learn. Comput.*, vol. 9, no. 6, pp. 840–848, Dec. 2019.

[85] E. Okewu, P. Adewole, and O. Sennaike, "Experimental comparison of stochastic optimizers in deep learning," in *Proc. Int. Conf. Comput. Sci. Appl.*, Saint Petersburg, Russia, Jul. 2019, pp. 704–715.

[86] S. Kim, S. Park, S. Lim, and D. Kim, "Classification performance analysis of weight update method applied to various convnet models," in *Proc. Int. Conf. Control Robots (ICCR)*, Hong Kong, Sep. 2018, pp. 78–83.

[87] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Process. Lett.*, vol. 24, no. 3, pp. 279–283, Mar. 2017.

[88] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 1724–1734.

[89] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *Proc. 28th Conf. Neural Inf. Process. Syst. (NIPS) Deep Learn.*, Montreal, QC, Canada, Dec. 2014, pp. 1–9.

[90] S. Jung, J. Park, and S. Lee, "Polyphonic sound event detection using convolutional bidirectional LSTM and synthetic data-based transfer learning," in *Proc. ICASSP-IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Brighton, U.K., May 2019, pp. 885–889.

[91] S. Adavanne, P. Pertila, and T. Virtanen, "Sound event detection using spatial features and convolutional recurrent neural network," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, New Orleans, LA, USA, Mar. 2017, pp. 771–775.

[92] S. Adavanne, A. Politis, and T. Virtanen, "Multichannel sound event detection using 3D convolutional neural networks for learning inter-channel features," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Rio de Janeiro, Brazil, Jul. 2018, pp. 1–7.

[93] X. Xia, R. Togneri, F. Sohel, and D. Huang, "Auxiliary classifier generative adversarial network with soft labels in imbalanced acoustic event detection," *IEEE Trans. Multimedia*, vol. 21, no. 6, pp. 1359–1371, Jun. 2019.

[94] M. Gong, Y. Xu, C. Li, K. Zhang, and K. Batmanghelich, "Twin auxiliary classifiers GAN," in *Proc. 33rd Conf. Neural Inf. Process. Syst. (NeurIPS)*, Vancouver, BC, Canada, Dec. 2019, pp. 1–10.

[95] W. Ding and L. He, "Adaptive multi-scale detection of acoustic events," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 28, pp. 294–306, Nov. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8897625

[96] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, Oct. 2016, pp. 483–499.

[97] Q. Kong, Y. Xu, W. Wang, and M. D. Plumbley, "Sound event detection of weakly labelled data with CNN-transformer and automatic threshold optimization," 2019, *arXiv:1912.04761*. [Online]. Available: http://arxiv.org/abs/1912.04761

[98] B. Kim and B. Pardo, "Sound event detection using point-labeled data," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust. (WASPAA)*, New Paltz, NY, USA, Oct. 2019, pp. 1–5.

[99] B. McFee, J. Salamon, and J. P. Bello, "Adaptive pooling operators for weakly labeled sound event detection," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 26, no. 11, pp. 2180–2193, Nov. 2018.

[100] Q. Kong, Y. Xu, W. Wang, and M. D. Plumbley, "A joint separation-classification model for sound event detection of weakly labelled data," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Calgary, AB, Canada, Apr. 2018, pp. 321–325.

[101] T. K. Chan, C. S. Chin, and Y. Li, "Non-negative matrix factorization-convolutional neural network (NMF-CNN) for sound event detection," in *Proc. Detection Classification Acoust. Scenes Events Workshop (DCASE)*, New York, NY, USA, Oct. 2019, pp. 40–44.

[102] D. Lee, S. Lee, Y. Han, and K. Lee, "Ensemble of convolutional neural networks for weakly-supervised sound event detection using multiple scale input," in *Proc. Detection Classification Acoust. Scenes Events Workshop (DCASE)*, Munich, Germany, Nov. 2017, pp. 1–6.

[103] Y. Xu, Q. Kong, W. Wang, and M. D. Plumbley, "Large-scale weakly supervised audio classification using gated convolutional neural network," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Calgary, AB, Canada, Apr. 2018, pp. 121–125.

[104] Y. Xu, Q. Kong, W. Wang, and M. D. Plumbley, "Surrey-CVSSP system for DCASE2017 challenge task 4," in *Proc. IEEE AASP Challenge Detection Classification Acoust. Scenes Events*, Nov. 2017, pp. 1–3.

[105] Y. Hou, Q. Kong, J. Wang, and S. Li, "Polyphonic audio tagging with sequentially labelled data using CRNN with learnable gated linear units," in *Proc. Detection Classification Acoust. Scenes Events Workshop*, Surrey, U.K., Nov. 2018, pp. 1–5.

[106] S. Kothinti, K. Imoto, D. Chakrabarty, G. Sell, S. Watanabe, and M. Elhilali, "Joint acoustic and class inference for weakly supervised sound event detection," in *Proc. ICASSP-IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Brighton, U.K., May 2019, pp. 36–40.

[107] T. Pellegrini and L. Cances, "Cosine-similarity penalty to discriminate sound classes in weakly-supervised sound event detection," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Budapest, Hungary, Jul. 2019, pp. 1–8.

[108] Y. Wang, J. Li, and F. Metze, "A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling," in *Proc. ICASSP-IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Brighton, U.K., May 2019, pp. 31–35.

[109] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang, S. Watanabe, T. Yoshimura, and W. Zhang, "A comparative study on transformer vs RNN in speech applications," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop (ASRU)*, Singapore, Dec. 2019, pp. 449–456.

[110] B. Zhang, D. Xiong, and J. Su, "Accelerating neural transformer via an average attention network," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, Melbourne, VIC, Australia, Jul. 2018, pp. 1789–1798.

[111] J. Lu, "Mean teacher convolution system for DCASE 2018 task 4," in *Proc. IEEE AASP Challenge on Detection Classification Acoust. Scenes Events*, Jul. 2018, pp. 1–5.

[112] A. Miech, I. Laptev, and J. Sivic, "Learnable pooling with context gating for video classification," 2017, *arXiv:1706.06905*. [Online]. Available: http://arxiv.org/abs/1706.06905

[113] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Proc. 31st Conf. Neural Inf. Process. Syst. (NIPS)*, Long Beach, CA, USA, Dec. 2017, pp. 1–10.

[114] L. Lin, X. Wang, H. Liu, and Y. Qian, "Guided learning for weakly-labeled semi-supervised sound event detection," 2019, *arXiv:1906.02517*. [Online]. Available: http://arxiv.org/abs/1906.02517

[115] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Proc. NIPS Deep Learn. Represent. Learn. Workshop*, Montreal, QC, Canada, Dec. 2015, pp. 1–9.

[116] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 4320–4328.

[117] L. Lin, X. Wang, H. Liu, and Y. Qian, "Guided learning convolution system for DCASE 2019 task 4," in *Proc. Detection Classification Acoust. Scenes Events Workshop (DCASE)*, New York, NY, USA, Oct. 2019, pp. 134–138.

[118] V. Mnih, H. Larochelle, and G. E. Hinton, "Conditional restricted Boltzmann machines for structured output prediction," in *Proc. 27th Conf. Uncertainty Artif. Intell. (UAI)*, Barcelona, Spain, Jul. 2011, pp. 514–522.

[119] M. A. C. Perpinan and G. E. Hinton, "On contrastive divergence learning," in *Proc. 10th Int. Workshop Artif. Intell. Statist. (AISTATS)*, Bridgetown, Barbados, Jan. 2005, pp. 33–40.

[120] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *Proc. 24th Eur. Signal Process. Conf. (EUSIPCO)*, Budapest, Hungary, Aug. 2016, pp. 1–5.

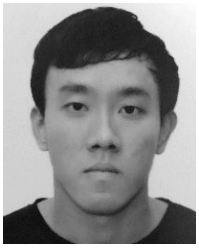
[121] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "DCASE 2017 challenge setup: Tasks, datasets and baseline system," in *Proc. Detection Classification Acoust. Scenes Events*, Munich, Germany, Nov. 2017, pp. 1–8.

[122] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, New Orleans, LA, USA, Mar. 2017, pp. 776–780.

[123] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Appl. Sci.*, vol. 6, no. 162, pp. 1–17, May 2016.

[124] C. Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulovic, "A framework for the robust evaluation of sound event detection," in *Proc. ICASSP-IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Barcelona, Spain, May 2020, pp. 61–65.

[125] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. 31st Conf. Neural Inf. Process. Syst. (NIPS)*, Long Beach, CA, USA, Dec. 2017, pp. 1–11.
[126] K. Duarte, Y. S. Rawat, and M. Shah, "VideoCapsuleNet: A simplified network for action detection," in *Proc. 32nd Conf. Neural Inf. Process. Syst. (NeurIPS)*, Montreal, QC, Canada, Dec. 2018, pp. 1–10.
[127] A. R. Kosiorek, S. Sabour, Y. W. Teh, and G. E. Hinton, "Stacked capsule autoencoders," in *Proc. 33rd Conf. Neural Inf. Process. Syst. (NeurIPS)*, Vancouver, BC, Canada, Dec. 2019, pp. 1–14.
[128] C. Xiang, L. Zhang, Y. Tang, W. Zou, and C. Xu, "MS-capsnet: A novel multi-scale capsule network," *IEEE Signal Process. Lett.*, vol. 25, no. 12, pp. 1850–1854, Dec. 2018.
[129] W. Zhao, H. Peng, S. Eger, E. Cambria, and M. Yang, "Towards scalable and reliable capsule networks for challenging NLP applications," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, Florence, Italy, 2019, pp. 1549–1559.
[130] D. Rawlinson, A. Ahmed, and G. Kowadlo, "Sparse unsupervised capsules generalize better," 2018, *arXiv:1804.06094*. [Online]. Available: https://arxiv.org/abs/1804.06094
[131] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2014, pp. 2672–2680.
[132] A. Pankajakshan, H. L. Bear, and E. Benetos, "Polyphonic sound event and sound activity detection: A multi-task approach," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust. (WASPAA)*, New Paltz, NY, USA, Oct. 2019, pp. 323–327.
[133] C. Donahue, J. McAuley, and M. Puckette, "Adversarial audio synthesis," in *Proc. 7th Int. Conf. Learn. Represent. (ICLR)*, New Orleans, LA, USA, May 2019, pp. 1–16.
[134] L. Yang, D. Jiang, and H. Sahli, "Feature augmenting networks for improving depression severity estimation from speech signals," *IEEE Access*, vol. 8, pp. 24033–24045, Feb. 2020.
[135] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Barcelona, Spain, Dec. 2016, pp. 1–10.
[136] M. Pal, R. Roy, J. Basu, and M. S. Bepari, "Blind source separation: A review and analysis," in *Proc. Int. Conf. Oriental COCOSDA Held Jointly Conf. Asian Spoken Lang. Res. Eval. (O-COCOSDA/CASLRE)*, Gurgaon, India, Nov. 2013, pp. 1–5.
[137] L. Delphin-Poulat and C. Plapous, "Mean teacher with data augmentation for DCASE 2019 task 4," in *Proc. Detection Classification Acoust. Scenes Events Challenge*, Jun. 2019, pp. 1–3. [Online]. Available: http://dcase.community/documents/challenge2019/technical_reports/DCASE2019_Delphin_15.pdf
[138] I. Kavalerov, S. Wisdom, H. Erdogan, B. Patton, K. Wilson, J. Le Roux, and J. R. Hershey, "Universal sound separation," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust. (WASPAA)*, New York, NY, USA, Oct. 2019, pp. 175–179.
[139] E. Tzinis, S. Wisdom, J. R. Hershey, A. Jansen, and D. P. W. Ellis, "Improving universal sound separation using sound classification," in *Proc. ICASSP-IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Barcelona, Spain, May 2020, pp. 96–100.
[140] S. Wisdom, J. R. Hershey, K. Wilson, J. Thorpe, M. Chinen, B. Patton, and R. A. Saurous, "Differentiable consistency constraints for improved deep speech enhancement," in *Proc. ICASSP-IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 900–904.



T. K. CHAN received the B.Eng. and M.Phil. degrees from Newcastle University upon Tyne, in 2015 and 2018, respectively. He is currently pursuing the Ph.D. degree through the Singapore Industrial Postgraduate Program with Xylem Water Solutions Singapore Private Ltd., and Newcastle University upon Tyne. He is also a Data Engineer with Xylem Water Solutions Singapore Private Ltd., includes investigation of advanced information fusion methods to coordinate the deployed large-scale heterogeneous sensor network and development of models to achieve a reliable, robust leak detection, and localization performance. His current research interests include water leakage detection and localization algorithms, signal processing, machine learning, and handling of imbalanced dataset.

CHENG SIONG CHIN (Senior Member, IEEE) received the B.Eng. degree (Hons.) in mechanical and aerospace engineering from Nanyang Technological University (NTU), in 2000, the M.Sc. degree (Hons.) in advanced control and systems engineering from The University of Manchester, in 2001, and the Ph.D. degree in applied control engineering from the Research Robotics Centre, NTU, in 2008.

He was with the consumer electronics industry for few years before moving into academia. He is currently a Reader (Associate Professor) and the Director of Innovation and Engagement with Newcastle University, Singapore. He holds over three U.S. Patents in the area of electronics test systems and components. He is the author/coauthor of nearly 100 peer-reviewed papers on journals and conference proceedings. He was invited as a Plenary Speaker and the Chief Guest of the IEEE International Conference on Power, Energy control and Transmission Systems, in 2018. He received seven research grants from the Singapore Maritime Institute (SMI) and the Economic Development Board-Industrial Postgraduate Program (EDB-IPP), Singapore, grants in his research areas. His research interests include design and simulation of complex systems, such as acoustic systems, underwater robotics vehicles, energy storage systems, and contactless sensing systems. He is an Elected Vice-Chairman of the IEEE Oceanic Engineering Society in Singapore Section. He is the general chair and the technical review committee member in various international conferences (ICRAI 2019) related to intelligent systems. He serves as an Associate Editor for the *IEEE Earthzine (Marine Power and Battery Systems)* and *Electronics (Systems and Control Engineering)*.