





# Robust and Interpretable Temporal Convolution Network for Event Detection in Lung Sound Recordings

Tharindu Fernando , Member, IEEE, Sridha Sridharan , Life Senior Member, IEEE, Simon Denman , Member, IEEE, Houman Ghaemmaghami, and Clinton Fookes , Senior Member, IEEE

## I. INTRODUCTION

**Abstract—Objective:** This paper proposes a novel framework for lung sound event detection, segmenting continuous lung sound recordings into discrete events and performing recognition of each event. **Methods:** We propose the use of a multi-branch TCN architecture and exploit a novel fusion strategy to combine the resultant features from these branches. This not only allows the network to retain the most salient information across different temporal granularities and disregards irrelevant information, but also allows our network to process recordings of arbitrary length. **Results:** The proposed method is evaluated on multiple public and in-house benchmarks, containing irregular and noisy recordings of the respiratory auscultation process for the identification of auscultation events including inhalation, crackles, and rhonchi. Moreover, we provide an end-to-end model interpretation pipeline. **Conclusion:** Our analysis of different feature fusion strategies shows that the proposed feature concatenation method leads to better suppression of non-informative features, which drastically reduces the classifier overhead resulting in a robust lightweight network. **Significance:** Lung sound event detection is a primary diagnostic step for numerous respiratory diseases. The proposed method provides a cost-effective and efficient alternative to exhaustive manual segmentation, and provides more accurate segmentation than existing methods. The end-to-end model interpretability helps to build the required trust in the system for use in clinical settings.

**Index Terms**—Lung sound event detection, adventitious sound, temporal convolution networks, respiratory monitor, electronic stethoscope, segmentation.

Manuscript received May 19, 2021; revised January 5, 2022; accepted January 10, 2022. Date of publication January 21, 2022; date of current version July 4, 2022. This work was supported by the Cooperative Research Centre Projects (CRC-P) Grant. (Corresponding author: Tharindu Fernando.)

Tharindu Fernando, Sridha Sridharan, Simon Denman, and Clinton Fookes are with Signal Processing, Artificial Intelligence, and Vision Technologies (SAIVT), Queensland University of Technology, Brisbane, Queensland 4000, Australia (e-mail: t.warnakulasuriya@qut.edu.au; s.sridharan@qut.edu.au; s.denman@qut.edu.au; c.fookes@qut.edu.au).

Houman Ghaemmaghami is with M3DICINE Pty Ltd., Brisbane, Queensland 4113, Australia (e-mail: houman@m3dicine.com).

This article has supplementary downloadable material available at <http://doi.org/10.1109/JBHI.2022.3144314>, provided by the authors.

Digital Object Identifier 10.1109/JBHI.2022.3144314

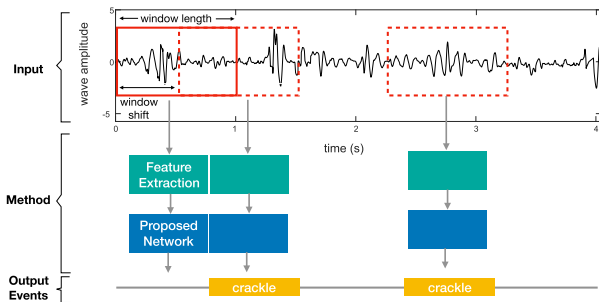
RESPIRATORY auscultation [1]–[3], where lung sounds are captured using a stethoscope placed on a patient's chest or back, is a primary diagnostic step for respiratory diseases. In addition to inferring breathing cycles (inhalation and exhalation), numerous works [2]–[4] have shown how respiratory auscultation can be utilised for disease diagnosis.

In [5] the authors illustrate how lung sounds can be broadly categorised into two classes: normal breathing sounds; and adventitious breathing sounds. Adventitious breathing sounds usually indicate the presence of pulmonary diseases, which can be either continuous (ex: wheeze, stridor, rhonchi) or discontinuous (ex: crackles). However, in addition to simply identifying the presence of adventitious sound, the temporal location of the adventitious sound within the patient's respiratory cycle carries clinical significance. For instance, [6] shows that fine-crackles occurring mid-to-late within the inhalation stage are indicative of interstitial lung fibrosis, congestive heart failure and pneumonia; while coarse-crackles during the early stages of inhalation and throughout exhalation are linked to chronic bronchitis.

Despite the fact that human practitioner based diagnosis of respiratory auscultation has been practiced for several hundred years, there exists numerous drawbacks and limitations to this process. Apart from the major limitations of the interpretation of observations being prone to human error and the experience of the practitioner, long-term monitoring of a single patient's recordings or comparisons between different patients is practically infeasible [2]. This motivates the need for an automated lung sound event detection framework where continuous lung sound recordings can be segmented into discrete events.

In spite of this need, only a handful of studies [2], [4], [5], [7], [8] have explored the use of deep learning techniques for event detection in lung sounds [2], and the majority of the literature [1], [9]–[12] has simply focused on distinguishing healthy participants from abnormal patients. Furthermore, existing lung event detection frameworks exhibit a lack of generalisation due to data scarcity. Their performance is limited to a small number of diseases/event types, and they have only been evaluated using a single dataset.

In addressing these research gaps we propose a lightweight, yet robust, simplified and completely interpretable framework



**Fig. 1.** Proposed Event Detection Framework: We apply a sliding window of size 1 sec with 50% overlap from the input lung sound recording. Extracted windows are passed through the feature extractor and then through the proposed network which outputs the event category for the window.

for lung sound event detection. We observe that recurrent neural networks such as Long Short-Term Memory (LSTM) [13] networks, Gated recurrent units (GRUs) [14], and attention based frameworks [15] are a popular choice when modelling temporal relationships within the auscultation recordings. Deviating from this common norm and seeking to develop a lightweight, simplified architecture, we utilise Temporal Convolution Networks (TCNs) [16] as the building block of the proposed framework.

Through our extensive evaluations we illustrate that a multi-branch TCN architecture can discover temporal relationships at different granularities, which can be missed by attention based recurrent architectures. Moreover, we illustrate how these pieces of identified salient information can be effectively fused such that a lightweight classifier can generate informed decisions without being overwhelmed by a multitude of information.

The proposed framework is illustrated in Fig. 1. We extract consecutive overlapping windows from lung recordings and pass these through our feature extraction pipeline. The resultant features are processed by the proposed network, which outputs the event category of the observed input window. Utilisation of model interpretation strategies to interrogate the decision-making process is overlooked in existing lung auscultation literature. To address this, we propose a novel strategy to interpret the operations of all components of the proposed architecture. Our interpretation strategy provides an end-to-end interpretation pipeline. We evaluate our system on three challenging datasets (2 public and one in-house) for the identification of numerous auscultation events including, inhalation, exhalation, crackles, wheeze, stridor, and rhonchi, and achieve state-of-the-art results in all evaluations. The main contributions of this work can be summarised as follows:

- We propose a novel lung event detection framework which is simple and light-weight in its design, but is capable of achieving state-of-the-art results on multiple challenging benchmarks for the detection of event categories including, inhalation, exhalation, crackles, wheeze, stridor, and rhonchi.
- We identify the importance of the multi-branch TCN architecture and propose an effective way to combine information from these branches.

- We demonstrate how end-to-end interpretations can be generated for the proposed framework, allowing a practitioner to understand what features within the input are leading to the final classification decision. Such interpretations increase the trust in the system, potentially enabling its clinical use in hospital settings.

## II. RELATED WORK

Since its advent, deep learning has revolutionised the field of machine learning and numerous works have exploited the computational power of deep learning for the detection of abnormality in lung sound recordings. For instance, in [17] the authors investigate the use of a CNN-SVM hybrid architecture for patient wise classification of abnormalities in lung sound recordings. Similarly in [1] the authors utilise a CNN network coupled with a Mixture of Experts (MoE) as the classifier. Specifically the MoE block contains a series of classifiers and a gating mechanism to automatically decide which classifier should be applied to which input region. The works [12], [18] use 2D CNN architectures for abnormality detection, while the authors of [11] use Deep Belief Networks as their classification framework and [19] explore the use of a CNN-RNN hybrid model. Despite these extensive investigations, these works have limited practical applications as they only focus on distinguishing healthy participants from abnormal patients. The utility of these methods, and the potential diagnostic outcomes, can be improved by temporally localising the events within the lung sound recordings [2], which we refer to as “lung sound event detection”.

Considering the literature on lung sound event detection, in [8] the authors adapt the Faster R-CNN [20] framework for lung event detection using a spectrogram representation of the lung recording. Specifically, the region proposal network of the Faster R-CNN provides proposals for inhale and exhale breathing phase locations in the spectrogram and the classifier network assigns the relevant event labels to those proposals. However, considering the large trainable parameter space of the Faster R-CNN the authors could not train the network using lung data from scratch. Hence, they utilised a pre-trained ResNet101 [21] architecture as the Faster R-CNN backbone network. Furthermore, they have performed post-processing on the network predictions based on author defined assumptions.

Another ResNet based transfer learning model is proposed in [5]. Similar to [8] the authors utilise a pre-trained ResNet101 to extract features from the input spectrogram. However, an attention based LSTM pipeline is used to generate event labels. The spectrograms of consecutive windows are sequentially passed through the network, which are encoded using the ResNet. By using the embedding of the current spectrogram and the previous embeddings, the LSTM then generates the event label of the current input.

The use of Recurrent Neural Network (RNN) is also proposed in [4]. The authors use a bidirectional GRU model which operates directly on the extracted Mel Frequency Cepstral Coefficients (MFCCs) and Spectrogram features. The authors simultaneously predict the breathing phase (i.e inhale/exhale) and the presence/absence of Crackle. However, their evaluations are

limited to a private in-house dataset and their model can only detect one category of adventitious sounds.

A comprehensive evaluation of different off the shelf deep learning architectures is presented in [2]. The authors have benchmarked different RNNs (including LSTM and GRU) and CNN-RNN hybrids (including CNN-LSTM, CNN-GRU, CNN-BiLSTM, and CNN-BiGRU) for the lung sound event detection task. Their evaluations revealed that the CNN-RNN hybrids perform slightly better compared to RNN only methods. But the authors have acknowledged that their performance is inconsistent across numerous event categories. Most importantly, they observe that the performance is heavily influenced by the smaller and imbalanced datasets which are available for the lung sound event detection task. This motivates the need for a lightweight model which can learn from limited data, but have sufficient capacity to model temporal relationships across different time intervals. Furthermore, the authors of [2] have observed that unusual breathing patterns such as shallow/fast breathing and different breathing rates in the recordings, contribute to the model's erroneous predictions. This clearly illustrates the need to capture temporal relationships within the input at different granularities. For instance, if the model is sequentially processing the input time-series (such as in LSTMs and GRUs) temporal relationships that exist across longer timescales can be missed [22], [23].

Leveraging these observations and motivated by the limitations of the existing literature, we propose a lightweight yet robust deep learning framework for lung sound event detection. We adapt lightweight Temporal Convolution Networks in place of parameter heavy RNN models and CNN-RNN hybrids. Despite being a popular choice in event detection [5], [15], we observe that soft-attention mechanisms dramatically increase the parameter space of the network, and hence are not suitable when learning from limited data. To this end we propose an effective feature fusion strategy which propagates only the most salient information to the classifier.

Methods that seek to segment smart-phone based recorded lung-sound recordings are also relevant, yet such recordings often contain significantly more noise compared to respiratory sounds collected via stethoscopes. In [24] the authors propose a novel virtual reality-based framework to encourage subjects regulate their breathing patterns. Specifically, they fine-tuned a voice activity detection method to differentiate the silence and breathing phases in the audio recording, and conduct a Mel-Frequency analysis to separate the inhalation and exhalation segments in the detected breathing phases. The authors have evaluated the proposed framework using 125 participants and their system achieves greater than 90% accuracy for the inhalation and exhalation segmentation task. This framework was extended in [25] where the authors have added an additional lung capacity estimation module to their virtual reality therapy framework. This allowed the system to simultaneously monitor the air intake in each breathing cycle and encourage the participants to take a deep breath if the previous inhalation was not sufficient. They authors demonstrate that the framework is beneficial, especially for lung cancer patients or subjects with breathing disorders. Similarly, in [26] the authors suggest the possibility of

utilising a smart-phone based setup for self-diagnosing respiratory diseases such as COVID-19 using breathing sounds recorded via a smart-phone. They illustrate the utility of such approach for higher risk groups such as patients with serious underlying medical conditions, or front-line workers, as a method for quick and frequent self-testing.

Despite the appealing nature of diagnostic methodologies that use a smart-phone microphone for data acquisition, to the best of our knowledge there are no publicly available datasets captured in this manner. Therefore, direct evaluation and comparison of systems developed for stethoscope based respiratory sounds using smart-phone microphone captured recordings is not feasible. However, we also note that with recent advances in digital stethoscope technology such as Stetsee,<sup>1</sup> it is possible to transmit stethoscope acquired lung sounds via Bluetooth to smartphones for processing. As such, the lightweight network that we have proposed is highly relevant for smartphone applications that use these such devices.

### III. METHODOLOGY

In this section we first provide a brief introduction to temporal convolutions, illustrating their limitations and motivating the need for a multi-branch Temporal Convolution Network (TCN) architecture. We then illustrate the proposed fusion strategy to fuse the information captured by the distinct branches and the details of the classifier which outputs the event labels.

#### A. Temporal Convolution

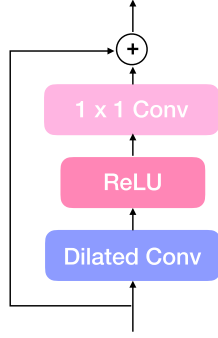
Recurrent architectures such as LSTMs and GRUs are outperformed by TCNs for numerous sequence modelling tasks [16], [27] due to their improved memory retention. Unlike LSTMs and their counterparts which maintain a fixed size memory, TCNs can attend to the entire sequence when extracting temporal relationships. Furthermore, the removal of recurrency leads to reduced computational cost. However, the effectiveness of TCNs relies on the amount of temporal connections that the kernel can capture. In this sub-section we illustrate the motivation behind the proposed multi-branch architecture.

Fig. 2(a) visualises the 1D convolution operation where a 3D tensor (of shape [batch size, seq length, input channels]) is converted to an output tensor of shape [batch size, seq length, output channels] using a 2D kernel. The right-hand side of Fig. 2(a) shows how the convolution output is generated for two consecutive elements in the output tensor, where the kernel is of size 3. The output is the dot product between the input elements within the window and the learned kernel weights. Then the kernel is shifted by one element to the right. Hence, in this case the receptive field size is 3. However, when modelling lengthier sequences we require a larger receptive field such that a particular output element is influenced by temporally distant input elements.

One way of achieving this is through stacking multiple 1D convolution layers. For instance, as shown in Fig. 2(b), by stacking 2 layers with a kernel size of 3 we can obtain a receptive

<sup>1</sup>[Online]. Available: <https://www.m3dicine.com/>





**Fig. 2.** Illustration of temporal convolution operation and dilation. (a) Input to output information propagation in 1D convolution with a kernel size of 3. (b) The receptive field size increases (to 5) when two 1D convolution layers are stacked. (c) Illustration of dilated convolutions with a kernel size of 3 and dilation factors,  $d$ , of 1, 2, and 4.

field size of 5. Another way of achieving this is through dilation. The distance between elements in the input sequence which is used to compute one entry in the output sequence is referred to as the dilation. Fig. 2(c) illustrates how dilation can be used to obtain a larger receptive field without the need to stack a large number of individual convolutions.

However, as illustrated in [16], [28], depending on the level of dilation and the number of convolution layers that are stacked together, there can be ‘holes’ in the receptive field such that elements in the input sequence are missed when generating the output sequence. To this end we propose a multi-branch network architecture which has different dilation bases (i.e 1, 2, 3, ...) and we exponentially increase the dilation value when moving up the layers in a particular branch. For example, when the dilation base is 2 the first layer has a dilation factor  $d = 2^0 = 1$ , the second layer  $d = 2^1 = 2$ , the third layer  $d = 2^2 = 4$ , etc. This allows different branches to capture temporal relationships at different granularities.

Following [28] we use dilation convolution layers with residual connections. Fig. 3 illustrates a residual block. Formally,

$$\hat{H}_l = \text{ReLU}(W_1 * H_{l-1} + b_1), \quad (1)$$

$$H_l = H_{l-1} + W_2 * \hat{H}_l + b_2, \quad (2)$$

where  $*$  denotes the convolution operation, and  $W_1$  and  $W_2$  are the weights of the dilated convolution layers (of kernel size 3) and weights of the second convolution layer with kernel size of 1, respectively.  $H_l$  is the output of layer  $l$ , and  $b_1$  and  $b_2$  are the biases.

## B. Multi-Branch TCN

As outlined earlier, we utilise a multi-branch network to capture different temporal granularities. Formally, let the input recording be denoted by the vector  $p$  which contains the audio frames from time instance 0 to  $T_{obs}$ ,

$$p = [X_0, \dots, X_{T_{obs}}], \quad (3)$$

and the feature extractor function,  $f$ , extracts features from this input such that,

$$s = f(p). \quad (4)$$

Then we pass the feature vector,  $s$ , through the individual encoders of the  $B$  branches which are composed of stacked TCN layers. Let the output  $o^b$  denote the output of the encoder of branch  $b \in [1, \dots, b, \dots, B]$  which consists of  $l \in [1, \dots, l, \dots, L]$  TCN layers, where the first layer has a dilation factor of  $d$ , and increases it exponentially from one layer to the next,

$$o^b = \text{Enc}^{b,l,d}(s). \quad (5)$$

## C. Classifier

The output,  $o^b$ , is a 3 dimensional tensor of shape [batch size, seq length, output dim]. In order to combine the outputs of the  $B$  branches we concatenate them on the 2nd dimension such that the combined tensor,  $\hat{o}$ , is of shape [batch size, seq length  $\times B$ , output dim]. Note that the dilation base and the dilation factors of different branches do not affect the output dimension of each branch. These parameters only effect the receptive field of each branch. Hence, all the breaches have the same output dimensions, [batch size, seq length, output dim], which allows us to directly concatenate them.

To enable the system to process input audio recordings/segments of variable lengths<sup>2</sup> we use a global average pooling operation to extract a fixed size 2D tensor from the concatenated output,  $\hat{o}$ . We note that global average pooling is performed separately for each feature dimension. Our multi-branch architecture generates different feature representations of the same input segment. Therefore, by concatenating features on the second dimension and pooling the features across this entire dimension, we force the branches to propagate only the most salient features to the classifier. Hence only the most important features across the  $B$  branches will be activated and the rest of the elements will be inactive. Therefore, the global average pooling operation will focus on these features, significantly reducing the overhead on the classifier. This allows us to use a simple, light-weight 3 layer multi-layer perceptron as our classifier. The overall architecture of the proposed method is visually illustrated in Fig. 4.

## IV. EXPERIMENTS

### A. Datasets

1) **HF Lung V1 Database [2]:** This dataset is captured from 279 patients and comprises 9,765 recordings, each of 15 sec duration and captured at 4 kHz. This is the largest publicly available lung sound recording dataset to date. The authors provide inhalation and (34,095) exhalation (18,349) labels, and label adventitious breathing sounds including 13,883 Continuous Adventitious Sounds (CAS) (8,457 wheezes, 686 stridor and 4,740 rhonchi events), and 15,606 Discontinuous Adventitious

<sup>2</sup>Note that while the system can process sequences of arbitrary lengths, in our evaluations we use 1 s windows.

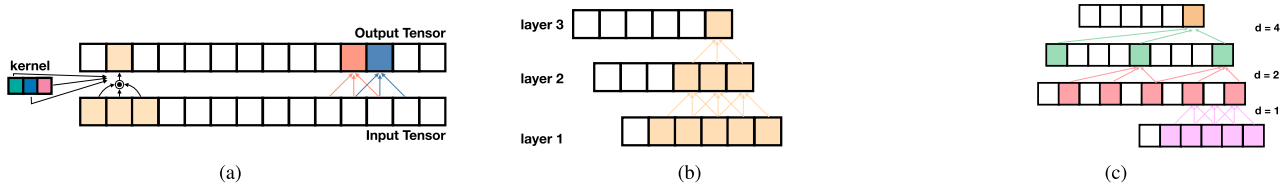


Fig. 3. Dilated residual layer utilised in the proposed work.

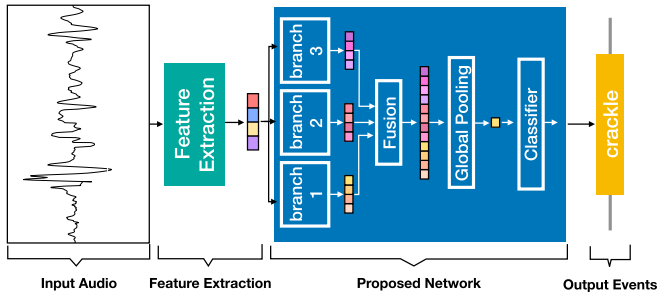


Fig. 4. Proposed model architecture: Once the features are extracted from the input lung sound recording they are passed through separate branches, which extract features at different temporal granularities. Resultant feature vectors are fused together using concatenation and global average pooling is applied to obtain a fixed-size 2D vector, which is subsequently passed through a classifier to classify the event category of the input window.

Sounds (DAS) (all crackles). Following the dataset authors we conducted separate experiments for inhalation, exhalation, CAS and DAS event detection. For training and testing we utilised the splits provided by the dataset authors, comprising 7,809 recordings for training and 1,956 for testing. As the validation set we used 10% of the training data.

**2) ICBHI Respiratory Sound Database [29]:** This dataset contains 920 annotated recordings with lengths varying from 10 to 90 sec, taken from 126 patients. The dataset has been annotated for respiratory cycles (6,898 cycles in total) and has 1,864 crackles, 886 wheezes and 506 events with both crackles and wheezes. Different recordings in this dataset have different sampling rates. 90 recordings are captured at 4 kHz; 6 at 10 kHz, and 824 are recorded at 44.1 kHz. It should be noted that this dataset replicates real life conditions with noisy recordings captured using different equipment (microphone, 3 M Littmann stethoscope and Meditron stethoscope), from different chest locations (7 locations in total), and spans different age groups (ranging from children, to adults and the elderly). As this dataset does not contain inhalation/exhalation annotations we utilise it only for the detection of adventitious breathing sounds. Specifically, we conducted experiments to detect crackles and wheezes. Following prior works [18], [30] we randomly split the data into training and testing, where 60% of the data is used for training and 40% for testing. We ensure that when sampling the recordings, samples from the same recording fall either to training or testing splits, not both. As the validation data we used 10% of the training set.

**3) M3DICINE Lung Sound Dataset:** The M3DICINE Lung Sound Dataset consists of 224 recordings collected from 78

patients captured at 4 kHz using M3DICINE's Stetsee<sup>3</sup> Pro 1 device. The recordings were collected from adults and children of different ethnic backgrounds and age groups. The recordings are 20 seconds in length, and are collected from the intersection of the mid-clavicular line and second intercostal space on the Right anterior surface, as well as posteriorly on the participant at a level lateral to T3. These locations were selected as they represent the locations of the Left Upper Lobe and Right Upper/Middle Lobes of the lungs, locations that are used routinely as part of respiratory clinical examinations and also in computer aided analysis of lung sounds. We randomly split the data into training and testing where 70% of the data is used for training and 30% for testing. 10% of the training data is used for validation. We ensure that when sampling the recordings, samples from the same recording fall either to training or testing splits, not both.

Note that in the ICBHI Respiratory Sound Database and M3DICINE Lung Sound Dataset we do not have access to patient level details, therefore, we cannot identify which recordings come from which patient. Hence, we cannot use a strict inter-patient division scheme. However, we treated each recording as a separate data sample and in our dataset splitting scheme we ensure that samples from the same recording fall either exclusively into either the training or testing split, and not both. As we are sampling multiple windows from every recording with overlap it is important to ensure such a division. Therefore, using this scheme we eradicate the chance of the model seeing part of a testing segment during training.

## B. Pre-Processing, Feature Extraction and Data Augmentation

Following [2], for all datasets we resampled recordings to 4 kHz and applied a high-pass filter with cut-off frequency of 80 Hz and a filter order of 10 to remove electrical interference and other background noise. We applied a Hamming window of size 4000 (1 sec) and a window shift of 2000 (.5 sec) to extract samples. This sampling strategy is recommended by the authors of [2]. As the label of the extracted window, we used the majority event label for the window. In situations where a window has an equal number of frames of both classes, our labelling algorithm will label the window as belonging to the first class. However, we note that this is a very rare occasion as we are conducting a binary classification with length 4000 segments.

As features, we extracted Mel Frequency Cepstral Coefficients (MFCCs) by obtaining 13 static, 13 delta and 13 delta-delta coefficients in the frequency range 0 – 2000Hz. In addition,

<sup>3</sup>[Online]. Available: <https://www.stetsee.com/>

we extracted the log mel filter bank energy feature of the window using 26 filters. All the features are concatenated together, yielding a feature vector of dimension 65. For all feature extraction processes we used a window length of 0.025 sec and a step size of 0.01 sec. Subsequently, min-max normalisation is applied to normalise the features to the range 0 and 1.

Considering the relatively small sizes of the utilised datasets, we employ data augmentation to artificially create new training samples to increase variation in training data and reduces overfitting on the limited training data. We used a random masking strategy to generate these artificial samples, where a random portion of the input is masked out. Specifically, we followed the data augmentation procedure of [31] and applied a frequency and time masking technique to randomly mask time slices and frequency bands with a probability of 0.5. We select time and frequency bands, each with 0.5 probability, and randomly select an index at which to start masking. For frequency masking, a window width between 2 and 8 is randomly selected; while for time masking we mask a random number of times slice between 5 and 10. We refer the reader to supplementary materials for further details regarding the masking process.

We would like to note that unlike [2], no post processing steps were applied in our framework. We threshold model predictions by setting values  $> 0.5$  to 1 and rest of the values to 0.

### C. Evaluation Metrics

We use the same evaluation protocol as the authors of [2]. We compare the predictions for each segment with their respective ground truth and evaluate the Jaccard similarity between the ground truth and predicted events. If the Jaccard similarity is greater than 0.5 we consider it to be a True Positive (TP), if similarity is between 0 and 0.5 we consider it a False Negative (FN) and if it is equal to zero we consider it a False Positive (FP). It should be noted that a True Negative (TN) event cannot be defined for this event detection task. We evaluated the Positive Predicted value (PPv), Sensitivity (Se) and F1 score using the following definitions,

$$PPv = \frac{TP}{TP + FP}, \quad (6)$$

$$Se = \frac{TP}{TP + FN}, \quad (7)$$

$$F_1 = 2 \frac{PPv \cdot Se}{PPv + Se}. \quad (8)$$

### D. Implementation Details

Implementation of the proposed model is completed using Pytorch (version 1.5.0) and CUDA (10.1), and the model is trained on an NVIDIA GForce RTX 2080 GPU.

The model contains number of branches,  $B$ , the number of layers,  $l$ , in each branch and the number of filters,  $k$ , of each convolution layer as hyper-parameters. These were evaluated experimentally and were set to  $b = 3$ ,  $l = 3$  and  $k = 80$ . Please refer to Section V for the evaluation details. The kernel size of all the convolution layers is set to 3 and dilation bases are set to 2, 3 and 4 for the 3 branches. The classifier network consist of three

TABLE I

EVALUATION RESULTS OF THE HF LUNG V1, ICBHI, AND M3DICINE LUNG DATASETS FOR THE DETECTION OF INHALATION, EXHALATION, CONTINUOUS ADVENTITIOUS SOUNDS (CAS), DISCONTINUOUS ADVENTITIOUS SOUNDS (DAS), CRACKLES AND WHEEZES EVENTS

Dataset	Task	Models	Results		
			PPv	Se	F1
HF Lung V1	Inhalation	LSTM [2]	0.890	0.664	0.761
		Bi-GRU [2]	0.898	0.800	0.862
		CNN-GRU [2]	0.906	0.742	0.820
		CNN-Bi GRU [2]	0.898	0.812	0.862
		Proposed	<b>0.998 <math>\pm 0.021</math></b>	<b>0.892 <math>\pm 0.015</math></b>	<b>0.942 <math>\pm 0.017</math></b>
	Exhalation	LSTM [2]	0.561	0.456	0.570
		Bi-GRU [2]	0.713	0.617	0.709
		CNN-GRU [2]	0.629	0.516	0.620
		CNN-Bi GRU [2]	0.693	0.600	0.685
		Proposed	<b>0.976 <math>\pm 0.011</math></b>	<b>0.729 <math>\pm 0.008</math></b>	<b>0.835 <math>\pm 0.032</math></b>
	CAS	LSTM [2]	0.120	0.095	0.122
		Bi-GRU [2]	0.237	0.227	0.256
		CNN-GRU [2]	0.556	0.402	0.498
		CNN-Bi GRU [2]	0.508	0.463	0.516
		Proposed	<b>0.895 <math>\pm 0.024</math></b>	<b>0.627 <math>\pm 0.053</math></b>	<b>0.737 <math>\pm 0.012</math></b>
	DAS	LSTM [2]	0.699	0.485	0.591
		Bi-GRU [2]	0.765	0.638	0.714
		CNN-GRU [2]	0.709	0.539	0.646
		CNN-Bi GRU [2]	0.718	0.626	0.700
		Proposed	<b>0.997 <math>\pm 0.032</math></b>	<b>0.787 <math>\pm 0.025</math></b>	<b>0.880 <math>\pm 0.034</math></b>
ICBHI	Crackles	LSTM [2]	0.646	0.507	0.568
		Bi-GRU [2]	0.667	0.556	0.606
		CNN-GRU [2]	0.646	0.507	0.568
		CNN-Bi GRU [2]	0.805	0.648	0.718
		Proposed	<b>0.938 <math>\pm 0.023</math></b>	<b>0.874 <math>\pm 0.031</math></b>	<b>0.905 <math>\pm 0.026</math></b>
	Wheezes	LSTM [2]	0.760	0.497	0.601
		Bi-GRU [2]	0.804	0.556	0.657
		CNN-GRU [2]	0.815	0.537	0.647
		CNN-Bi GRU [2]	0.792	0.518	0.627
		Proposed	<b>0.994 <math>\pm 0.035</math></b>	<b>0.858 <math>\pm 0.020</math></b>	<b>0.921 <math>\pm 0.019</math></b>
M3DICINE Lung	Exhalation	LSTM [2]	0.623	0.702	0.660
		Bi-GRU [2]	0.667	0.735	0.699
		CNN-GRU [2]	0.769	0.727	0.748
		CNN-Bi GRU [2]	0.710	0.831	0.766
		Proposed	<b>0.984 <math>\pm 0.031</math></b>	<b>0.940 <math>\pm 0.023</math></b>	<b>0.962 <math>\pm 0.020</math></b>

We report positive predicted value (PPv), sensitivity (Se) and F1 scores.

multi-layer perceptions with hidden dimensions, 80, 32 and 1, respectively. The proposed model was trained for 200 epochs using binary cross entropy as the objective and the Adam [32] optimiser, using mini-batch of size 64, and a learning rate of  $10^{-5}$ .

### E. Results

An evaluation of the proposed framework together with the current state-of-the-art results for the HF Lung V1, ICBHI, and M3dicine Lung datasets is provided in Table I. To the best of our knowledge, apart from [2] none of the existing works have evaluated their lung event detection methods on publicly available data. Therefore, we utilise the LSTM, BiGRU, CNN-GRU and CNN-Bi-GRU models proposed in [2] as our baselines for evaluations. As the implementations of [2] are not publicly available we carefully re-implemented these models to provide comparison results for the ICBHI and M3dicine Lung datasets (results on HF Lung V1 are taken from [2]). To illustrate the robustness of the models, we randomly initialised the network weights and ran each experiment 3 times, and report the average metric values along with the standard deviation.

Analysing the results, it is clear that the proposed model has been able to outperform the recurrent and CNN models by a significant margin. We would like to highlight the number of trainable parameters in the baseline models LSTM (300,609), Bi-GRU (552,769), CNN-GRU (2,605,249) and CNN-Bi GRU (5,240,513), whereas the proposed method only contains 276,225 trainable parameters. We would like to emphasise the fact that our lightweight model has been able to achieve significantly higher results compared to these baselines.

TABLE II

EFFECT OF NUMBER OF BRANCHES  $B$  ON THE OVERALL ACCURACY

Models	Results		
	PPV	Se	F1
1 Branch	0.937 $\pm$ 0.027	0.880 $\pm$ 0.045	0.908 $\pm$ 0.051
2 Branch	0.991 $\pm$ 0.036	0.877 $\pm$ 0.029	0.931 $\pm$ 0.038
3 Branch (Proposed)	<b>0.998 <math>\pm</math>0.034</b>	<b>0.905 <math>\pm</math>0.013</b>	<b>0.966 <math>\pm</math>0.025</b>
4 Branch	0.995 $\pm$ 0.033	0.889 $\pm$ 0.042	0.939 $\pm$ 0.038

The results are presented for the validation set of HF Lung V1 dataset for the Inhale detection task.

Furthermore, its performance is consistent across all considered event categories, while we observe significant fluctuations in the performance of the baselines. We believe this behaviour of the baselines results from the high number of trainable parameters, which leads to over-fitting (and poor generalisation) when trained using smaller datasets. In the following section we quantitatively and qualitatively analyse the contributions from each of the proposed innovations, which allow our model to achieve this level of performance.

## V. DISCUSSION

In this section we evaluate the contributions from the proposed multi-branch TCN architecture and the importance of the proposed feature fusion strategy. Moreover, we illustrate the proposed model interpretation algorithmic framework along with the visualisations for randomly selected samples.

### A. Importance of Multi-Branch Architecture

We evaluate the performance of three ablation models, with the number of branches set to 1, 2 and 4. Dilation bases for the networks are set to 2, (2, 3) and (2, 3, 4 and 5) respectively. These dilation bases are selected based on the recommendations in [28]. The ablation evaluation results along with the performance of the proposed method (using 3 branches) is presented in Table II. For the single branch model we simply take the global average across the time dimension. Note that evaluation results here are presented for the validation set of the HF Lung V1 dataset for the inhale detection task.

When analysing the accuracy gain between the single branch and multi-branch architectures, we observe the importance of considering dependencies across multiple temporal granularities. In particular, we observe that the performance gain is maximised in the 3 branch setting and slightly decreases when it is increased above 3. We believe this behaviour is due to the rapid increase in size of the concatenated feature vector, which could confuse the classifier when the information is redundant.

Moreover, we visualise the neuron conductance [33] of the 3 branches for 2 randomly chosen examples from the HF Lung V1 dataset. The conductance plot shows the flow of information from the input to a particular hidden unit, and helps us understand the importance of a hidden unit in generating a relevant output for a specific input. Specifically, the notion of conductance can be derived based on integrated gradients [33], where the integration (sum) over the gradients denotes the net difference between the

baseline predictions (i.e. near zero predictions) and the target prediction value for a particular input.

By aggregating it across all the hidden units of a particular layer we can identify which regions of the input are activated (i.e. conductance with respect to each input feature), and therefore the attributed output is of the same size as the input. Fig. 5 illustrates the neuron conductance of the last layers of the 3 branches of the proposed model for two examples, and demonstrates that the salient information identified by the 3 branches varies. Recall that each branch has different dilation bases, which allows them to see different temporal dependencies. Therefore, we see different input regions being activated by different neurons among these branches. Specifically, we see that specific temporal intervals and frequency levels are conducting information from different branches, denoting the importance of different dilation factors and highlighting their ability to capture different levels of the temporal relationships.

Similarly we evaluated the optimal number of layers,  $l$ , in each branch and the output dimension,  $k$ , of convolution layers, which are presented in Fig. 6(a) and (b), respectively. Note that for this evaluation we fixed  $B = 3$  and used the validation set of HF Lung V1 dataset for the inhale detection task. The plots show the F1 score.

### B. Importance of the Proposed Feature Fusion Mechanism

Next, we investigate the importance of the proposed feature fusion mechanism where we concatenate the features of the 3 branches on the time axis (i.e 2nd dimension), and apply global average pooling across this dimension to extract a fixed 2D tensor.

Firstly, we evaluate the performance of naive concatenation (i.e concatenating across the feature dimension, the 3rd dimension) and then taking global average pooling across the 2nd dimension. We evaluate two other attention based feature fusion mechanisms. In the first model we apply multi-head self-attention [34] to extract features from the naive concatenated tensor where the number of attention heads is set to 4. In the second model we use two levels of self-attention, where individual attention weights were applied to individual branches to extract salient features from those branches. Then these ‘enhanced features’ are concatenated together on the feature dimension (i.e 3rd dimension). Then we apply another level of attention to summarise this concatenated feature across the time axis (i.e convert it to a 2D tensor). The evaluations of these ablation models along with the proposed method for the HF Lung V1 dataset’s inhale detection task are presented in Table III.

We observe poor performance from both attention based ablation models as well as the naive concatenation mechanism. We believe this is due to the large amount of features present in the 3 branches, which confuses the classifier as the attention/naive concatenation mechanism fail to effectively filter out unnecessary information. Specifically, if our framework receives a 1 s window, then the input is  $99 \times 65$  (where 65 is the feature dimension). Each of the 3 branches outputs a  $99 \times 80$  feature vector. With naive concatenation this becomes a



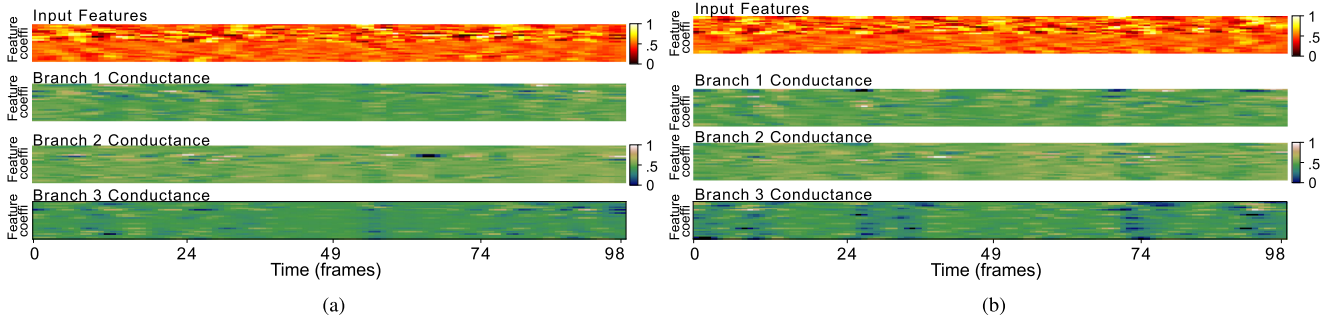


Fig. 5. Neuron Conductance [33] for two randomly chosen windows from the ‘steth\_20190801\_10\_54\_04.wav’ recording from the test set of the HF Lung V1 dataset. We visualise the input features along with the conductance of the last layers of the 3 branches. Note that to ensure clarity of the visualisation, we include only the first 26 features (i.e MFCC and Deltas) of our 65 dimensional feature.

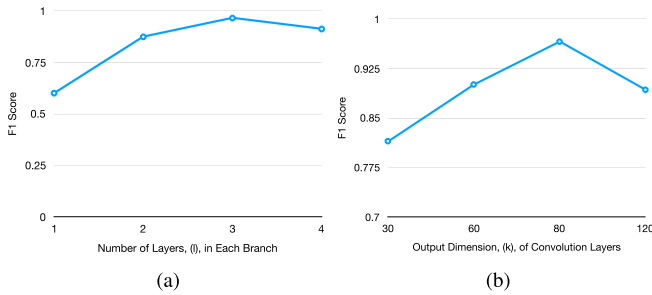


Fig. 6. Hyper-Parameter Evaluations: (a) Change in F1 score against number of layers,  $l$ , in each branch. (b) Change in F1 score against output dimension,  $k$ , of convolution layers. Results are for the validation set of HF Lung V1 dataset for the inhale detection task, using three branches in the network.

TABLE III

COMPARISON BETWEEN THE PROPOSED FEATURE FUSION MECHANISM, NAIVE CONCATENATION AND ATTENTION BASED FEATURE FUSION

Models	Results		
	PPv	Se	F1
naive concat	0.927 $\pm$ 0.036	0.861 $\pm$ 0.018	0.893 $\pm$ 0.026
Attention 1	0.936 $\pm$ 0.042	0.878 $\pm$ 0.027 $\pm$ 0.037	0.906 $\pm$ 0.031
Attention 2	0.938 $\pm$ 0.021	0.877 $\pm$ 0.056	0.906 $\pm$ 0.047
Proposed	<b>0.998 <math>\pm</math> 0.034</b>	<b>0.905 <math>\pm</math> 0.013</b>	<b>0.966 <math>\pm</math> 0.025</b>

The results are presented for the validation set of HF Lung V1 dataset for the Inhale detection task.

$99 \times 240$  feature vector as the feature of the 3 branches are concatenated vertically. Hence, the attention weights become larger. Despite the high capacity of attention based models we do not have sufficiently large dataset to effectively train such a model, leading to poor performance on unseen data. However, in the proposed fusion method we force the branches to pass only the most salient information as we pool across time  $\times$  number of branches when passing information to the classifier. Therefore, the classifier has less information to process, and less chance to get confused by redundant information. To better illustrate this we obtained activations of the concatenated output of the 3 branches along with the attribution of this feature (feature importance to the classifier) generated using the DeepLift [35] algorithm. Utilising DeepLift, we show of the “conducted” information through the branches, what information is important to the classifier at a particular timestep. Neural conductance

cannot produce this as it’s simply showing the information flow. Therefore, using DeepLift we try to interpret what changes in the concatenated features from the branches cause the classifier to change its predictions. Our main objective here is to illustrate that at different time instances, different features coming from different branches become salient to the classifier; i.e. the most relevant branch changes over time. DeepLift computes feature importance by accumulating the difference in the activation of the target neuron (i.e. classification layer) to its reference state. It assumes that the changes in this neuro-activation are caused due to the changes in the input (i.e. in our change the concatenated features of the 3 branches). Hence, we can measure the contribution scores of each feature at the point where the classifier changes its prediction from one class to the other.

The 5th row of Fig. 7(a) contains the heatmaps of the horizontally concatenated features of the 3 branches (i.e. prior to global average pooling). We did not visualise the feature map after global average pooling as it’s of dimension  $1 \times 80$  and is harder to visualise. The 6th row of Fig. 7(a) is the extracted portion of the important features from those shown in the 5th row (i.e. features retained by the classifier based on DeepLift). Similarly, the 5th and 6th rows of Fig. 7(b) contain the vertically concatenated (naive concatenation) features of the 3 branches and DeepLift based important features, respectively. In the visualisations presented in Fig. 7(a) we observe only a small portion of the concatenated tensor is activated and the pooling operation has suppressed most of the features, resulting in a simplified task for the classifier. In Fig. 7(b) we present the same visualisation for the attention model 2. Despite the self-attention based feature selection mechanism we observe that most of the features are activated, hence passing a myriad of information (much of which is redundant) to the classifier.

### C. End-to-End Interpretation Pipeline

As illustrated in previous subsections, by combining the interpretations generated by DeepLift [35] and Neuron Conductance [33] algorithms we can trace back to the time and frequency coefficients within the input feature representation that are salient for the event detection task, which can be used to



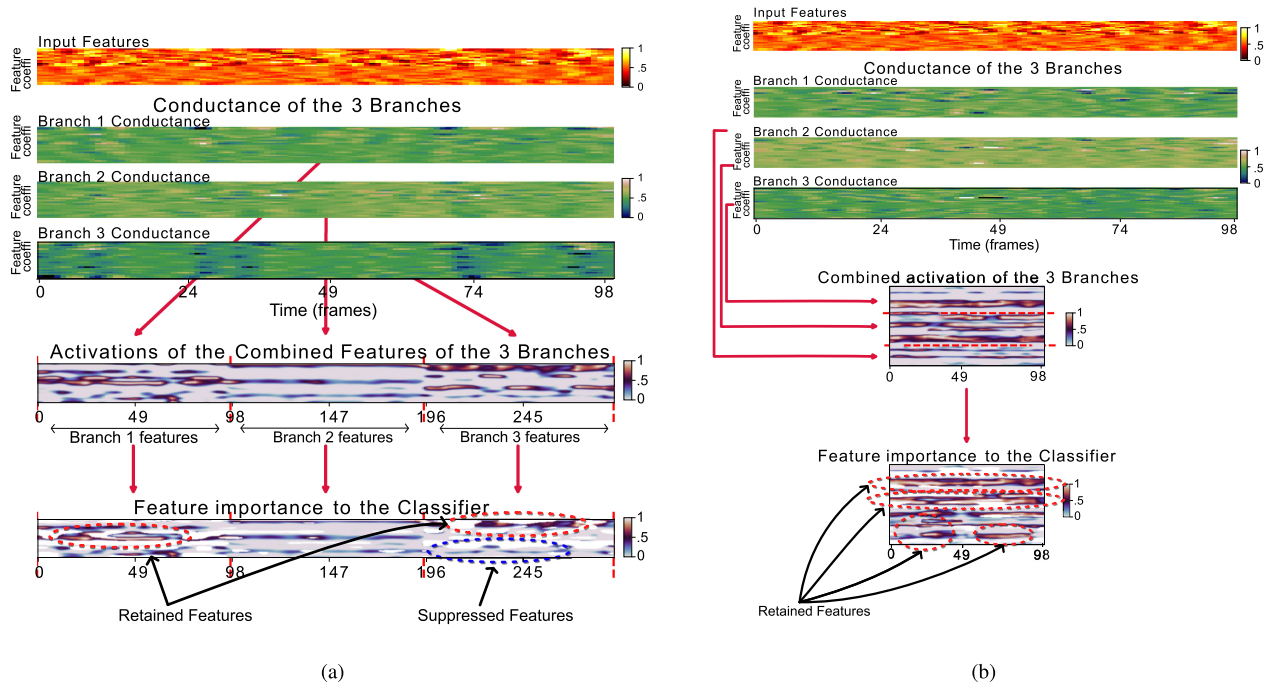


Fig. 7. The activation of the combined feature and its saliency to the classifier (generated using the DeepLift [35]) for a randomly chosen window of the recording 'steth\_20190801\_10\_54\_04.wav' from the test set of the HF Lung V1 dataset. In (a) we visualise the saliency for the proposed feature concatenation mechanism and in (b) for the attention based concatenation (attention method 2).

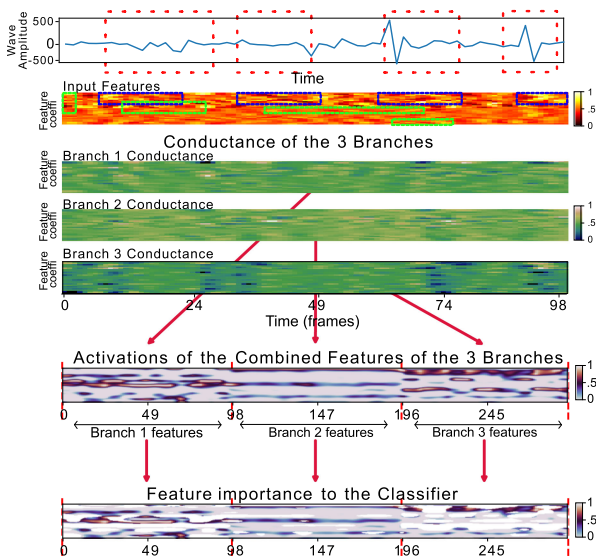


Fig. 8. Complete Interpretation generated for a randomly chosen window of the recording 'steth\_20190801\_10\_54\_04.wav' from the test set of the HF Lung V1 dataset.

identify the relevant respective time intervals in the input lung recording. In Fig. 8 we provide one such end-to-end interpretation generated for a randomly chosen window of the recording 'steth\_20190801\_10\_54\_04.wav' from the test set of the HF Lung V1 dataset. Note that this example is for the detection of CAS and we have highlighted the salient regions (in green: branch 1; and blue: branch 3) in the input feature based on its saliency to the classifier.

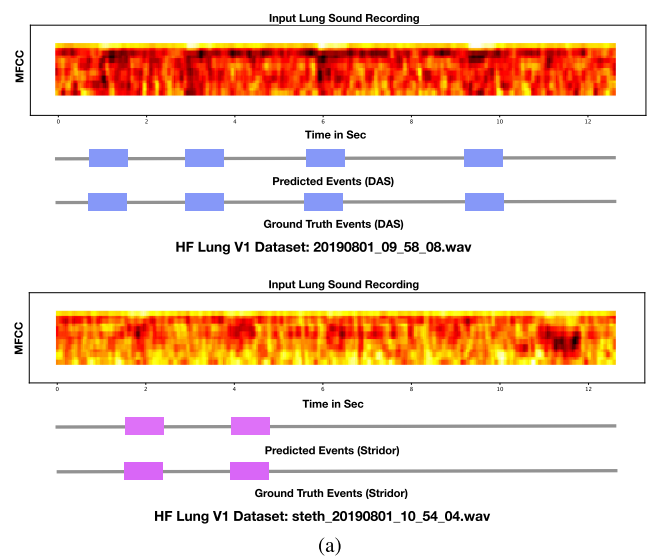


Fig. 9. Qualitative Results for HF Lung V1 dataset.

#### D. Qualitative Results

In Figs. 9, 10, 11, and 12 we show detected events from the proposed model along with the ground truth for randomly selected recordings from the HF Lung V1, ICBHI and M3dicine Lung datasets. Note that we are utilising 1 s windows as our input and there exists a 50% overlap between consecutive windows. Therefore, we are able to detect events within the 1 s window and there is only a 0.5 sec delay between the first occurrence of an event and the detection of this event by the proposed

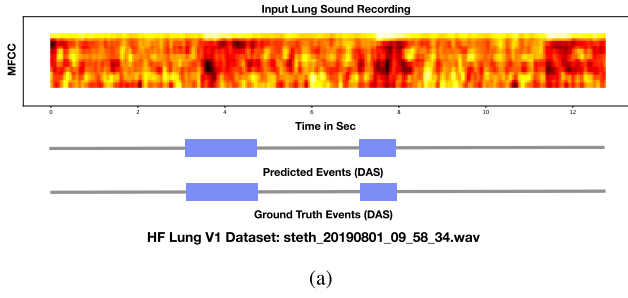


Fig. 10. Qualitative Results for HF Lung V1 dataset.

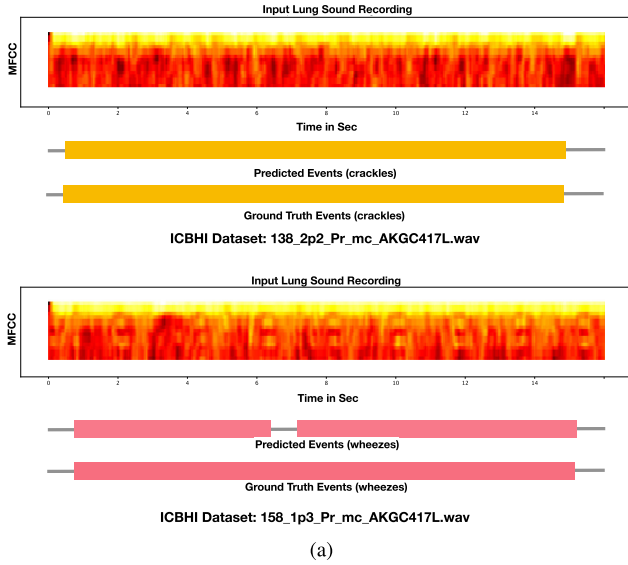


Fig. 11. Qualitative Results for ICBHI dataset.

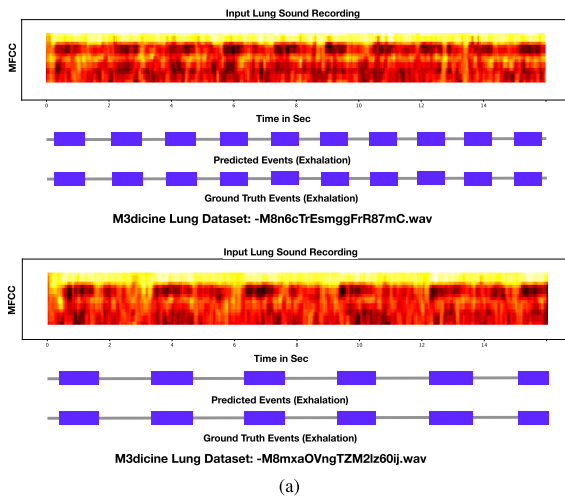


Fig. 12. Qualitative Results for M3dicine Lung dataset.

algorithm. For clarity of visualisations, note that in the MFCC we show only the first 13 dimensions of the feature (i.e. only the MFCC feature), and not the entire feature vector of used in the proposed model. Considering the visualisation, we observe that the proposed model attends to both spatial and temporal features

and achieves consistent performance across all considered event categories. We also refer the readers to the supplementary video file where we provide qualitative results and model interpretations for additional recordings.

### E. Time Complexity

As indicated earlier the proposed method only contains 276,255 trainable parameters (which is significantly less than the 5,240,513 trainable parameters of the CNN-Bi GRU method of [2]). The proposed framework generates predictions for 100 recordings (each of which has a length of 15 sec) in 1.231054 sec using an Nvidia-GeForce RTX 2080 GPU. Note that this includes time taken for both feature extraction and the generation of model predictions. While these evaluations were conducted using a GPU due to its efficiency, the lightweight nature of our model allows it to be deployed in end user devices such as smart phones and has the ability to generate predictions in real-time.

## VI. CONCLUSION

In this paper, we introduced a novel deep learning framework for event detection in respiratory auscultation. We demonstrated how the recent successes in temporal convolution neural network-based temporal modelling, as well as efficient feature strategies, can be incorporated to mitigate the challenges of data scarcity and irregular/noisy recordings. We conducted experiments using multiple benchmarks, including, the HF Lung V1 Database, the ICBHI Respiratory Sound Database and privately collected data (M3DICINE Lung sound database) to evaluate performance for numerous lung sound event detection tasks. The proposed method outperforms the current state-of-the-art methods in all the evaluated event categories, including detection of inhalation, exhalation, crackles, wheeze, stridor, and rhonchi. Moreover, our architecture is simple and efficient in terms of the number of trainable parameters and computational requirements during inference, which affirms its utility in resource constrained environments. In addition, we provided empirical evaluations of the novel components of the proposed architecture as well as qualitative interpretations of the model prediction, which has been overlooked by existing literature. The evaluations demonstrated the utility of the proposed multi-branch temporal modelling strategy and the effectiveness of the proposed feature fusion mechanism in overcoming the challenges posed by data limitations. Such cost effective implementations, combined with the availability of model interpretations (which increases the trust in the output decisions), make the proposed system suitable for adoption by clinicians for use in hospital settings. Furthermore, we would like to highlight that the proposed framework is not restricted to the analysis of lung sound recordings and could be applied to the analysis of other one-dimensional signals, to detect potential abnormalities in these signals.

## ACKNOWLEDGMENT

The authors would also like to thank M3DICINE HOLDINGS PTY LTD for providing access to the M3DICINE lung sound database.

## REFERENCES

- [1] L. D. Pham, H. Phan, R. Palaniappan, A. Mertins, and I. McLoughlin, "CNN-MoE based framework for classification of respiratory anomalies and lung disease detection," *IEEE J. Biomed. Health Informat.*, vol. 25, no. 8, pp. 2938–2947, Aug. 2021.
- [2] F.-S. Hsu *et al.*, "Benchmarking of eight recurrent neural network variants for breath phase and adventitious sound detection on a self-developed open-access lung sound database-HF\_Lung\_V1," *PLoS ONE*, vol. 16, no. 7, pp. 1–26, 2021.
- [3] V. Raj, A. Renjini, M. Swapna, S. Sreejyothi, and S. Sankaraman, "Non-linear time series and principal component analyses: Potential diagnostic tools for COVID-19 auscultation," *Chaos, Solitons Fractals*, vol. 140, 2020, Art. no. 110246.
- [4] E. Messner *et al.*, "Crackle and breathing phase detection in lung sounds with deep bidirectional gated recurrent neural networks," in *Proc. 40th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2018, pp. 356–359.
- [5] C.-H. Hsiao *et al.*, "Breathing sound segmentation and detection using transfer learning techniques on an attention-based encoder-decoder architecture," in *Proc. 42nd Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2020, pp. 754–759.
- [6] "The World Health Organization website," Accessed: Jan. 21, 2022. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/asthma>
- [7] Y. Liu *et al.*, "Respiratory sounds feature learning with deep convolutional neural networks," in *Proc. IEEE 15th Int. Conf. Dependable, Autonomic Secure Comput., 15th Int. Conf. Pervasive Intell. Comput., 3rd Int. Conf. Big Data Intell. Comput. Cyber Sci. Technol. Congr.*, 2017, pp. 170–177.
- [8] C. Jácome, J. Ravn, E. Holsbø, J. C. Aviles-Solis, H. Melbye, and L. Ailo Bongo, "Convolutional neural network for breathing phase detection in lung sounds," *Sensors*, vol. 19, no. 8, pp. 1798–1799, 2019.
- [9] A. Gurung, C. G. Scraftford, J. M. Tielsch, O. S. Levine, and W. Checkley, "Computerized lung sound analysis as diagnostic aid for the detection of abnormal lung sounds: A systematic review and meta-analysis," *Respir. Med.*, vol. 105, no. 9, pp. 1396–1403, 2011.
- [10] D. S. Morillo, A. L. Jimenez, and S. A. Moreno, "Computer-aided diagnosis of pneumonia in patients with chronic obstructive pulmonary disease," *J. Amer. Med. Informat. Assoc.*, vol. 20, no. e1, pp. e111–e117, 2013.
- [11] G. Altan, Y. Kutlu, and N. Allahverdi, "Deep learning on computerized analysis of chronic obstructive pulmonary disease," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 5, pp. 1344–1350, May 2020.
- [12] S. B. Shuvo, S. N. Ali, S. I. Swapnil, T. Hasan, and M. I. H. Bhuiyan, "A lightweight CNN model for detecting respiratory diseases from lung auscultation sounds using EMD-CWT-based hybrid scalogram," *IEEE J. Biomed. Health Informat.*, vol. 25, no. 7, pp. 2595–2603, Jul. 2021.
- [13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [14] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Empir. Meth. Nat. Language Process.*, 2014, pp. 1724–1734.
- [15] T. Fernando, H. Ghaemmaghami, S. Denman, S. Sridharan, N. Husain, and C. Fookes, "Heart sound segmentation using bidirectional LSTMs with attention," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 6, pp. 1601–1609, Jun. 2020.
- [16] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1003–1012.
- [17] F. Demir, A. Sengur, and V. Bajaj, "Convolutional neural networks based efficient approach for classification of lung diseases," *Health Inf. Sci. Syst.*, vol. 8, no. 1, pp. 1–8, 2020.
- [18] K. Minami, H. Lu, H. Kim, S. Mabu, Y. Hirano, and S. Kido, "Automatic classification of large-scale respiratory sound dataset based on convolutional neural network," in *Proc. 19th Int. Conf. Control, Automat. Syst.*, 2019, pp. 804–807.
- [19] E. Messner *et al.*, "Multi-channel lung sound classification with convolutional recurrent neural networks," *Comput. Biol. Med.*, vol. 122, 2020, Art. no. 103831.
- [20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
- [21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
- [22] T. Fernando, S. Denman, A. McFadyen, S. Sridharan, and C. Fookes, "Tree memory networks for modelling long-term temporal dependencies," *Neurocomputing*, vol. 304, pp. 64–81, 2018.
- [23] T. Fernando *et al.*, "Neural memory plasticity for medical anomaly detection," *Neural Netw.*, vol. 127, pp. 67–81, 2020.
- [24] A. Abushakra and M. Faezipour, "Acoustic signal classification of breathing movements to virtually aid breath regulation," *IEEE J. Biomed. Health Informat.*, vol. 17, no. 2, pp. 493–500, Mar. 2013.
- [25] A. Abushakra and M. Faezipour, "Augmenting breath regulation using a mobile driven virtual reality therapy framework," *IEEE J. Biomed. Health Informat.*, vol. 18, no. 3, pp. 746–752, May 2014.
- [26] M. Faezipour and A. Abuzneid, "Smartphone-based self-testing of COVID-19 using breathing sounds," *Telemed. E-Health*, vol. 26, no. 10, pp. 1202–1205, 2020.
- [27] E. Matthew Davies and S. Böck, "Temporal convolutional networks for musical audio beat tracking," in *Proc. 27th Eur. Signal Process. Conf.*, 2019, pp. 1–5.
- [28] Y. A. Farha and J. Gall, "MS-TCN: Multi-stage temporal convolutional network for action segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3575–3584.
- [29] B. Rocha *et al.*, "A respiratory sound database for the development of automated classification," in *Proc. Int. Conf. Biomed. Health Informat.*, 2017, pp. 33–37.
- [30] L. Pham, H. Phan, R. Palaniappan, A. Mertins, and I. McLoughlin, "CNN-MoE based framework for classification of respiratory anomalies and lung disease detection," *IEEE J. Biomed. Health Informat.*, vol. 26, no. 10, pp. 1202–1205, Oct. 2020.
- [31] D. S. Park *et al.*, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech*, 2019, pp. 2613–2617.
- [32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.
- [33] K. Dharmadhere, M. Sundararajan, and Q. Yan, "How important is a neuron," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–15.
- [34] A. Vaswani *et al.*, "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.
- [35] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3145–3153.