Tampere University

Mikyong Deborah Shin

Adaptation of Pre-trained Deep Neural Networks for Sound Event Detection Facilitating Smart Homecare

Faculty of Information Technology and Communication Sciences (ITC)
Master's thesis
Examiner: Professor Tuomas Virtanen
Examiner: Academy Research Fellow Okko Räsänen
May 2023

Abstract

Mikyong Deborah Shin: Adaptation of Pre-trained Deep Neural Networks
for Sound Event Detection Facilitating Smart Homecare
Master's thesis
Tampere University
Master's Degree Programme in Information Technology
May 2023

As foreseen by numerous researchers, the worldwide demographic changes of the elderly population in 2050 will be expected to grow by over 30% in the global population, which has urged to development of cost-efficient and effective automated sound recognition systems to assist the well-being of the self-living older people in their homecare environment. Consequently, in recent research in sound event classification and detection systems, there has been increasing research on adapting the pre-trained model YAMNet because it can classify 521 sound event classes trained with a large-scale AudioSet dataset. Despite the huge potential, the main problem of using the YAMNet predictions was observed in our early investigation difficulty in finding associated YAMNet classes for the target events predefined in public benchmark acoustic datasets. This study aimed to investigate this class mapping complication to adapt the YAMNet pre-trained model into a sound event detection system with temporal information for monitoring abnormalities in residential homecare environments. A new Y-MCC methodology was developed based on the Matthews correlation coefficient (MCC) to resolve the original YAMNet class map and produce new class maps according to the MCC thresholds. The performance of the Y-MCC system successfully demonstrated the SED system feasibility by achieving the best F1 score of 59.46% in the overall micro-average on the SINS dataset and class-wise F1-score performance of 'sheep' at 100% and 'brushing teeth' at 96.8% in ESC-50 and 'vacuum cleaner' at 94.7% in SINS, and 'water tap running' at 58.5% in TUT-SED 2016 Home datasets. This indicates the potential use of the Y-MCC method for facilitating automated sound event monitoring systems in smart homecare applications.

Keywords: Sound event detection system, MCC-based YAMNet class mapping (Y-MCC), Matthews correlation coefficient (MCC), YAMNet pre-trained model, Smart homecare application, TUT-SED 2016 dataset, ESC-50 dataset, SINS dataset.

Contents

1 Introduction

In recent years, sound event detection (SED) systems have been broadly investigated to support homecare for elderly people as a subdomain of the artificial intelligence (AI) driven smart home market [1]. In terms of the share of the elderly in the global population in the near future, a study conducted by the World Health Organisation (WHO) has made predictions about the age older than 65 population, which will rise rapidly in recent decades and expect to reach 38% of the global population, approximately 2.1 billion people by 2050 [2]. Similarly, a European level of study regarding long-term care challenges investigated over 35 countries has estimated that the age group with over 80 population will be more than double by 2070 [3]. These studies have urged strategic action plans for the glowing demand for long-term care services dedicated to aging people in the home and institutional care settings because of critical indications of the lack of care resources and insufficient financial support [2, 3]. This is why it should be necessary to emphasize that smart home technologically driven systems, including automated SED systems, should be more considerably investigated and researched to economically assist the well-being of the homecare residents and their healthcare professionals [4].

In recent literature publications, there is much evidence that research environments for the SED system development have been introduced for fostering to support the needs of the homecare inhabitants. Especially the SED research community has been considerably grown by the annual Detection and Classification of Acoustic Scenes and Events (DCASE) challenges and public benchmark dataset providers, such as the ESC-50 dataset, SINS database, and AudioSet dataset [5, 6]. Moreover, the SED systems are evolving from the predefined sound detection functionality toward semantic interpretation of the sound event scene to react to the system according to the interpretation [1, 7]. Consequently, to develop the SED system for the current and future needs of smart homecare applications, it could be beneficial to adapt the YAMNet model suggested in [8], which has established hierarchically structured sound event ontology with 521 classes trained and evaluated based on AudioSet dataset. Furthermore, the YAMNet pre-trained model based on the MobileNet version 1 depth-wise convolutional architecture, in comparison with the high computational complexity of the VGGish pre-trained deep neural network model also trained on the AudioSet dataset, has a much smaller size of the weights in the number of parameters (3.7M vs. 72.1M). Therefore, the YAMNet model with approximately 16 MB could easily be portable to systems with smaller computing resources, such as mobile devices and the Internet of Things (IoT) devices.

Closely looking at the performance of the YAMNet model-based classification systems in recent years has found that there has been remarkable performance improvement using different approaches. As a common approach used in the healthcare field, a study conducted using the YAMNet model in the pre-processing stage of the COVID-19 cough classification system has achieved an accuracy of approximately 97%, over 14% significantly higher than their VGGish-based model [9]. And another approach using the feature extraction functionality of the YAMNet model for speech event recognition Alzheimer's dementia classification system has been introduced with 83% accuracy [10]. As a more advanced approach by retaining and optimizing the YAMNet model for multi-event classification on three selected public benchmark datasets, including ESC-10 and UrbanSound8K, has reached over 90% accuracy for both datasets, which was slightly higher than their VGGish-based model [11].

However, in the YAMNet-based studies, less attention has been paid to unreliable YAMNet class predictions, which could easily observe mostly the YAMNet model testing with a publicly available acoustic dataset with its predefined classes. Our early investigation of the YAMNet model showed that class mapping from YAMNet 521 classes to the multiple target classes in given datasets was often incredibly untrustworthy. Firstly, YAMNet predictions with top highest probability were often from the classes belonging to the higher hierarchy, which might not be strongly necessary to indicate the target sound event, for example, 'Inside, small room' for the 'eating' target class. Secondly, YAMNet might produce multiple classes with high probability, which seemed unrelated to the target class. For example, the target class of 'vacuum cleaner' could be predicted by YAMNet as multiple 'Blender,' 'Hair dryer,' 'Tools,' and 'Vacuum cleaner.' Finally, these unpredictable class-label predictions were more complex with real-life recording datasets and highly overlapping multiple sound events, such as the TUT-SED 2016 Home dataset. As a result, the performance of YAMNet was extremely poor. Therefore, additional studies are needed to resolve the class association between the YAMNet and the target classes specified in the public datasets.

This research proposes a statistical class mapping method based on the Matthews correlation coefficient (MCC) for a sound event detection system to improve YAMNet performance for given target datasets containing essential sound activities for homecare applications. The method, Y-MCC, consists of two channels and three stages in each channel. The first channel aims to produce reliable MCC-threshold-based class maps by computing correlations between the YAMNet predicted classes and the given target classes. On the other hand, the second channel is designed to produce system predictions with temporal boundaries based on the MCC-threshold-

based new class maps, which replace the original YAMNet class map. Therefore, the performance of the Y-MCC method can be evaluated for the given dataset according to the SED system evaluation tool used in the DCASE Challenge. Consequently, our results indicate that the Y-MCC system has achieved close to optimum class-wise performance scores compared to their baseline systems of carefully selected three benchmark datasets; TUT-SED 2016, ESC-50, and SINS datasets contain a total of 70 sound event classes mostly related to monitoring indoor homecare activities.

The remaining part of the thesis paper proceeds as follows: Chapter 2 presents the theoretical background of SED systems and previous related work for Smart Homecare applications, reviews regarding commonly used acoustic datasets for SED system training and evaluation, and further details of the YAMNet architecture and performance evaluation methodologies. Chapter 3 presents the procedures and processing steps of the proposed Y-MCC methods in detail. The class mapping, performance results, and analysis of the Y-MCC system using the three benchmark datasets are presented and analyzed in Chapter 4, and a conclusion is given in Chapter 5.

2 Theoretical Background

This chapter presents the most relevant publications that have been the basis of this study using the YAMNet pre-trained model for homecare applications. The first Chapter 2.1 explains the fundamental theories for developing the SED system for smart homecare applications. Next, Chapter 2.2 discusses the SED system design aspects of adapting the YAMNet pre-trained model for the SED system that could optimize for achieving adequate performance by understanding hyperparameters defined in the model and relevant feature extraction methods. Finally, Chapter 2.3 elaborates on benchmark datasets highly recognized as important for domestic SED system development and standardized evaluation methodologies for SED system performance evaluation.

2.1 Sound Event Detection System for Smart Homecare

This chapter introduces sound event detection system development for smart homecare applications that could support elderly people living in assisted environments and their healthcare professionals. Firstly, review the recent trend of the smart homecare applications in the acoustic event recognition field and briefly discuss the sound characteristics and how these are related to sound features and tasks of the SED system.

2.1.1 Smart Homecare Applications

Sound events in older people's homes might differ from those commonly observed in urban life. For example, instead of sound events widely regarded in urban daily life, such as children shouting and people walking, they might be replaced by sound events like silence and slow footsteps assisted by a walking bike in the elderly homecare facilities. Notably, the elderly are highly vulnerable to reacting to protect their well-being by themselves due to physical disability or psychological problems [12]. Therefore, as a key driver of the Homecare market, the SED system offers plenty of benefits that can contribute to the safety of the care receivers and the efficiency of their care providers if they are connected through the network to receive alarms for emergencies [1, 4].

The SED system for older people can be categorized depending on whether the system is used in institutions like nursing homes or individual homecare facilities. One is the acute care domain, where the SED system is applied to nursing home patients, whereas the homecare domain, where the SED system is used to assist in

self-management of the elderly living in their home environment [13]. The basic design concept of the SED system could detect pre-defined abnormal sound event classes to trigger alarms for homecare patients or healthcare professionals, such as falling or calling for help as a most interesting sound event for the homecare domain [14, 15]. The more recent trend of the SED systems investigated to provide advanced functionality on top of the pre-selected sound event detection called context-aware functionality, which could be built on the SED system to provide situational information [13, 16, 17]. For example, specific location (bedroom or bathroom), situational context (meal or shower), and event duration; this situational information could be more accurate when multiple ambient sensors are installed in the homecare or nursing home environment. Unlikely wearable or mobile devices, ambient sensors could be preferred for monitoring the elderly homecare facilities because they reduce the burdens of the elderly people no need to wear sensors and allow them to do natural daily activities [18].

2.1.2 Acoustic Features for Sound Analysis

Understanding sound for sound analysis starts with human perception of sound, which requires human reaction based on interpreting sound events. Human listeners could receive waveforms of sounds produced by many different objects traveling through mediums, such as air, liquid, and solid objects [19, Chap. 3, 20]. The perception of sounds for human listeners involves complex auditory systems and brain regions. In the human auditory system, as illustrated in Figure 2.1, the waveforms of sounds propagate from the external ear to the inner ear, followed by the peripheral auditory structures (outer, middle, and inner ear, including the cochlea and auditory nerve) [20–22]. After that, perceptual information, such as pitch, timbre, and spatial location, is processed in the brain's auditory circuits (from cochlear nuclei to the auditory cortex) to transform the sounds into encoded sound information [23]. Subsequently, it is further analyzed by isolating essential sounds from background noise and identifying them as sound sources or environmental indications [19–21]. Furthermore, the auditory system in the brain regions interprets the acoustic scene in a temporal context, which could lead the human listener to conduct appropriate behaviors or reactions according to the interpretation of the sound events [21, 23, 24].

The sound signal recorded with microphones can be characterized by two properties: frequency and amplitude [26]. Frequency is the number of sinusoidal waves per second that determine the sound pitch, from low to high, and its scale is measured in Hertz (Hz). Hearing capability varies significantly between individuals; hence, those with a sensitive auditory system can listen to frequencies ranging from 20 Hz to 20 kHz [23]. And the other attribute of a sound signal is the amplitude of the

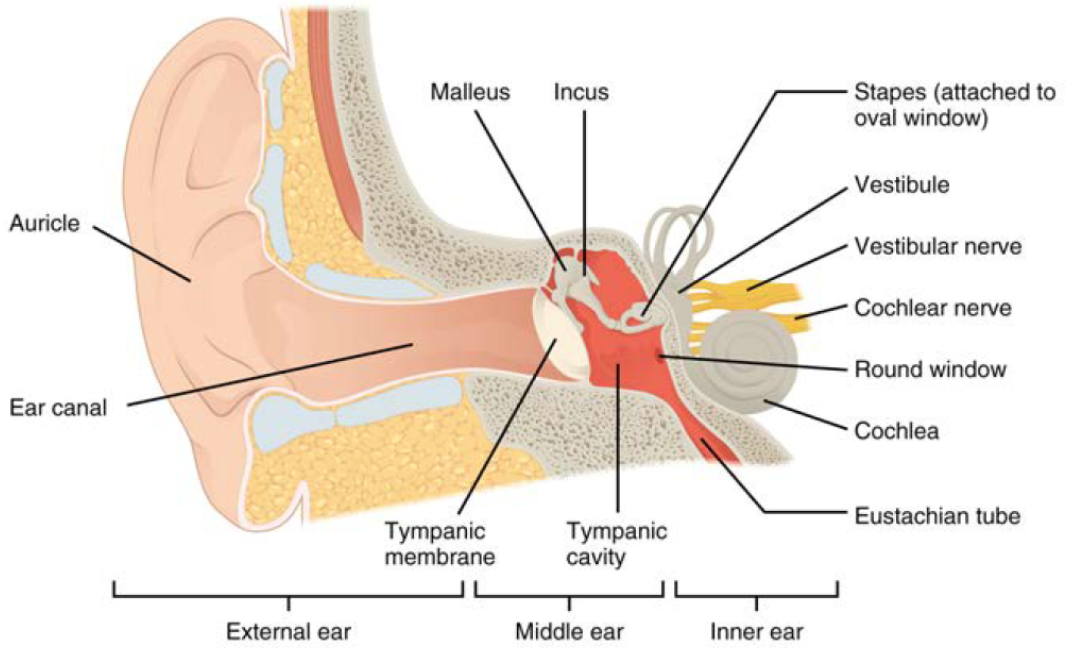

Figure 2.1 *The physiology of the ear concerning human hearing consists of three peripheral auditory structures: outer, middle, and inner ear, that are connected to integrate information into perception in the auditory cortex in the brain [22]. Image adapted from [25] under the Creative Commons Attribution license.*

vibration, which has a subjective correlation with loudness perceived differently by different individuals. The objective measure of the sound amplitude scale is called sound pressure level (SPL), computed in decibels (dB). Ordinary humans can hear the sound range of amplitudes from 0 to 120 dB.

To analyze the acoustic signal, acoustic features of sound can be represented in different domains, as many researchers suggested, it can be applied based on the application of the sound analysis system [27, p. 71-114, 19, Chap. 5, 28, Chap. 3]. Especially time domain, frequency domain, and time-frequency domain signal analysis are very well-known. First, for time-domain signal analysis, the signal is represented by amplitude over a specific duration of time scale. The time-domain signal representation is a sum of sine waves with phase and amplitude information. In addition, the time domain signal saved in computing devices is a digital format sampled with a sampling rate (SR, or f_s) by taking the samples according to the sampling theorem of Nyquist frequency [29, Chap. 4]. The Nyquist frequency is to avoid the frequency aliasing phenomena by determining the SR to be double the rate of the maximum frequency sinusoidal present in the audio signal [29, Chap. 4]. Second, the frequency-domain signal analysis called power spectral density (PSD), or power spectrum (PS), can be obtained by taking power, or square, of the absolute value of a discrete Fourier transform (DFT) of the time domain signal [30]. In prac-

tice, the DFT computation can be obtained using the most well-known and efficient algorithm called fast Fourier transform (FFT) over the finite length of the input waveform signal [29, Chap. 3]. The PSD can be represented with a decibel scale (dB) for the magnitude of frequencies presented in the sound signal that provides more clear acoustic properties than the time-domain analysis. However, a drawback of PSD estimation is that it does not contain time information due to PSD computation over whole signal samples. Finally, the time-frequency domain representation, a spectrogram of 2-dimensional representation, is obtained by blocking and windowing the signal into short analysis frames and then applying the discrete Fourier transform to each frame. Because it preserves time information and spectral magnitude, it is a widely used feature extraction method for machine learning.

Figure 2.2 visualizes the three domains of audio signal analysis using a 10-second audio clip taken from the DESED dataset provided by the DCASE 2019 Challenge task 4 [31], which contains doorbell ringing sounds overlapped with dog barking sounds in a domestic environment. The time domain in Figure 2.2 (a) shows the signal amplitude over the 10 seconds, sampled with a sampling rate of 44.1 kHz. It can be noticed that the two sound sources have different amplitude ranges (smaller for alarm bell ringings and larger for dog barking), with events occurring at time points and their duration; however, it could be hard to differentiate two sound events when they are overlapped. On the other hand, the PSD of the frequency-domain representation, Figure 2.2 (b), can easily distinguish two sound events by estimating the frequency of the sound sources. The PSD estimation using a NumPy library function of fast Fourier transformation (FFT) [32] decomposed the audio wave signal into a discrete frequency range [0, 22050] over 10 seconds. The spectrogram representation of the time-frequency domain, shown in Figure 2.2 (c), was obtained by applying logarithm operation after performing the short-time Fourier transform (STFT) function [33] to extract time-frame-wise frequency component representation with parameters of 1024 frame size and 512 hope length. Using the logarithmic spectrogram, it can be easily noticed that the alarm bell ringing events with red horizontal lines contrast the dog sounds with a higher power of magnitude as shown by many vertical lines, indicating a wider frequency range for dog sounds. Furthermore, it is clear to recognize two different sound events overlapping, for example, at 4.5 seconds. Therefore, the 2-dimensional view of the spectrogram analysis provides a better possibility to analyze the sound sources even though overlapping sound events with time and frequency information than the time domain and PSD estimation. Various time-frequency domain feature extract methods used in SED systems are discussed further in Chapter2.2.3.

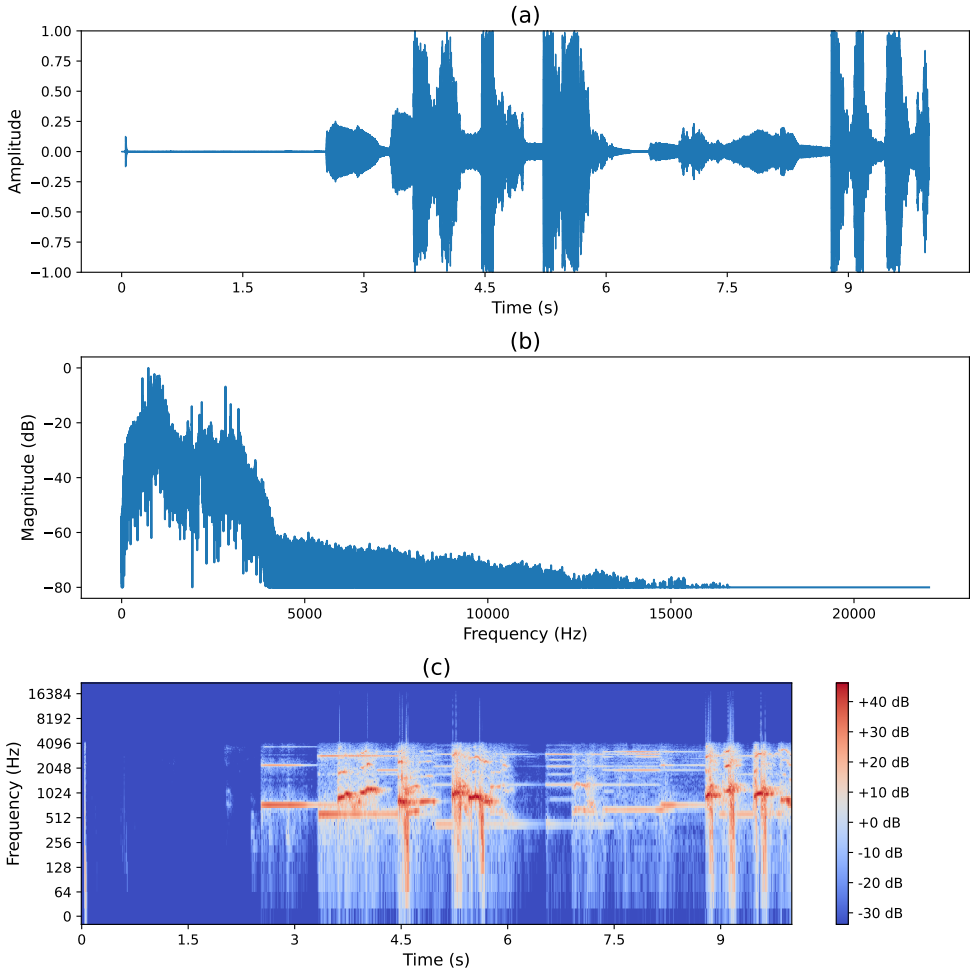

Figure 2.2 *Visualization of three audio signal analysis domains using an audio sample from DESED dataset provided in the DCASE 2019 Challenge task 4 [31]: (a) time-domain, (b) frequency-domain, and (c) time-frequency domain with a log-scale frequency representation. The 10-second duration of the audio sample contains doorbell ringings (starting from approx. 2.5 seconds) and dog barking sounds overlapped (starting from approx. 3.5 seconds) in a domestic environment.*

2.1.3 Sound Event Detection System

Sound event detection (SED), also known as audio event recognition (AER), is a well-established research area in the signal processing field aiming to research and develop automated sound event detection systems that can produce class labels with event occurrence temporal information in real-time or recorded audio streams [34]. The event classes of everyday sound in the urban environment can be used in various sound categorization methods reflecting cognitive psychology into the SED system [19, Chap. 7]. As an example illustrated in Figure 2.3, it is common that sound categorization can be hierarchically divided by the source of the sound from general to specific. For example, the sound source of humans and animals belongs to the higher level of animate agents; in comparison, 'door' and 'window' are grouped

under the solid sound source branched from inanimate agents. Moreover, the action descriptors are grouped under the subcategories of the sound sources, such as 'yell' and 'cough,' which belong to the vocalization of human stationary sound. Consequently, it is more precise and apparent when the source and action descriptors are combined, such as children shouting, door opening and car engine accelerating. One of the studies has suggested that the categorization for everyday sounds could be based on multiple descriptors: action, source, and context [19, Chap. 7]. The action descriptors are often related to daily human habits, such as cooking, eating, cleaning, exercising, and sleeping. And using the three descriptors of action associated with the source and context of the sound helps humans to perceive sound events and analyze the information for what actions are required for the situations [19, Chap. 7].

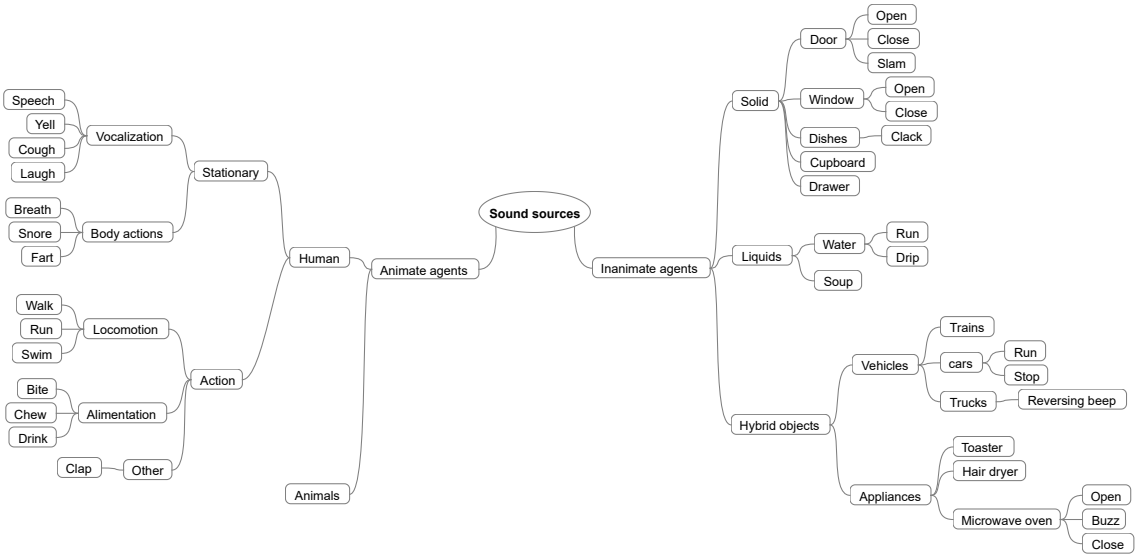

Figure 2.3 *An example of the taxonomy of everyday sound presents relationships of sound sources and actions categorized in multiple depths for helping to understand human perception of sound in the context of an urban environment, adapted from [19, Chap. 7]*

Figure 2.4 depicts that a simplified workflow of the polyphonic SED system consists of three stages: pre-processing, machine learning algorithms, and post-processing. In the first pre-processing stage, the SED system processes audio recordings and transforms their temporal characteristics into time-frequency spatial features. In the next step, the core part of the SED system employs a machine learning algorithm to produce predictions of recognized sound events from the input features. Finally, post-processing can perform specific tasks to analyze the predictions and make human-readable output. As a polyphonic SED system illustrated in the figure, the system output can contain multiple class labels for the same time slices marked with the onset and offset time boundaries. For instance, three sound events occurred at the same time, close to the end of the input audio clip with class labels

of "Object impact," "People walking," and "Door closing." As explained earlier, in this example, sound event class labels use two descriptors of an action word with an animate or inanimate agent of the sound source. As can be seen, this is the key challenge of the SED system, especially in processing real-life audio recordings to deal with overlapping sound events [35, 36, 19, Chap. 8]. In contrast, a monophonic SED system can detect only one predefined sound event label, producing the most prominent sound event in each temporal region [19].

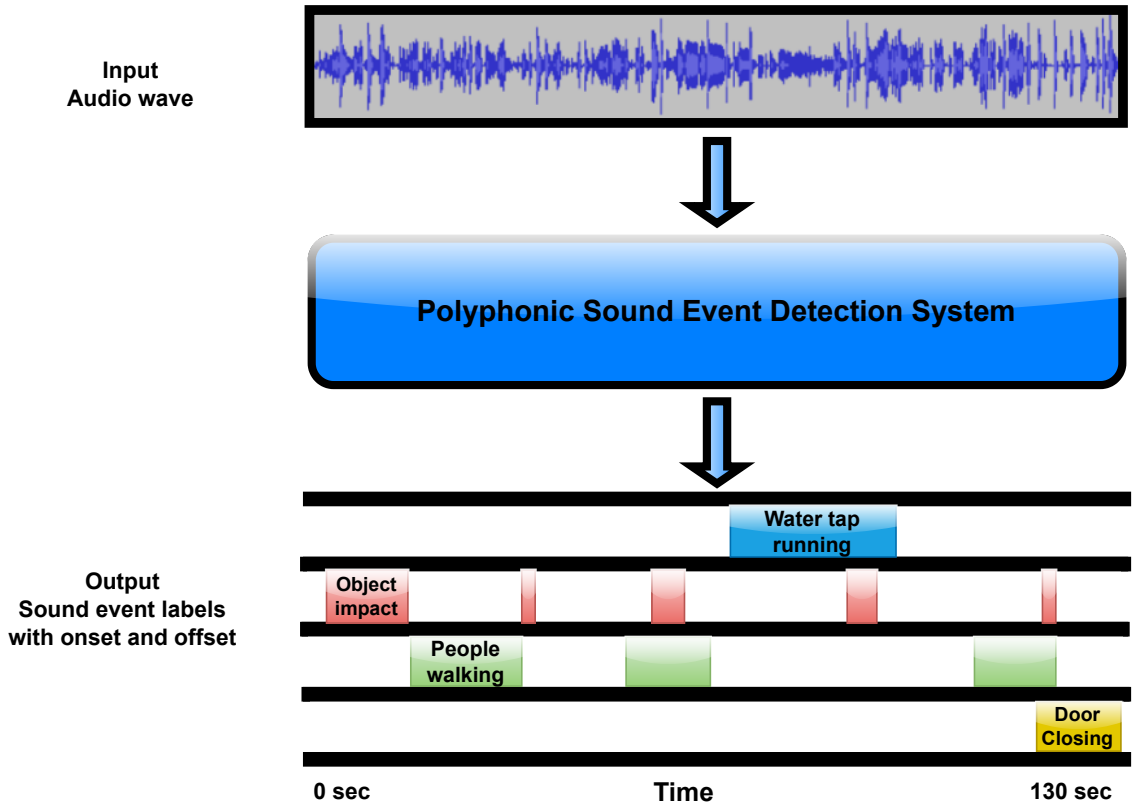

Figure 2.4 *A simplified workflow of the polyphonic SED system shows the relationship between the input waveform data processed through machine learning algorithms and produced output with multiple event labels, some overlapping, with temporal information.*

2.1.4 Tasks of Sound Event Detection System

Generally, tasks of SED systems introduced in the DCASE challenges have changed accordingly to support its community and the SED system development. The competition is sponsored by IEEE Audio and Acoustic Signal Processing Technical Committee and provides platforms for baseline systems, public datasets, standardized evaluation tools, and benchmark results [37]. The first DCASE challenge was organized in 2013, becoming an annual event in 2016. Most top-ranking submissions are considered state-of-the-art machine learning methods in this field of study [5, 37].

The initial fundamental four challenge tasks introduced in the DCASE 2016 challenge[1] were acoustic scene classification, domestic audio tagging, sound event detection in synthetic audio, and sound event detection in real-life audio [5]. The classification task is an entry point of the challenge, which needs to produce a single class label for the most prominent sound event per audio recording. On the other hand, the audio tagging task need to find multiple event labels per audio recording. Unlike previously discussed SED tasks, the classification and tagging tasks do not necessarily produce temporal information for the detected event labels.

SED task with the synthetic audio dataset as a counterpart of the real-life audio dataset can benefit the SED system because it has produced a controlled way that keeps balance among classes and reliable annotations. Conversely, the real-life audio dataset could be unbalanced class-wise and often has inadequate class labels and onset and offset annotations. The real-life annotation made by human annotators could be a significantly challenging task to annotate precisely, especially for overlapping events [5, 36]. Consequently, some degree of mismatching information in reference files to the real-life audio dataset might affect to decline in the performance of the machine-learning algorithms.

Classifying domestic activities recorded with multi-channel sensors has been introduced in the DCASE 2018 challenge [38]. The dataset used in the challenges was taken from the SINS database, which included the sound event from the everyday daily activities of a person living in a home setting and recorded for one week using multiple sensors, including a linear microarray [39]. With the multi-channel dataset recorded in the living room and kitchen, this challenging task enhanced the concept of the SED system from a basic event detection task toward context awareness in monitoring home activities.

[1] `https://dcase.community/challenge2016/`

2.2 Deep Neural Networks for Sound Event Detection

This chapter explores important aspects of the SED system using the YAMNet pre-trained model, firstly discusses a recent trend in the YAMNet-based systems that have achieved significant performance on their various sound event detection tasks after that YAMNet model architecture and its hyperparameters are addressed. Lastly, various feature extraction methods and main aspects from the time-frequency domain for deep neural network models are explained because of their importance in the SED system's performance considering the SED system applications.

2.2.1 YAMNet Pre-trained Deep Neural Networks

Pre-trained Models on AudioSet Dataset

YAMNet[2] and VGGish are pre-trained deep neural networks (DNN), which were trained on the large volume of AudioSet dataset published by Google researchers [8]. These two breakthrough pre-trained models trained on sound samples extracted from over 2 million YouTube videos. A trend of recent studies showed an increasing adaptation of the pre-trained models in many research areas. However, it is vital to know the advantages of the YAMNet pre-trained model against the VGGish model, which might explain the selection of the YAMNet for this study.

As presented in Table 2.1, despite the two models trained on the same AudioSet dataset, YAMNet has fewer network parameters used in its CNN core networks than VGGish, 3.7 versus 72.1 million, respectively. Moreover, the size of the VGGish model over 500MB is quite huge among the traditional pre-trained models because its network architecture is based on the traditional CNN model of VGGNet [40]. Conversely, the size of the YAMNet model is much smaller, approximately 16 MB, because it was built based on the first version of MobileNet [41] designed for mobile applications. The MobileNet employed depthwise-pointwise separable convolutions convolutional networks, which reduced the number of parameters and made it easily portable to systems with limited computational resources and latency problems, such as mobile and IoT devices.

YAMNet-based Deep Neural Network Systems

To understand the benefits of using the pre-trained YAMNet for sound recognition tasks, it is crucial to analyze recent research on the YAMNet-based systems relevant to homecare applications. Different and meaningful approaches to YAMNet utilization have been explored for various needs of applications, as summarised

[2] `https://github.com/tensorflow/models/tree/master/research/audioset/yamnet`

Table 2.1 *A comparison of pre-trained deep neural network models in connection with YAMNet.*

Pre-trained Model	Dataset	Parameters	Input	Classes
MobileNet-v1 [41]	ImageNet	4.3M	3-D vector (224x224x3)	1000
VGGish [8]	AudioSet	72.1M	log-mel spectrogram	527
YAMNet [8]	AudioSet	3.7M	log-mel spectrogram	521

in Table 2.2. These works of literature show four different approaches to utilizing YAMNet for classification, detection, and transfer learning tasks to improve its accuracy for given experimental datasets.

Table 2.2 *Recent publications based on the YAMNet pre-trained model for domestic sound event classification.*

Reference Method	YAMNet Approach	YAMNet Class	Model Accuracy(%)	Dataset
[9] Detection of COVID-19 cough	Classification	Cough	97.59	COUGHVID
				COSWARA
				VIRUFY
[10] Classification of Alzheimer's dementia	Feature extraction	Speech	83.33	CTP audio
[42] Detection of abnormal respiratory sound	Feature extraction	Wheeze, Crackle	81.49	ICBHI-2017 challenge
[11] YAMNet retrained	Optimization	Multi-classes	96.16	UrbanSound8K
			91.25	ESC-10
			100.00	Air Compressor

The first approach in [9] applied YAMNet's classification approach in the pre-processing phase of their COVID-19 cough detection downstream method. In their experiment, the identified cough segments by the YAMNet model were collected and processed to the classification method called ViT; otherwise, non-cough segments such as silence or speech segments were removed. Implementing the extracting segments with only the cough event method made their ViT classification model more robust, which classify healthy or symptomatic decision based on the fractal image of the pre-processed cough segments. It obtained higher accuracy scores of over 97% on COVID-19 symptomatic class detection evaluated on the three public COVID-19 datasets: COUGHVID, COSWARA, and VIRUFY. The authors emphasized that their ViT method outperformed approximately 14% of previous research based on a fine-tuned VGGish model on the COSWARA dataset [43].

The second approach proposed in [10] confirmed the effectiveness of the YAMNet pre-trained model compared to MobileNet for classifying Alzheimer's disease (AD) based on input from a Cookie Theft Picture (CTP) audio dataset. The similarity of speech data in AudioSet used for training YAMNet and CTP audio dataset might be improved the performance of YAMNet. Furthermore, the CPT dataset has a

relatively small amount of speech samples. Their downstream model of deep neural network needed to be supported by the learned features from the YAMNet pre-trained model, which was specially trained on a large scale of speech data to bring benefits of avoiding overfitting problems. In their proposed model with the YAMNet pre-trained model, the best accuracy at 83.33% to classify AD and non-AD exceeded approximately 4.16% more than the YAMNet without trained weights and 4.25% higher than the MobileNet pre-trained model.

In the same way, the approach of feature extraction from the YAMNet pre-trained model was applied to a downstream model suggested in [42]. They chose to utilize the YAMNet model because the dataset provided by ICBHI challenge [44] was small and unbalanced to detect abnormal respiratory sounds, such as wheezing and crackling. However, combining two feature extraction models, one based on YAMNet and the other based on the temporal coefficients extracted from the Discrete Wavelet Transform method, achieved a marginally better performance than the single YAMNet-based model: ICBHI-score of 81.49% vs. 75.88%, respectively.

Finally, the pre-trained YAMNet model can be retrained and optimized for its parameters to perform classification task better with specific datasets, referred to as transfer learning. Researchers proposed this method mentioned in [11] explored the transfer learning for three datasets: UrbanSound8K (10 classes from outdoor sounds, inc. 'Dog bark' and 'Car horn'), ESC-10 (10 classes, inc. 'Clock tick' and 'Person sneeze'), and Air Compressor dataset (8 classes from industrial sound, inc. 'Flywheel' and 'Rider belt'). They have optimized YAMNet and VGGish models for the three datasets by using nine different combinations of hyperparameters of the pre-trained models, for instance, with Adam optimizer to set mini-batch sizes with 64, 128, or 256, and maximum epochs at 10. The performance of the YAMNet over the three datasets achieved 96.16%, 88.06%, and 100% accuracy, respectively. The fine-tuned and retrained VGGish model with the same hyperparameters performed slightly below the retrained YAMNet model. However, their experiment of the two retrained sound-based models exceeded three image-based pre-trained models, such as GoogleNet, SqueezeNet, and ShuffleNet.

2.2.2 YAMNet Network Architecture

The YAMNet network architecture can be characterized as a series of depth-wise and pointwise convolutional neural network blocks that are layered with 14 blocks of hidden deep neural networks (DNNs) [8], as illustrated in Figure 2.5. YAMNet extracts meaningful features across the DNNs layers by employing this network architecture. Furthermore, the number of parameters of YAMNet is 3.7 million, which is significantly reduced compared to traditional convolutional neural networks

(CNNs) used in VGGish. These depthwise-separable convolution blocks are fully connected from the input and output layers, followed by a global average pooling layer [41]. In each depthwise-separable convolution block, batch normalization and rectified linear activation function (ReLU) layers are conducted after each depthwise convolution layer and pointwise convolution layer [41]. Finally, a nonlinear logistic sigmoid activation function estimates the 521 class-membership probability.

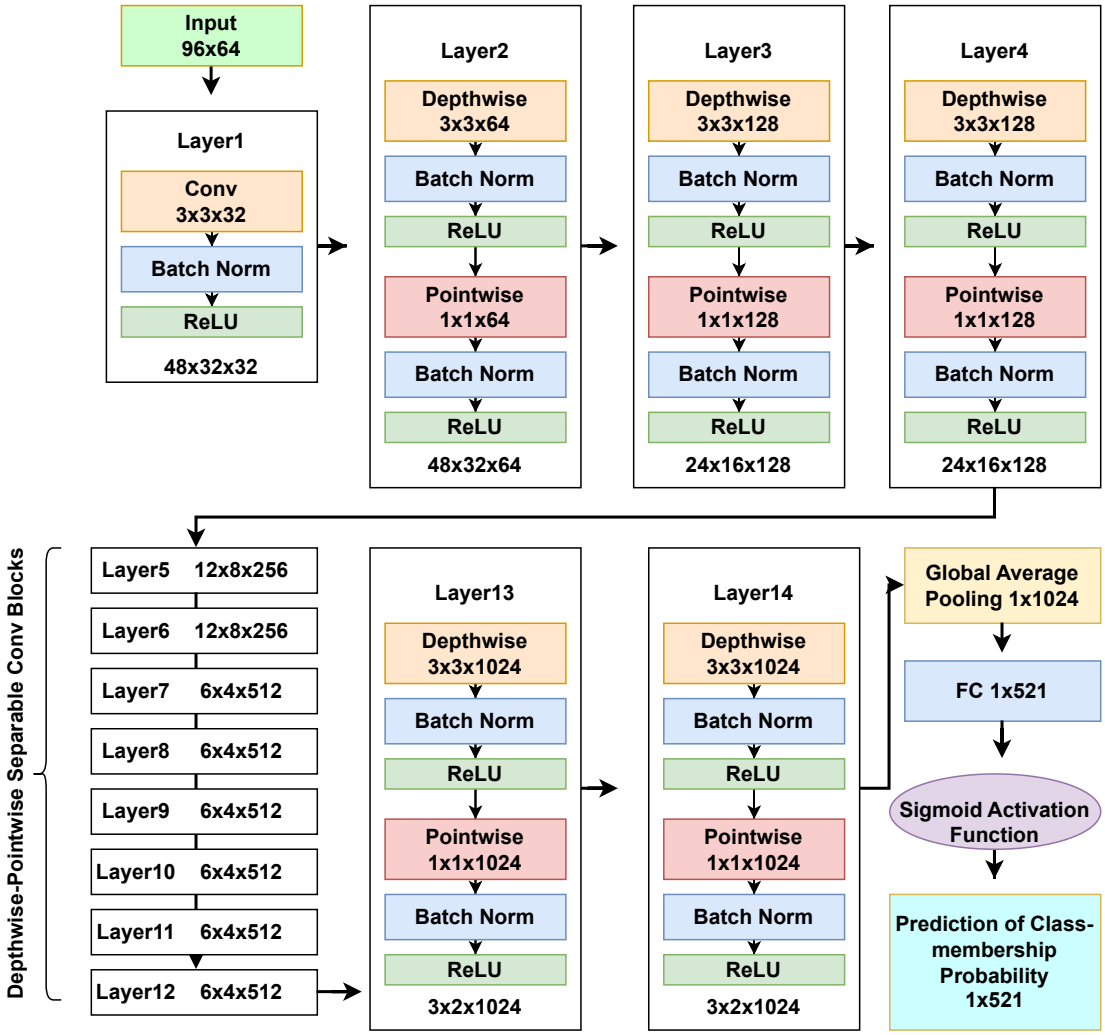

Figure 2.5 *Flows of the YAMNet deep neural network architecture based on MobileNetV1 [41] employed blocks of depthwise-pointwise separable convolution layers followed by the final global average pooling to produce 521 class membership probability using sigmoid activation function, adapted from [8].*

An S-shaped logistic sigmoid function as an activation function used in the YAMNet DNNs leads to non-linear decisions, which can be represented with the positive probability scores ranging in [0, 1] for each class label of 521 multiple events [45]. For example, if the results of the sigmoid denoted as $f(x) = 0.7$ for class number 10, 'children shouting,' meaning that the chance of the input sample belonging to class 10 is 70 percent. The sigmoid function $f(x) = 0.7$ can be calculated as

$f(x) = P(y = 10|x; w) = 0.7$, where given a feature x weighted by w [46]. The sigmoid function formula is defined as:

$$f(x) = \frac{1}{1 + e^{-x}}, \tag{2.1}$$

where x is obtained from weights and sample features combined by the classifier's fully connected layer, i.e., $x = w_0x_0 + w_1x_1 + ... + w_mx_m = \mathbf{w}^T\mathbf{x}$ [46, Chap. 3].

For feature extraction, YAMNet uses the log-mel spectrogram method that can be described as following procedures and hyperparameters [8]:

- Resampling all input audio with sampling rate 16 kHz mono
- Using the magnitude of the input signal and applying a short-time Fourier transform (STFT) to convert the mono resampled signals into a time-frequency linear spectrogram set with 25 ms window size, hop-size with 10 ms, default Hann window filter.
- Simulating triangular Mel filterbank with 64 Mel frequency bands and frequency range to cover with 125 Hz and 7500 Hz as the minimum and maximum, respectively.
- Applying logarithm multiplication into the Mel-spectrogram to produce the log magnitude of the Mel spectrogram.
- Divide the feature obtained from the log-mel spectrogram into samples with 0.96 seconds overlapping with 0.48 seconds, producing a patch with a matrix of 96 frames and 64 Mel bands.

The outcomes of the feature extraction are patches converted and divided from the input waveform into a log-mel spectrogram covering 0.98 seconds of time frames per patch. These patches are delivered to the YAMNet classifier model based on MobileNetV1. The classifier's output is represented with probability for 521 classes individually called scores, $\mathbf{S}_p \in \mathbb{R}^{1\times n}$ where $\mathbf{S}_p$ denoted a matrix with 1 row and n=521 columns representing scores of a patch with probability values ranged $[0, 1]$. Figure 2.6 visualized these three steps of YAMNet data handling, starting from input audio, then feature extraction to the log-mel spectrogram, and class predictions with the top-5 highest scores among the 521 classes.

2.2.3 Feature Extraction for Deep Neural Networks

Feature extraction is obtaining relevant and distinguishable characteristics from the given input audio signal. The machine learning-based SED classifier learns from the feature-extracted data, significantly impacting how the classifier performs for

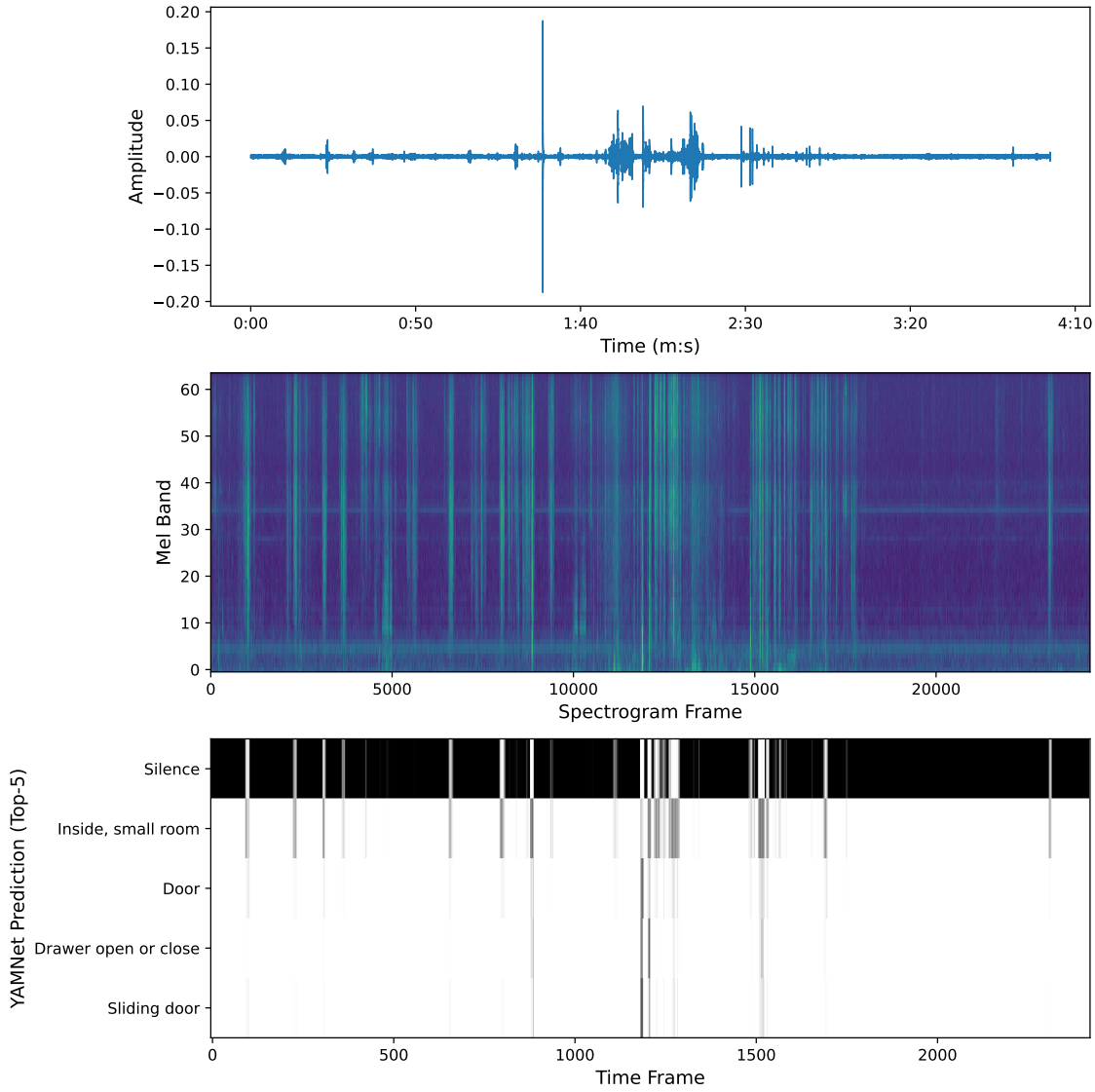

Figure 2.6 *YAMNet inference visualized [8] with an audio sample file taken from TUT-SED2016Home dataset [47]: (Top) Audio input shown as a waveform in time-domain down-sampled from SR=44.1kHz to SR=16kHz, (Middle) the log-mel spectrogram (mel band=64) of the feature extracted from the input waveform, (Bottom) Top-5 highest class predictions by the YAMNet model, where the probability score has been color-coded as the darker black color indicating higher probability. As can be seen, the highest scores of the YAMNet predictions have mostly 'silence' for the entire audio recording, while very short 'door' related events have been recognized in the middle with a medium probability.*

its prediction [48]. The feature extraction process involves converting the raw audio signal in waveform into feature vectors, representing acoustic characteristics in a much more compact size and form than the waveform, reducing the overload of the computing resources [48, 49]. For environmental sounds or human voices, capturing time-variant sound characteristics of the feature extraction is essential, which can be done in the time domain [27, 49, 48]. The feature extraction methods used in the temporal domain do not require transformation. Still, they can obtain directly

from the temporal waveform, such as Zero-crossing rate (ZCR), signal power, and autocorrelation coefficients [48]. However, a trend of recent SED system studies has shown great performance improvement using the time-frequency-domain features such as human perception-based Mel spectrogram and log-mel spectrogram [5, 48]. It has been more strongly moved toward the log-mel energy feature extraction method as used by most participants in the DCASE 2022 challenge SED task in domestic environments [31]. The visualized feature characteristics of the feature engineering methods are illustrated in Figure 2.7, which shows how various feature extraction methods can be performed in the time-frequency domain using the input audio recording, shown in Figure 2.7 (a), a 5-second audio clip of vacuum cleaning sound event taken from the SINS dataset [39].

To closely examine feature extraction methods shown in Figure 2.7, the audio signal analysis techniques have heavily involved frequency-domain feature extractions, which require transforming the waveform audio signal (a) into time-frequency representations (b, c, d, e), which can be performed using the DFT method mostly well-known as a short time Fourier transform (STFT) technique. Moreover, the STFT technique can yield frequency features and time-variant representation by dividing the audio recording into short segments [27, 28]. The traditional feature extraction methods for the SED system commonly used Mel-frequency cepstral coefficients (MFCCs) shown in Figure 2.7 (e), which was used in more than half of submissions of SED task of the DCASE 2016 Challenge for real-life audio[3]. However, the recent trend has moved toward the log-mel spectrogram, or log-mel energy [19, Chap. 3, 13, 31]. The log-mel energy, Mel spectrogram, and MFCCs could be seen as methods that mimic the human auditory system since they have been processed by applying triangular shapes of Mel frequency bands, scaled to imitate the non-linear perception of human hearing critical frequencies. As seen in (d) Mel spectrogram, the lower frequency range is more emphasized and unevenly spaced than (b) linear-frequency power spectrogram with evenly spaced frequency ranges from 0 to 10 kHz. Therefore, those perceptually motivated feature extraction methods have proven to impact robust SED system performance by discarding higher frequency information and larger frequency space for frequency bands closely related to human hearing perception [50, 51].

Generally, the feature extraction process for the time-frequency domain methods begins with frame blocking and windowing of the input waveform audio data. The primary purpose of this stage is to preserve the temporal information when transforming the audio signal into the frequency domain. It could be done by splitting

[3] `https://dcase.community/challenge2016/task-sound-event-detection-in-real-life-audio`

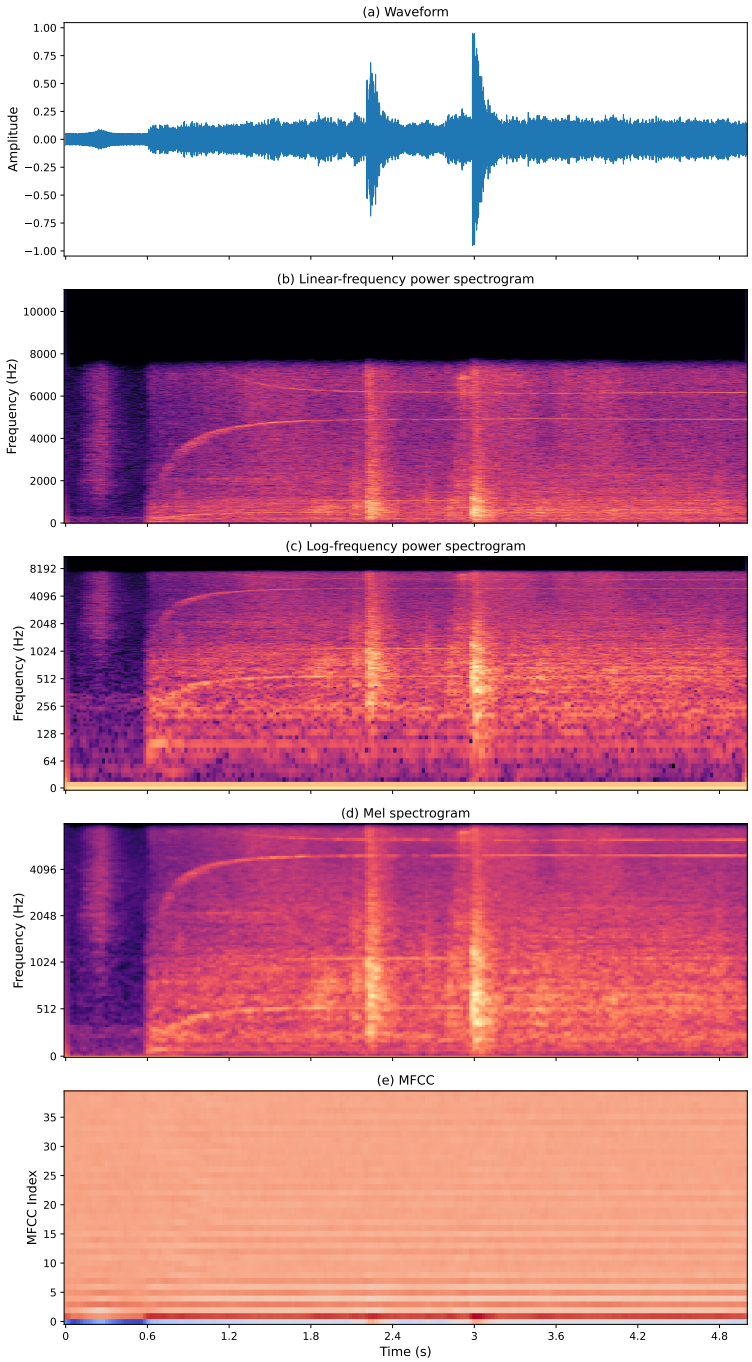

Figure 2.7 *Various time-frequency domain feature extraction methods obtained from a 5-second audio clip taken from the SINS dataset [39], mainly containing vacuum cleaning sound event started approximately 0.5 seconds and lasted until 5 seconds. Figure extraction methods from top to bottom: (a) Waveform audio signal in the time domain, (b) Linear-frequency spectrogram in the time-frequency domain, (c) Log-frequency spectrogram in the time-frequency domain, (d) Mel spectrogram (mel bands=128, hope length = 512) in the time-frequency domain, and (e) MFCC feature representation (number of MFCC = 40).*

the total number of samples into small-length analysis frames and applying window functions to smooth the boundary of the frame signals edge [19, Chap. 2]. For ex-

ample, the YAMNet feature extraction employed a frame size of 0.96 seconds, called patch, and Hann window function by default after resampling the input signal to 16 kHz [8]. If the frame blocking is not performed, the average of the whole input signal will not reflect the time-varying sound information [27]. In addition, a hop size smaller than the desired window size controls the window moving by overlapping with the next consecutive frame, called the overlap-add process, as YAMNet specified a 0.48-second hop size by default [8]. Therefore, using the techniques of frame blocking, windowing, and hop size moving signal analysis, the waveform signals can be transformed by using STFT operation to time-frequency representations.

To elaborate the feature extraction process using mathematical notations, a waveform input signal $x(n)$ with length N can be transformed into a frequency domain using the DFT function for the whole input signal known as the PSD; thus, no time information is extracted. The DFT algorithm can be written as follows:

$$X_{freq}[k] = \sum_{n=0}^{N-1} x[n] \mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}nk}, \tag{2.2}$$

where output in magnitude of frequency components $X_{freq}[k]$ frequency indices of $k = 0, 1, ..., N-1$ corresponding to evenly dividing the unit circle in $[0, 2\pi]$ into N intervals according to time domain N length of input signal samples denoted as $x[n]$, with $n = 0, 1, \ldots, N-1$ [28, Chap. 3, 29, Chap. 3]. The DFT function in equation 2.2 can be rewritten as a function of *cos* and *sin* in the complex sinusoidal plane:

$$\begin{aligned} X_{freq}[k] &= \sum_{n=0}^{N-1} x[n]\{cos(\frac{2\pi}{N}nk) - j\, sin(\frac{2\pi}{N}nk)\} \\ &= \sum_{n=0}^{N-1} x[n]\, cos(\frac{2\pi}{N}nk) - j \sum_{n=0}^{N-1} x[n]\, sin(\frac{2\pi}{N}nk), \end{aligned} \tag{2.3}$$

where this equation indicates that the coefficients of the DFT are the complex numbers of the real and imaginary unit representing the magnitude of the sinusoidal frequency present in the given input signal denoted as $x[n]$ [28, Chap. 3]. Then, the power distribution of the DFT in equation 2.2 and 2.3 is called the power spectral density (PSD), which can be calculated by taking the power of the absolute value of the DFT as follows [30]:

$$PSD = |X_{freq}[k]|^2 \tag{2.4}$$

On the other hand, the STFT feature extraction method provides more useful information than the DFT method by using the frame blocking and windowing techniques

applied to the input signal, which are split with a short length of the desired frame size and moved by the specified hop length. The most commonly used windowing functions are Hann, Hamming, or Blackman window functions used to smooth the signal and prevent discontinuity between consecutive frames [48]. And every short windowed frame is transformed into a frequency spectrum by computing with the DFT function, which in practice it is calculated using a fast Fourier transform (FFT) operation suggested by Cooley and Tukey [52]. The FFT is known as the most efficient way of computing DFT by reducing the computational complexity of the DFT from N^2 to $NlogN$ [52, 53]. The STFT technique to apply the FFT operation to each short frame [54] could be expressed as:

$$X[k,m] = \sum_{n=0}^{N-1} \omega[n]x[n+mH]\mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}nk}, \tag{2.5}$$

where $X[k,m]$ as time-frequency matrix obtained for frequency components of $k = 0, 1, ..., N-1$ for temporal frames $m = 0, 1..., M-1$ by multiplying the DFT function and a particular window function $\omega[n]$ to a N length of frame data samples $x[n]$, with $n = 0, 1, ..., N-1$, starting at m^{th} hop position [48]. The term spectrogram is defined as a matrix representing STFT-based time-frequency energy features in 2-dimension, as shown in Figure 2.7 (b), where time frames are shown in the x-axis and frequency bands in the y-axis [48, 49].

The STFT operation using the equation 2.5 is considered linear frequency representation because frequency bins are equally spaced. However, the human auditory system employs a non-linearity of frequency scales; therefore, mel-filterbank could be utilized to change the linear frequency scale suggested in [55] to mimic the frequency of human perception [48, 49]. Because the human auditory system is more sensitive to low-frequency bands than high-frequency bands, the mel-filterbank, as a triangular shape, takes narrower bands in the low-frequency and wider bands in higher frequency to map with frequency Hertz. It converges 1000 mel frequency to 1000 Hertz frequency. The mel scale conversion from the linear frequency in Hertz is computed as follows:

$$M = 2595\, log_{10}\,(1 + \frac{f(Hz)}{700}), \tag{2.6}$$

where M refers to the frequency with the Mel scale frequency, converted from $f(Hz)$ represented frequency in Hertz by multiplying it with the constant value and taking the logarithm [55]. The triangular shape of the mel-filterbank can be designed to convert a specified frequency range in Hz to the evenly spaced number of mel bands that are usually application-dependent parameters. For example, YAMNet

uses 125 Hz and 7500 Hz as the minimum and maximum frequency range to be converted to 64 mel bands [8]. Mel-scale filterbank magnitude response, called mel spectrogram, for the STFT frame denoted as $X[k, m]$ in equation 2.5 can be computed as follows [50, 56]:

$$S_{mel}[k, m] = \sum_{i=0}^{N-1} |X[k, m]|^2 \Psi_i(k), \tag{2.7}$$

where $S_{mel}[k, m]$ mel spectrogram matrix for the STFT magnitude of a frame at m^{th} position is the sum of all N number of mel-band energy with index i, $i = 0, ..., N-1$, by multiplying the i^{th} triangular-shaped mel-filterbank denoted as $\Psi_i(k)$ to the STFT power spectrum. For example, the mel spectrogram is illustrated in Figure 2.7 (d).

The log-mel spectrogram is a powerful representation widely used for the recent SED systems [49, 57], which employ advanced deep neural networks and also YAMNet pre-trained model [8]. The log-mel spectrogram takes the log magnitude of the mel spectrum defined as follows:

$$S_{logmel}[k, m] = 10 \log_{10}(S_{mel}[k, m]) \tag{2.8}$$

Finally, the MFCC, as shown in Figure 2.7 (e), has been used for speech recognition and music classification for decades. It is extremely effective for traditional classifiers like Gaussian mixture models (GMMs), thus, used for the GMM model baseline of the TUT-SED2016 dataset [47]. The MFCC is in condensed form, decorrelated the log-mel spectrum using Discrete Cosine Transform (DCT) [27]. The MFCCs could be computed as follows:

$$M_{fcc}[c, m] = \sqrt{\frac{2}{K}} \sum_{k=0}^{K-1} S_{logmel}(k) \cos(c(k - \frac{1}{2})\frac{\pi}{K}), \tag{2.9}$$

where K is the number of Mel frequency bands, k=0, 1,..., K-1 is mel frequency band index, and $S_{logmel}(k)$ is the log mel energy of m^{th} frame and c is the index of the cepstrum coefficient, c=0,1,...,C-1 [48, 58].

2.3 Evaluation Methodology for Sound Event Detection

This chapter discusses datasets especially common for the SED system development for homecare applications and used for performance evaluation. It is important to understand and carefully select relevant benchmark datasets for the SED system development that can lead to the transparency of the system's performance. Therefore, it could be objectively compared the performance of the system to other systems in the same research field. After the dataset selection, standardized SED system evaluation methodology, performance metrics, and SED toolbox used in the DCASE challenges for generating automated SED system performance metrics are introduced and further discussed.

2.3.1 Benchmark Datasets

Audio benchmark datasets are packaged mainly with audio data and metadata (annotation, reference, or ground truth). Audio data contains audio recording files, on the other hand, the metadata provides sound event labels for the audio recordings. The freely available audio benchmark datasets focused on domestic sound events are listed in Table 2.3. The sound event labels are descriptions of sound events presented in audio recordings. Because labeling is somewhat subjective, combining the sound source agent and action descriptors is often suggested to avoid misinterpretation of the sound events [19, Chap. 6].

Depending on the benchmark dataset's purpose or task to solve by the system, the metadata can be provided with single or multiple class labels with or without its time boundary information. Firstly, it can be categorized as a weakly and strongly labeled dataset depending on the existence of the temporal information in the reference file. The weakly labeled dataset without onset and offset of the time boundaries is considered relatively easy to produce, for example, UrbanSound [59], UrbanSound8K [59], and ESC-50 [60], datasets mentioned in Table 2.3. It is used mostly in sound scene classification for single class detection and sound event tagging for predicting none or more than one presence of multiple class labels [61]. On the contrary, the strongly labeled dataset with time boundaries, such as the events' onset and offset, is commonly used for SED system development. Still, it is relatively difficult to produce regarding the time and effort of annotations, like the TUT-SED2016 dataset [47]. It also has some limitations on the ambiguity of class labeling and setting time boundaries due to different human perceptions from multiple annotators [19, Chap. 6].

Furthermore, as seen from the view of the overlapping sound events in a temporal audio recording segment, the strongly labeled dataset can be categorized into monophonic and polyphonic annotations. The polyphonic annotation method could

represent mostly occurring in real-life situations where multiple sound events overlap but have a high degree of challenge to produce, for instance, TUT-SED2016. Conversely, the monophonic annotation should mark only the most prominent sound event in the temporal boundary of each audio segment [19, Chap. 6], the case used in the ESC-50 dataset.

Table 2.3 *A list of freely available benchmark datasets for the SED system development and evaluation considering the homecare application. Types of the dataset; Collected refers to data collected from available audio file repositories, while Recorded refers to the dataset produced by recording newly in sound fields.*

Dataset	**Type**	**Annotation**	**#Classes**	**Tot. Events**	**Size(min)**
UrbanSound [59]	Collected	Strongly-labeled (monophonic)	10	3075	1620
UrbanSound8K [59]	Collected	Strongly-labeled (monophonic)	10	8732	525
ESC−10 [60]	Collected	Weakly-labeled	10	400	33
ESC−50 [60]	Collected	Weakly-labeled	50	2000	167
TUT-SED2016 [47]	Recorded	Strongly-labeled (Polyphonic)	18	954	~ 78
AudioSet [62]	Collected	Weakly-labeled	527	$> 2M$	$> 350K$
SINS Database [38, 39]	Recorded	Weakly-labeled	9	72984	$\sim 12K$
DESED [31] (Real recordings)	Collected	Weakly-labeled	18	954	~ 550

Nevertheless, as stated, producing the strongly labeled is the most complex and expansive work. Therefore, there is a shortage of datasets that come with strongly labeled. Furthermore, there are claims that the small size of the strongly annotated dataset, such as the TUT-SED 2016 dataset, is insufficient for training deep neural networks [6, 36, 37]. The collected with weakly-labeled metadata has become a trend in recently organized DCASE challenges because they could reduce the burden of producing the newly recorded dataset from the field, such cases like the SINS database [39] and TUT-SED2016. It could be produced relatively large-scale datasets by collecting audio samples from freely available audio collection repositories. For instance, Freesound audio clips tailored for ESC-10, ESC-50, UrbanSound, UrbanSound8K, or YouTube videos extracted for AudioSet [62]. The large scale of collected datasets could bring benefits of generalizing the SED system learning with various sources of sound. However, they might have drawbacks of low quality in labeling and sound recording [6, 19, Chap. 6].

2.3.2 Datasets for Evaluation

It is essential to provide suitable audio datasets to the supervised machine learning algorithms because they learn from the feature extracted from audio data. Publicly available benchmark datasets are crucial in comparing different algorithms and

supporting the research process to develop machine learning models. When selecting datasets for the neural network model development, three properties must be considered: coverage, variability, and size [19, Chap. 6]. The coverage means how many different categories are included in the dataset, and these categories should be relevant to the application's use cases. The variability implies that audio recordings should be generated in many conditions, considering capturing diverse acoustic characteristics for each category. It is because the acoustic characteristics can usually be inevitably dissimilar, even though sound events belong to the same category. For instance, the sounds of closing a door depends on the type of door material, the mechanical impact of the door, the size of the door, locations, etc. Finally, the dataset size should be considered as they have a balanced number of sufficient samples for each category. Therefore, selecting benchmark datasets considering the three properties for the application area, training the SED model, and conducting the performance evaluation will help achieve a robust and well-generalized SED algorithm development.

The following part of this section describes more details about the benchmark SED datasets presented in Table 2.3 with respect to this thesis.

AudioSet Dataset

AudioSet dataset[4] released in 2017 is an extensive dataset containing 527 sound classes collected from YouTube videos with approximately 350k minutes of audio taken from over 2 million videos, which might cover most of the existing sound categories in the urban environment [62]. The events are 10-second segments extracted from YouTube videos labeled by human annotators providing one or multiple sound event labels without start and end time envelopes. The dataset's performance trained with the baseline system showed that average precision with the 485 event classes obtained 0.314. The best precision score was the "Music" class with 0.896, while the "Rattle" class gained the worst precision score with 0.020 [62]. Despite the large scale of the AudioSet dataset, some drawbacks have been indicated: low quality of annotation and the dataset not being shared as waveforms but feature extractions [6].

The AudioSet research presented a hierarchically structured ontology categorized for sound events experienced in real-world recordings. In this way, human annotators involved in the research project could immediately find the corresponding labels to given audio events. An overall view of urban sound taxonomy from a homecare perspective is presented in Figure 2.8. The ontology hierarchy is also limited to the

[4] `https://research.google.com/audioset/`

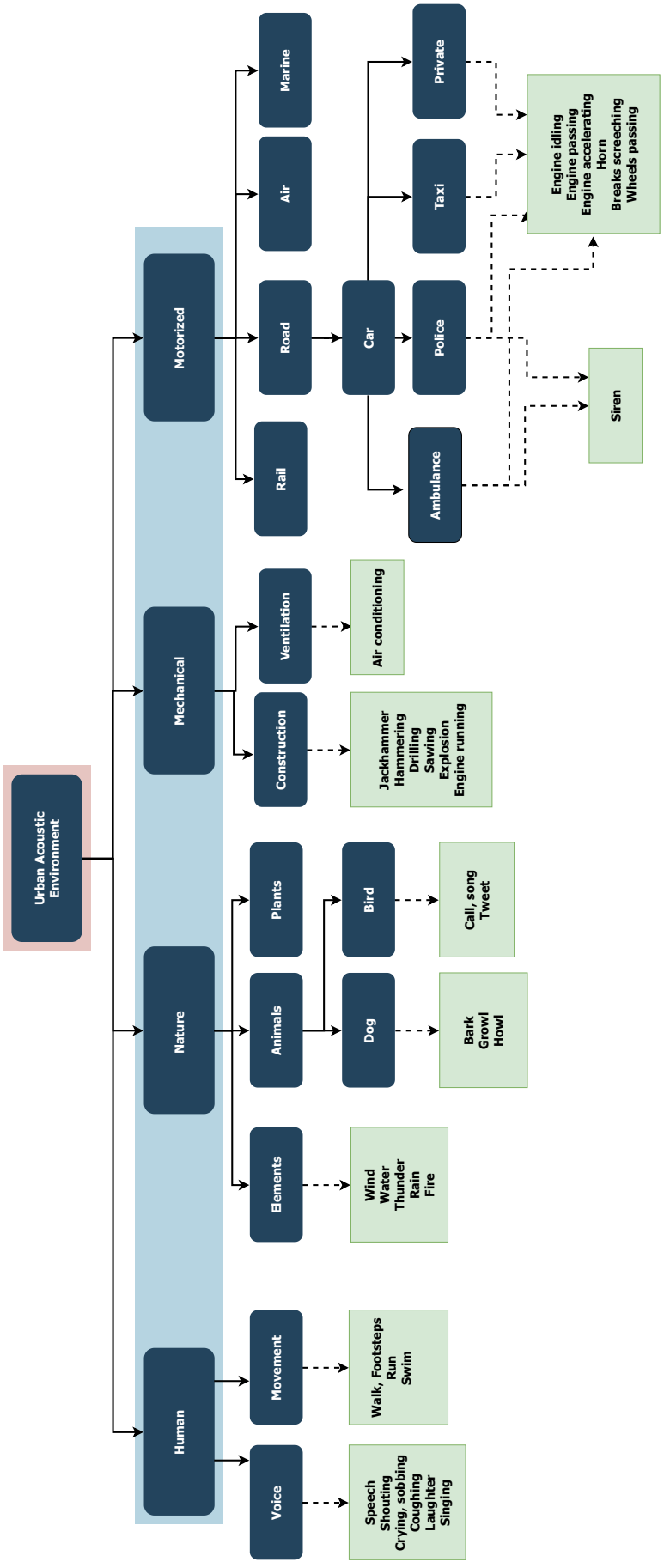

Figure 2.8 *An urban sound ontology. Adapted from [8, 59, 62].*

maximum depth at six levels for rapid scanning, based on earlier research on the Urban Sound Taxonomy [59]. As an example, the leaf node of "Ambulance siren" can be found, followed by the node of "Sound of things" to child nodes "Vehicle," "Motor vehicle," "Emergency vehicle," and "Siren." Moreover, the research findings suggest that the hierarchical approach is flexible for ambiguous classes. If sound event detection algorithms yielded a low performance with the leaf nodes, their parent node could be used as a target class such that the "Dog" category can be used instead of "Bark," "Growl," and "Howl," as illustrated in Figure 2.8.

The research released ontology data, with six fields describing each category: Knowledge Graph Machine ID (MID), display name, description, examples, children, and restrictions. From these fields, the first two fields are taken into use in the YAMNet class-map [8]: MID and display name. MID is the primary identifier, and the display name is the actual class label, which can be one descriptor or several acoustic event descriptors with comma-separated [62].

ESC-50 Dataset

A publicly available ESC-50 dataset[5], summarized in Table 2.3, consists of 2000 5-second-long audio clips with a tagging label for each audio file [60]. The recordings are organized into five folds in every 400 files for cross-validation. Each file contains one of 50 event classes categorized into five environmental sound groups: human non-speech sound, domestic sound, animals, urban noises, and natural sound [60]. The dataset is more suitable for benchmarking the sound classification system; however, it can be used to some extent for the SED system benchmark. The advantage of using the dataset is that the audio clips are well balanced, thus 40 clips for each of the 50 classes. Furthermore, most target classes could be useful for the SED system to detect critical activities of a home environment, such as 'Door knock,' 'Toilet flush,' and 'Clock alarm.'

The classification performance using the ESC-50 dataset has been reported using many different machine-learning methods on the dataset's public site. Initially, a baseline applied to a random forest ensemble method was achieved at 44.3% accuracy [60]. Moreover, the additional baseline of a CNN model with a data augmentation method [63] improved the performance by approximately 20% higher than the previous random forest model. Furthermore, recent studies developed with various machine learning techniques have announced their significant performances evaluated on the ESC-50 dataset, which are noticeably higher than 90%, one of them, as a multi-stage sequential learning model suggested in [64].

[5] `https://github.com/karolpiczak/ESC-50`

TUT-SED 2016 Dataset

TUT Sound Events 2016 (TUT-SED2016) dataset[6] [47] was recorded in a real-world environment and introduced for the SED task in the DCASE Challenge 2016 [5]. The dataset has two acoustic scenes: one from a residential area and another from an indoor home environment. The dataset can be utilized for developing application areas such as security and home surveillance. The dataset's acoustic properties achieved a high variability and quality by recording with binaural audio in-ear microphones using a 44.1 kHz sampling rate and 24-bit resolution captured audio events from various locations, including many homes. The recording duration varies from 3 to 5 minutes resulting in a total duration of approximately 78 minutes, as shown in Table 2.3. However, the TUT-SED 2016 Home scene dataset might not be appropriate for training machine learning algorithms due to the limited size of the audio recordings and class-wise unbalanced instances, as summarized in Table 2.4 [5].

The dataset consists of 18 target sound event class labels, of which 7 classes are for the residential area, and the other 11 classes are for the home events used to evaluate the proposed model performance. For example, the home dataset contains unbalanced event instances of 47 'water tap running,' 250 'object impact,' and 151 'dishes,' as summarised in Table 2.4. The dataset's reference provides time-wise overlapping sound events strongly labeled as polyphonic annotations with the starting and ending.

Table 2.4 *Table presents a list of target classes and instances used in two datasets: TUT-SED 2016 Home dataset for DCASE 2016 Challenge Task 3 and SINS dataset for DCASE 2018 Challenge Task 5.*

Dataset	**TUT-SED 2016: Home [5]**		**SINS: DCASE 2018 Task5 [38]**	
	Target Class	**Instance**	**Target Class**	**10s segment**
	(object) rustling	60	Absence	18860
	(object) snapping	57	Cooking	5124
	cupboard	40	Dishwashing	1424
	cutlery	76	Eating	2308
	dishes	151	Other	2060
	drawer	51	Social activity	4944
	glass jingling	36	Vacuum cleaning	972
	object impact	250	Watching TV	18648
	people walking	54	Working	18644
	washing dishes	84		
	water tap running	47		
Total	**11 classes**	**906**	**9 classes**	**72984**

The baseline system for measuring the TUT-SED 2016 dataset performance suggested employing a Gaussian mixture model (GMM) classifier with features extracted from MFCC computation [47]. The overall micro average performance was

[6] `https://zenodo.org/record/45759#.YCrtAGj7Sbg`

quite poor at 18.1% F1 and 0.95 ER. And the class-wise performance of three out of the 11 target events was reported with no true positive detection, such as '(object) snapping' and 'glass jingling.' However, the highest F-score was obtained by the "water tap running" class at 41.2%, followed by 'washing dishes' at 26.4% [5, 47].

SINS Database

The SINS database contains activities produced by a person in a vacation home for a week, which scenario was based on the assumption that the dataset could be used to enhance health and security monitoring for elderly people who require healthcare professionals' support in the nursing home environment [39]. Thus, the dataset contains one week of real-life audio recorded using a sensor technology distributed to five rooms with 13 sensor nodes. Each node contained a linear microphone array consisting of four microphones set with a 16 kHz sampling rate. A subset of the SINS database from the living and kitchen rooms utilized in the DCASE 2018 Challenge Task 5[7] as a development dataset for classifying the acoustic events [38]. In terms of class-wise instances, the development dataset is highly unbalanced with 9 target classes, as presented in Table 2.4, including the most dominated classes of 'Absence,' 'Watching TV,' and 'Working,' while the least instances of 'Vacuum cleaning' and 'Dishwashing.' The dataset, as presented in Table 2.3, provides audio recordings with a 10-second duration and corresponding ground truth of a class label per each audio segment, resulting in a total audio recording duration of approximately 200 hours, considered a large-scale dataset.

The DCASE 2018 Challenge Task5 baseline system employed a simple architecture of CNN 2 convolutional layers with dense layers to output 9 classes based on Softmax activation [38]. The baseline for the SINS dataset was evaluated class-wise, and the macro average of all classes in the F1-score metric. The overall macro averaged F1-score was at 84.50%, contributed by the top three classes over 95% F1-score, such as 'Watching TV' at 99.59%, 'Vacuum cleaning' at 99.31%, and 'Cooking' at 95.14%. On the other hand, the least F1-score obtained by 'Other' at 44.76%.

DESED Dataset

The DCASE 2019 Challenge for Task 4 introduced the domestic environment sound event detection (DESED) dataset[8]. The DESED dataset consists of two datasets: a subset of the AudioSet dataset with weakly labeled and a subset of the synthetic dataset that is strongly labeled [31]. The two subsets of audio clips contain single

[7] https://zenodo.org/record/1247102#.YClzM2j7Sbh

[8] http://dcase.community/challenge2019/task-sound-event-detection-in-domestic-environments

or multiple sound events that can be occurred in the urban domestic environment. The target detection classes are Speech from the human voice, Cat and Dong animal sounds, and four different mechanical sounds classified as Vacuum cleaner, Electric shaver/toothbrush, Blander, Alarm bell ringing, and other sound sources of Dishes and Frying. Although the first audio subset (called real recordings) is weakly labeled due to no time boundary information, the dataset volume of approximately 23 GB might be large enough to train the SED system. Furthermore, the strongly annotated but smaller size of synthetic data (1.8 GB) suggested improving the SED system's performance during training [31]. It was emphasized that the winning team of DCASE 2019 Challenge Task 4 advanced 10 % of their SED system performance by comparing the best submission of the DCASE 2018 challenge Task4[9], which used only the real audio recordings without the synthetic audio subset [31].

2.3.3 Evaluation Metrics and Toolbox

As the last phase of the SED system development, it is essential to evaluate its performance using standardized evaluation methods. However, the evaluation methods should be considered at the beginning phase of SED system design, not afterward. The purpose of the evaluation methods can be first to benchmark the system and second to identify performance gaps to improve SED classifiers. The benchmark requires discriminative rules that can be used to analyze the performance of different machine learning classifiers, which should be trained using the same dataset [65, p.444, 66].

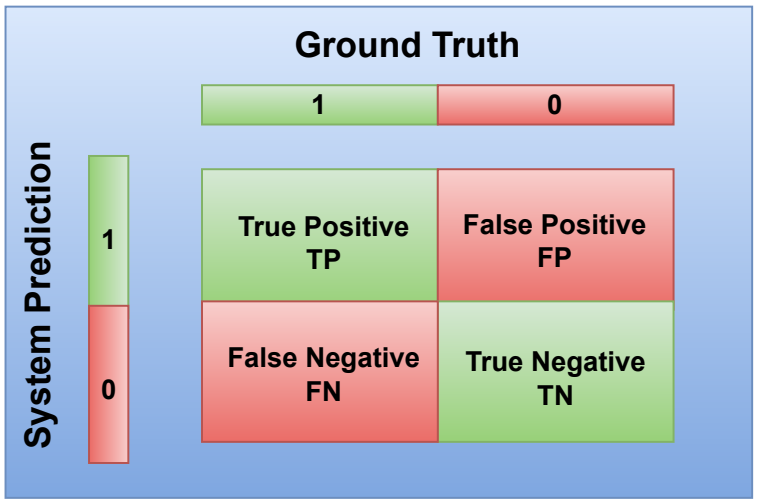

Figure 2.9 *Classifier confusion matrix consists of four system prediction cases resulting by comparing it against the ground truth.*

The SED performance metrics are driven by comparing the system prediction to the

[9] https://dcase.community/challenge2018/task-large-scale-weakly-labeled-semi-supervised-sound-event-detection

ground truth. Figure 2.9 illustrates the confusion matrix, also called contingency table, that represents possible comparison combinations categorized into a 2 x 2 matrix [19, Chap. 6, 65]. The prediction of the SED classifier for a given sound event class can be either positive (1) or negative (0), and the same for the ground truth resulting in four categories of the matrix table named true positive (TP), false positive (FP), true negative (TN), and false negative (FN). Accumulating these four case statistics for each class can yield a class-wise performance evaluation known as macro-average and be globally summed to get overall performance known as overall micro-average [46, Chap. 6, 19, Chap. 6]. The four categories can be explained as follows:

- True positives (TP): the system correctly predicted the given event as positive as the corresponding event in the reference marked as positive.
- False positive (FP): the system output is falsely predicted as positive in contrast to the ground truth marked in negative.
- True negatives (TN): the system output predicted the given event negative correctly aligned with the reference marked in negative.
- False negative (FN): the system output incorrectly predicted the given event negative against the reference marked in positive.

The most commonly used standardized SED evaluation methods derived from the confusion matrix are F_1, or $F1 - score$, and $Errorrate$ (ER). In principle, the metric calculation is based on a fixed size of audio recording called a segment. By default, it uses a one-second segment to create class-wise confusion metrics, and overall metrics, called micro-average [19, Chap. 6, 66]. To calculate the F1-score can be computed with precision and recall beforehand. Let the $precision(P)$, as known as a positive predictive value, be computed by:

$$P = \frac{TP}{TP + FP}, \tag{2.10}$$

where the total number of true positives that are divided by the sum of predicted condition positive. And the $recall$ (R), as known as sensitivity and a true positive rate (TPR), is calculated as:

$$R = \frac{TP}{TP + FN}, \tag{2.11}$$

where the total number of true positives is divided by the sum of the true condition

positive. Then F1-score based on precision and recall is computed as:

$$F_1 = \frac{2 \cdot P \cdot R}{P + R}. \tag{2.12}$$

Although F1-score is widely recognized as for SED performance metric, some limitations are encountered. For instance, in most real-life audio recordings having a high likelihood of unbalanced event classes, F1-score will be calculated by averaging across the event classes. Consequently, F1-score gained from the circumstance might be over-optimistic or under-estimated by classes having many instances [19, Chap. 6]. Similarly, recent research has argued that the MCC matrix, see Equation 3.4, can give a better-quantified evaluation for imbalanced data than $F1$-score because $F1$-score does not consider TN whereas MCC uses it in the formulation [67, 68]. Therefore, the overall score and class-wise performance should be assessed to get every aspect of performance evaluation, especially for real-life imbalanced event classes.

For certain use cases, when precision is not equally important as recall, F-score allows adjusting the weight by substituting α by greater than 1. It can be defined as follows:

$$F_\alpha = (1 + \alpha^2) \times \frac{P \times R}{(1 + \alpha^2) \times P + R} \tag{2.13}$$

For example, missing the tumor detection (FN) is more severe than a false alarm (FP) to a healthy sample in the medical domain. If we want to weigh Recall as twice as important as precision, by setting $\alpha = 2$, we can obtain $F2$-score:

$$F_2 = (1 + 2^2) \times \frac{P \times R}{(1 + 2^2) \times P + R} \tag{2.14}$$

Accuracy (ACC) is also commonly used to measure sound classification or detection system performance, which can be calculated as the sum of correctly identified instances of positives and negatives divided by the total population. However, accuracy might be misleading if the class imbalance has a few TP and TN predictions [19, Chap. 6]. It is defined as follows:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}, \tag{2.15}$$

Errorrate (ER) represents the portion of errors produced by the SED system associated with the ground truth. It is an accumulative calculation based on three measurements: substitution (SUB), deletion (DEL), and insertion (INS) [19, Chap. 6, 66]. In the segment-based error rate calculation, Substitution

is the number of events from the reference that are incorrectly detected. This means the system produced no true positives but FN or FP. Substitution takes one of the minimum values by comparing the two values. After substitutions are calculated, the remaining events wrongly detected are counted by Deletion and Insertion. Deletion calculates the number of reference events wrongly not detected, while Insertion counts for the wrongly detected system output:

$$\begin{aligned} SUB(j) &= min(FN(j), FP(j)) \\ DEL(j) &= max(0, FN(j) - FP(j)) \\ INS(j) &= max(0, FP(j) - FN(j)), \end{aligned} \tag{2.16}$$

where j is the number of segment $j = 1, 2, , N$, incorrect detection of FN and FP for each segment are counted and computed for SUB, DEL and INS.

The segment-wise overall $ErrorRate$ is quantified across the total number of segments measured with earlier defined three parameters of SUB, DEL, and INS. The error rate can be calculated using the following mathematical formulation:

$$ER = \frac{\sum_{j=1}^{N} SUB(j) + \sum_{j=1}^{N} DEL(j) + \sum_{j=1}^{N} INS(j)}{\sum_{j=1}^{N} SEG(j)}, \tag{2.17}$$

where N indicates the total number of segments, $SEG(j)$ accumulates the number of active sound events in the j^{th} segment of the reference. Because the ER is a score rather than a percentage, it is sometimes difficult to interpret. For example, the ER value of 1 might obtain when the SED system detects no events. Therefore, ER should support the $F1$-score to determine the SED system performance. To obtain as less as possible, a ER score below 1 and close to zero should be targeted for a competitive SED system performance [19, Chap. 6].

SED Evaluation Toolbox

Evaluation for the SED system to be standardized primarily for the DCASE challenges. For this purpose, an evaluation toolbox called `sed_eval` [66, 69] is provided and widely used in the DCASE challenges for SED system evaluation. The `sed_eval` is open-source software freely downloaded, and its tutorials are available from this website[10]. The toolbox provides two types of metrics that can evaluate the performance of SED systems; segment-based metrics and event-based matrices. Users can define the evaluation's time resolution; otherwise, it will use one second as default. Firstly, the segment-based evaluation metrics compute every four categories of the confusion metrics by comparing the system predictions with the reference

[10] `https://tut-arg.github.io/sed_eval/`

or ground truth for each time resolution. After that, the statistics are aggregated overall data and produce a micro-average metrics value, as all the instances have the same weight. In addition, class-wise metrics, also called macro-average, are produced from intermediate statistics aggregated for each sound event class, which is beneficial information to understand the system output for in-depth analysis of the class level, especially for unbalanced class instances in a dataset. Secondly, the event-based metrics are an event instance level of measurement compared between the system output and the ground truth divided by the desired event duration [66].

3 Methods

This chapter consists of three aspects of the YAMNet-based proposed system: Chapter 3.1 describes the challenges discovered from the YAMNet model for adapting to a SED model during the investigation, Chapter 3.2 discusses the Y-MCC methodology for the SED system how it has designed to overcome the YAMNet challenges explaining in procedure and process levels, and Chapter 3.3 describes the core part of the statistical method of MCC-based YAMNet class mapping techniques.

3.1 Challenges with the Pre-trained YAMNet Model

This thesis aims to develop a polyphonic SED solution for homecare applications using the pre-trained YAMNet classifier. YAMNet based on MobileNetV1, introduced in Chapter 2.2.2, has many advantages. The essential benefit of the YAMNet classifier trained on the large AudioSet dataset is that it can predict 521 different sound classes, especially including most of the sound events that possibly happen in the home environment.

However, the early stage of the YAMNet investigation revealed the critical drawbacks of YAMNet for utilizing it for the SED system task. The first challenge was that YAMNet high-scored event predictions were often more likely to provide classes describing high-level sound scenes, which might not be necessary for the SED system running in the indoor home environment. This issue caused more important sound events to be scored with a lower probability resulting in low SED system performance. For example, the problem occurred in most of the ten audio recordings in the TUT-SED2016 Home Development dataset. As presented in Table 3.1, the two highest YAMNet predictions of most of the audio files were 'Silence' and 'Inside, small room.' These predicted labels considered as general high-level scene classes might be redundant information. These predictions were the main bottlenecks for achieving the excellent performance of the SED system designed to detect prominent sound events occurring primarily in rooms in quiet environments. Therefore, it could suggest selecting lower-level class predictions rather than the high-level context descriptor to improve the SED performance in this case.

The second challenge of the YAMNet prediction for a SED system was that multiple correlations existed between many YAMNet classes and target reference classes, which caused hard to map by human effort due to some degree of illogical and complicated. It could be seen as a set of YAMNet class labels to be combined to detect a specific target class. Table 3.2 explains this YAMNet prediction phenomenon

Table 3.1 *The top 3 class events detected by YAMNet with the highest probability score toward audio recordings are included in the TUT-SED2016Home dataset. The observation indicated an issue of YAMNet that the two top predictions, such as ' Silence' and 'Inside, small room,' of all audio files seemed unnecessary for the SED system to detect prominent sound events occurring indoor environments.*

Audio File	**YAMNet Top-3 Predictions**
a030.wav	'Silence', 'Inside, small room', 'Mechanisms'
a031.wav	'Silence', 'Speech', 'Inside, small room'
a034.wav	'Silence', 'Inside, small room', 'Door'
a036.wav	'Silence', 'Inside, small room', 'Writing'
a038.wav	'Silence', 'Inside, small room', 'Door'
b029.wav	'Silence', 'Inside, small room', 'Door'
b030.wav	'Silence', 'Inside, small room', 'Dishes, pots, and pans'
b032.wav	'Silence', 'Inside, small room', 'Sizzle'
b033.wav	'Silence', 'Inside, small room', 'Speech'
b044.wav	'Silence', 'Inside, small room', 'Clock'

extracted from DCASE 2019 Task4 train audio dataset, refer to DESED dataset Chapter 2.3.2 with weak labels. As it shows, YAMNet predictions of 'Air horn, truck horn,' 'Buzzer,' and 'Alarm' were related to the reference label of 'Alarm bell ring,' which might be a reasonable inference. By contrast, YAMNet prediction classes of 'Music,' ' Jingle, tinkle,' and 'Music for children' were cases that were hard to map with Cat and Dog reference labels. The same issue was observed with the SINS database as shown in Table 3.3. For example, with the SINS dataset, the reference class of 'Vacuum cleaner' was predicted by YAMNet with 'Aircraft,' 'Helicopter,' and 'Jet engine' along with 'Vacuum cleaner,' considered as challenging to find corresponding YAMNet classes.

Table 3.2 *Examples of YAMNet predictions against 10-second audio files are included in the DESED dataset. It shows the level of inaccuracy challenging to map between YAMNet classes and the target reference labels.*

Reference Labels	**YAMNet Top Predictions**
Alarm_bell_ringing	'Air horn, truck horn', 'Buzzer', 'Alarm', 'Vehicle horn'
Blender	'Aircraft', 'Blender', 'Jet engine', 'Vehicle'
Blender,Speech	'Aircraft', 'Vacuum cleaner', 'Tools', 'Helicopter'
Cat,Dog	'Music', 'Jingle, tinkle', 'Music for children'
Dishes	'Animal', 'Wild animals', 'Bird', 'Inside, small room'
Electric_shaver_toothbrush	'Animal', 'Alarm clock', 'Inside, small room', 'Buzzer'
Frying	'Boiling', 'Liquid', 'Frying (food)', 'Patter', 'Animal'
Frying,Speech,Dishes	'Frying (food)', 'Animal', 'Sizzle', 'Crumpling'
Running_water	'Animal', 'Bird', 'Wild animals', 'Bird vocalization'
Vacuum_cleaner	'Blender', 'Hair dryer', 'Vacuum cleaner', 'Tools'

In summary, the problems with YAMNet predictions toward the real-life audio datasets would require tremendous time and effort to find corresponding YAMNet classes to the target classes of the selected benchmark datasets. The observed challenges indicated finding a solution by computing the correlation coefficients between the 521 YAMNet class labels and the given target classes. Moreover, the evidence showed that correlation computation for finding class maps between YAMNet classes and the target reference classes should be done for each dataset. Therefore, to tackle the challenges of adapting the YAMNet model for the SED task, specifically with real-life audio recordings, we propose an entirely new approach for YAMNet class mapping, called the Y-MCC methodology, a reliable YAMNet class mapping based on Matthews correlation coefficients (MCC). The following chapter explains details about the Y-MCC methodology.

Table 3.3 *Examples of YAMNet event predictions produced for the SINS database. It is a confusing class prediction that can not be used for a SED system directly but needs to make a proper map for better predictions and better performance of a SED system.*

Reference Labels	**YAMNet Top Predictions**
absence	'Snake', 'Animal', 'Silence', 'Outside, rural or natural'
cooking	'Inside, small room', 'Spray', 'Domestic animals, pets'
dishwashing	'Spray', 'Liquid', 'Hiss', 'Steam', 'Animal', 'Inside, small room'
eating	'Inside, small room', 'Wood', 'Tools', 'Rub', 'Silence'
other	'Inside, small room', 'Door', 'Drawer open or close'
social_activity	'Speech', 'Inside, small room', 'Silence', 'Snake', 'Animal'
vacuum_cleaner	'Vacuum cleaner', 'Vehicle', 'Tools', 'Aircraft', 'Jet engine'
watching_tv	'Snake', 'Animal', 'Music', 'Silence', 'Speech'
working	'Animal', 'Inside, small room', 'Outside, rural or natural'

3.2 Y-MCC Methodology

Y-MCC methodology as the proposed solution has been developed in order to make YAMNet perform better with the SED task for various benchmark datasets recorded in the real-life home environment. It utilized the YAMNet pre-trained DNNs and built on additional stages to automatically produce class maps between YAMNet 521 classes to a set of target classes defined in each benchmark dataset. The development environment of the Y-MCC method was on the CentOS Linux version 7 operating system platform installed Anaconda (release=22.9.0) and Python programming language (version=3.8.13) with signal processing libraries provided by Librosa (version=0.9.2), TensorFlow (version=2.4.1), SciPy (version=1.7.3), NumPy (version=1.19.5), data analysis package using Pandas (version=1.5.2), and many other open source utilities and libraries.

Three benchmark datasets were carefully selected to develop the Y-MCC method;

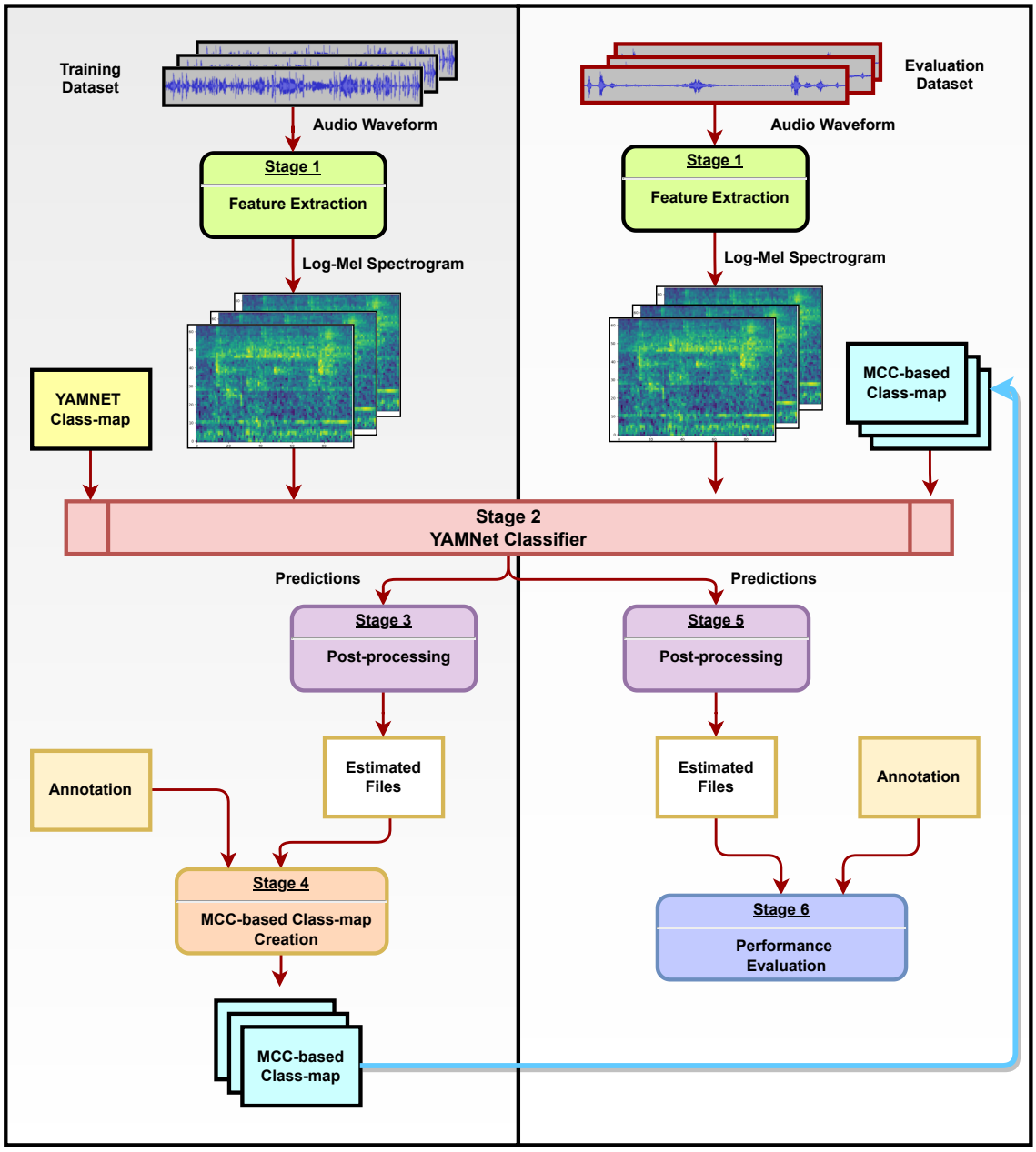

Figure 3.1 *The proposed Y-MCC method consists of a pipeline with two channels for creating MCC-based class maps (Left) and performance evaluation using the MCC-based new class maps generated by the first channel (Right).*

TUT-SED 2016, ESC-50, and SINS dataset, described in Chapter 2.3.2. The three datasets' common property is that they have been recorded from the home environment and frequently evaluated by numerous researchers for their SED system training and evaluation [57]. The datasets available from the DCASE challenges, such as TUT-SED 2016 and SINS dataset, are well-divided subsets for developing and assessing the SED system. Thus, this experiment mostly used a development subset from the datasets firstly to compute the MCC matrix for the MCC-based class maps and another subset for performance evaluation.

The proposed Y-MCC system, as illustrated in Figure 3.1, consists of two channels and six stages. In the first channel, the procedure aims to create class maps based on the MCC computation between the YAMNet prediction using the original 521 class map. After then, in the second channel, replace the original YAMNet class map with the new MCC-based class maps generated from the first channel containing the target class labels. Therefore, the YAMNet predictions can be made based on the target classes defined in the given dataset with temporal event boundaries.

Look more closely, the two channels of the Y-MCC consist of 4 stages. The 4 stages of the first channel (on the left in Figure 3.1) are; (1) Data pre-processing and feature extraction, (2) YAMNet prediction using the original class map, (3) Post-processing, (4) MCC-threshold-based class mapping. On the other hand, the second channel (on the right in Figure 3.1) includes the following stages; Data pre-processing and feature extraction, YAMNet prediction using the MCC-threshold-based class maps, post-processing, and Performance evaluation. The following Chapter 3.3 discusses mainly the stages of the first channel of the Y-MCC, and Chapter 4 discusses MCC-based class mapping and performance evaluation following the stages in the second channel for each 3 selected datasets.

3.3 MCC-based Statistical Methodology for Class Mapping

3.3.1 Pre-processing and Feature Extraction

The Y-MCC SED system core resides in the first channel to create MCC-based class maps implemented through 4 stages using the YAMNet model. In the first stage, data pre-processing and feature extraction are to obtain high-level acoustic features from the raw audio clips. The datasets should be divided so that a sufficient amount of dataset is allocated to this process than the dataset needed for the performance evaluation channel. In general, the ratio of the dataset allocation for machine learning is approximately 60% (70%), 20% (20%), and 20% (10%) for training, testing, and evaluation, respectively. Because the ML models can learn from the data and achieve more generalized parameters, assigning more portions of the training dataset is recommended. However, the YAMNet classifier was already trained; thus, a similar proportion of the dataset for the training can be used to create the MCC-based class maps and the rest of the dataset to evaluate the SED system. Consequently, the class maps generated with the larger subset of the dataset can produce more reliable class maps.

A brief description of Stage 1 is first, all the input audio recordings with the waveform will be down-sampled to 16 kHz if the sampling rate exceeds 16 kHz. After then, all the audio samples are sliced into fixed sizes of frames using the frame-blocking

methods. In this case, the frame size was defined as a one-second length with a 16k sample length of the input audio sample with 50% overlapping moving hop size. For a strongly-labeled dataset like the TUT-SED2016 dataset, a one-second time resolution was defined for processing to the YAMNet classifier to get the prediction score output. These frames are then processed to the YAMNet feature extraction function to produce an image-like 2-D shape of the log-mel spectrograms, as described in Chapter 2.2.3. After then, the log-mel spectrogram is broken down into a stack of fixed-size patches, $\mathbf{x}^p \in \mathbb{R}^{l \times m}$ where l and m denoted number of log-mel samples per patch l=98 and m=64 Mel-bands. Consequently, one patch per input frame size with one second was produced and fed to the YAMNet classifier in the next stage.

This section explains the pre-processing steps common for all three selected datasets, taking an example audio file from the TUT-SED2016 Home development dataset. Three steps were followed in the pre-processing to meet YAMNet's requirements (according to the YAMNet specification discussed in Chapter 2.2.2). First, normalized the wave data to obtain data values between [-1, 1], and next, multichannel recordings of the input data were averaged to get mono channel data. Finally, the averaged-mono channel data was resampled with 16 kHz instead of the original 44.1 kHz sampling rate. After then, the pre-processed data were divided into 1-second time frames with 0.5-second overlapping windows and fed into the data transformation function to change the input waveform to log-mel spectrogram followed by STFT transformation. The transformed 2-D shape of the log-mel spectrogram with 64 Mel bands became the input of the YAMNet CNN model. Figure 3.2 visualize the three different representations of the data transformed during the Y-MCC pre-processing stage. These similar pre-processing steps were also applied to the two other datasets, ESC-50 and SINS dataset, considering their characteristics of audio recording clips.

3.3.2 YAMNet Classifier and Post-processing

In the second stage of the YAMNet classifier, the patch obtained from the previous feature extraction stage is fed into the YAMNet classifier. Then the classifier returns a row vector of scores with a probability value for every 521 multi-label computed by:

$$P(\hat{y}^p|\theta) = \theta(z) = \frac{1}{1 + e^{-z}}, \tag{3.1}$$

The probability prediction of $\hat{y}^p$ for a patch is computed by the acoustic model of the classifier θ, obtained by calculating net input z applied to the sigmoid function.

$$P(\hat{y}^p) \in \mathbb{R}^{1 \times n} \in [0, 1]^p, \tag{3.2}$$

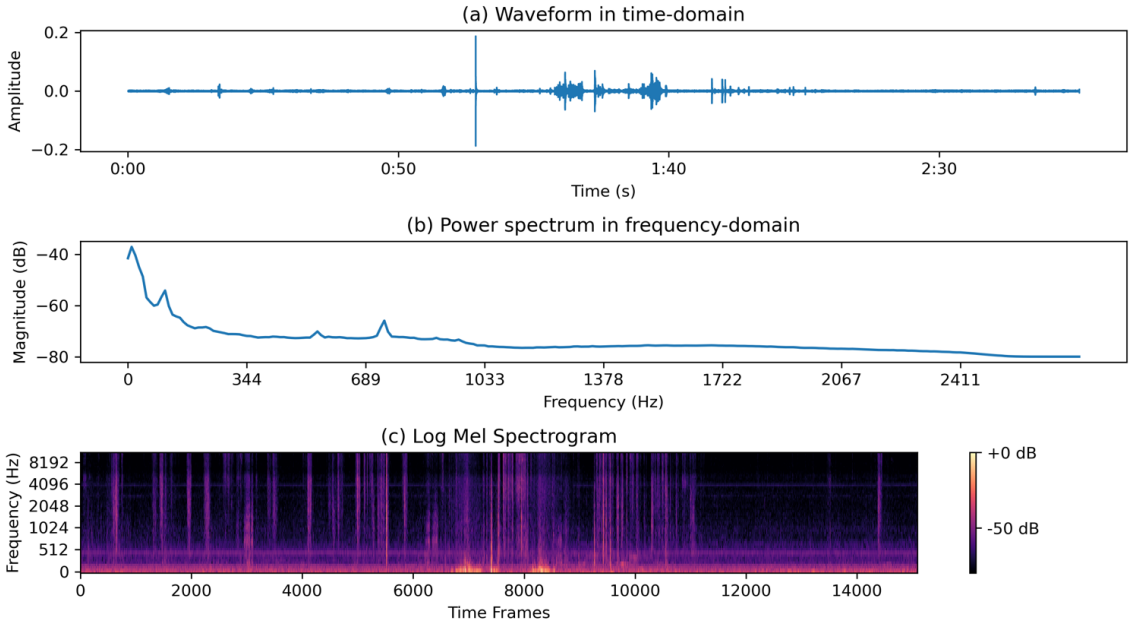

Figure 3.2 *A visualization of an audio data transformation describes the steps of the pre-processing in the Y-MCC method. A file (b029.wav) selected from the TUT-SED2016Home dataset shows (a) audio input data in a waveform representing a time-domain signal, (b) A frequency domain representation transformed from the input time-domain signal using an STFT converter, and (c) the final form of the input data transformed as a time-frequency domain with 2-dimensional shapes of the log-mel spectrogram.*

where the prediction $P(\hat{y}^p)$ with a matrix of a row and $n = 521$ columns contained probability value ranged from 0 to 1.

The next post-processing stage is designed to produce estimated output files containing the detected event labels with temporal information following the SED system output format suggested by [66]; audio file name, starting time, ending time, and event class label. To create the estimated file for each input audio recording, all YAMNet predictions are aggregated into a matrix shape of (Number of patches, Top5_indices, Top5_probability) as the total frames of a given audio recording. In this case, the desired number of top probability using top 5, thus TOP5 class indices have the highest probability among the 521 classes in the patch. It can be obtained using an argsort function defined by:

$$Top5 = argsort(P(\hat{y}^p)), \tag{3.3}$$

where the top five classes with the highest probability rank for a patch are obtained using the argsort function, and all other less significant classes are discarded.

3.3.3 MCC-Based YAMNet Class Mapping

Matthews correlation coefficient (MCC) was introduced in 1975 by biochemist Brian W. Matthews in studying lysozyme protein structure predictions [70]. Nowadays, the MCC is widely used in bioinformatics and also machine learning to evaluate the performance of classifier models [68, 71] using the four conditions of the confusion matrix shown in Figure 2.9. The computation of the MCC matrix requires all statistical information of TP, FP, TN, and FN, whereas the F1-score needs to consider the three variables except TN. Due to this reason, many researchers claimed that classifier evaluation based on the MCC is more reliable for imbalanced datasets and class sizes than the F_1 score [67, 68, 72, 73].

The MCC produces a value ranging $[-1, +1]$ in a correlation calculation between prediction and ground truth. It can be interpreted as a perfect prediction when the $MCC = +1$ compared to the ground truth. In contrast, there is a total inverse association with $MCC = -1$, and $MCC = 0$ refers to no correlation between the predictions and their references [68]. The MCC formula is defined as follows:

$$MCC(Pre, GT) = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}}, \quad (3.4)$$

where the MCC correlation probability score computation between the Pre (prediction) and the GT (ground truth) is computed based on the 4 conditions in the confusion matrix [72].

There are mainly three steps in creating and evaluating MCC-based class maps commonly used for the three selected benchmark datasets. Firstly, for the MCC matrix creation, two types of files that should be fed to compute for an MCC matrix computation are estimated and reference files corresponding to the training portion of the dataset. The estimated files produced in the Y-MCC method channel 1 and stage 2 using the YAMNet model with the original class map according to the parameter settings, such as for the Y-MCC method top 5 high-probability classes to be taken into a prediction vector with their time boundaries for writing them into an estimated file. In the next stage, the confusion matrix must be created by comparing the pairs of estimated predictions and ground truth in the reference files to produce output as the four conditions of the confusion matrix (TP, FP, TN, FN). After that, an MCC matrix can be created using the confusion matrix concatenated for computing Matthews correlation coefficient. The MCC matrix will have a dimension of $(521 \times numberof\_target\_classes)$ and coefficients value ranging in [-1, 1]. The MCC threshold for creating new class maps can be defined based on the maximum correlation scores of the MCC matrix for the target classes. Finally, the MCC-threshold-based class maps are created for the Y-MCC method

evaluation process in the second channel, it will be replaced the original YAMNet class map with the MCC-threshold-based class maps.

To elaborate on the relationship between the MCC matrix and the MCC-threshold-based class mapping, a sample of the MCC matrix computed based on the SINS dataset and its visualized bar plot are presented in Figure 3.3. The MCC matrix of the SINS dataset has a dimension of 521×9, which has 521 YAMNet classes in rows and 9 SINS target classes in columns. As seen from the top figure, a snapshot of the MCC matrix, presenting only the first 6 rows from the whole matrix, shows a trend of the MCC correlation score between YAMNet and the SINS classes. The overall trend described by the plot shows the MCC matrix score in the range of $MCC_{min,max} \in [-0.369, 0.883]$. However, most YAMNet classes have below 0.4 MCC scores, except for only one YAMNet class having the maximum MCC score of 0.883 for the SINS 'vacuum_cleaner' class. Generally, considering the majority of the maximum MCC score at the class-wise level, it can be decided to define the MCC threshold for class mapping. If the range of the MCC threshold is defined with [0. 0.5] with 0.05 steps, there will be 11 class mapping created for the Y-MCC method evaluation.

	absence	cooking	dishwashing	eating	other	social_activity	vacuum_cleaner	watching_tv	working
0	-0.176	-0.082	-0.042	-0.054	-0.051	0.208	-0.035	0.339	-0.175
1	-0.008	-0.004	-0.002	-0.002	-0.002	-0.004	-0.002	0.023	-0.008
2	-0.042	-0.02	-0.01	-0.013	-0.012	0.001	-0.008	0.111	-0.042
3	-0.089	-0.041	-0.021	-0.027	-0.026	0.031	-0.018	0.214	-0.088
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

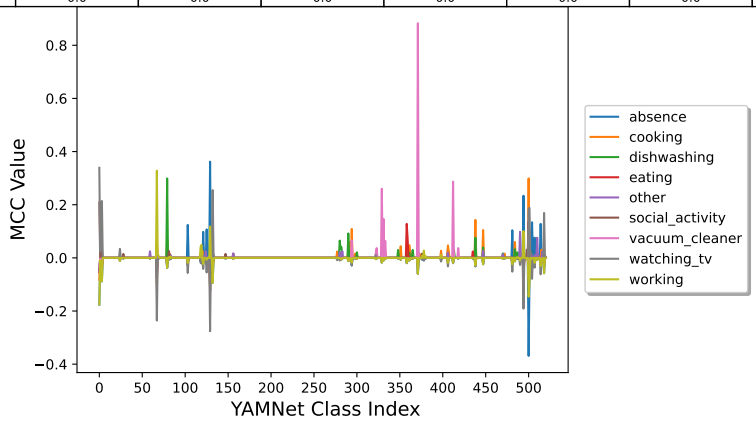

Figure 3.3 *A visualization of the MCC matrix extracted from the SINS dataset: (Top) A snapshot of the MCC matrix with the shape of [521, 9], (Bottom) A plot of the MCC scores obtained by the 521 YAMNet classes against the 9 SINS target classes.*

To examine more closely, the first row of the MCC table with the '0' index corresponding to the 'Speech' YAMNet class has the highest MCC value for 'watching_tv' in the SINS target classes with a 0.339 MCC score. With the rule of class mapping, only one target class is assigned to the YAMNet class. Therefore, the YAMNet 'Speech' can be activated in the MCC-threshold-based class maps from 0 to 0.3 for 7 class maps while deactivated in the higher MCC-threshold-based class maps. The

same rules apply to all other YAMNet classes. Hence one YAMNet class can be mapped to one in the target classes, whereas one target class can be mapped with multiple YAMNet classes if the MCC score is greater than the MCC threshold.

Table 3.4 and 3.5 show concrete examples of how the different class maps can be generated based on the MCC threshold with 0.5 and 3.0 for the SINS dataset. In the first table, 3 YAMNet class entries of 0, 2, and 3 have mapped to the SINS target class of 'watching_tv'. By contrast, other YAMNet classes have not been mapped and marked with '998' as an indication of unmapped classes presented in the 'T_ID' column. Based on the MCC matrix presented in Figure 3.3, the mapped 3 YAMNet classes have MCC scores higher than 0.05 with 0.339, 0.111, and 0.214, respectively. On the other hand, the later class map in Table 3.5 has been generated based on the MCC threshold of 0.3, which shows the two YAMNet entries of 2 and 3 indices have been deactivated. However, the first 'Speech' YAMNet class remains as activated and mapped to 'watching_tv' due to its MCC score being higher than the mapping MCC threshold of 0.3.

Table 3.4 *The table presents an example of the MCC-threshold-based class mapping for the SINS dataset with a 0.05 MCC threshold showing only the first 6 entries of the 521 YAMNet classes mapped to the SINS 9 target classes. The 'index' column represents the YAMNet class index and YAMNet class names in 'display_name', whereas the SINS class index in 'T_ID' and class name in 'T_Class'.*

index	mid	display_name	T_ID	T_Class
0	/m/09x0r	Speech	7	watching_tv
1	/m/0ytgt	Child speech, kid speaking	998	None
2	/m/01h8n0	Conversation	7	watching_tv
3	/m/02qldy	Narration, monologue	7	watching_tv
4	/m/0261r1	Babbling	998	None
5	/m/0brhx	Speech synthesizer	998	None

Table 3.5 *The table presents an example of the MCC-threshold-based class mapping for the SINS dataset with a 0.3 threshold showing only the first 6 entries of the 521 YAMNet classes mapped to the SINS 9 target classes.*

index	mid	display_name	T_ID	T_Class
0	/m/09x0r	Speech	7	watching_tv
1	/m/0ytgt	Child speech, kid speaking	998	None
2	/m/01h8n0	Conversation	998	None
3	/m/02qldy	Narration, monologue	998	None
4	/m/0261r1	Babbling	998	None
5	/m/0brhx	Speech synthesizer	998	None

The proposed Y-MCC method utilized an MCC library for Python code implementation found from Scikit-learn[1] [74]. The MCC computation in the Y-MCC proposed method is to obtain an MCC matrix for each given dataset by computing the MCC association score between the 521 YAMNet classes and the target classes defined in the given dataset. The MCC matrix creation and the MCC-threshold-based class mapping are highly dependent on the target dataset; therefore, it is further discussed in sub-chapters dedicated to the three selected benchmark datasets in Chapter 4 Performance Evaluation.

[1] https://scikit-learn.org/dev/modules/generated/sklearn.metrics.matthews_corrcoef.html?highlight=mcc

4 Performance Evaluation

This chapter presents and discusses the performance evaluation results regarding the second channel procedure of the Y-MCC methodology discussed in Chapter 3.2 and continued from Chapter 3.3 MCC-based Statistical Methodology for Class Mapping for each evaluation dataset. The performance was evaluated according to the SED evaluation metrics and tools suggested for the SED tasks of the DCASE Challenges [66], described in Chapter 2.3.3. And the official evaluation tool `sed_eval` from the SED evaluation toolbox [66] was used to measure the segment-based and class-wise performance explained in Chapter 2.3.3. For the Y-MCC method evaluation, three publicly available datasets were chosen by considering their significant contribution to the SED system research targeting the homecare environment. The details of the three selected datasets, TUT-SED2016, ESC-50, and SINS dataset, can be found in Chapter 2.3.2. Therefore, the following sub-chapters discuss the MCC-based class mapping and performance evaluation on the Y-MCC method for the three selected datasets.

4.1 Y-MCC Performance on TUT-SED2016 Home Dataset

As a first evaluation experiment, a subset of the TUT-SED2016 development dataset consisting of home event recordings [47], hereafter named TUT-SED2016Home, was considerably chosen to assess the Y-MCC method adaptability for the polyphonic SED task. The tasks were carried out from the MCC matrix creation for generating new class maps and performance evaluation using the MCC-threshold-based new class maps. However, the dataset has been claimed to be quite challenging for many SED systems because the dataset was recorded in a real-life environment containing ambient background noises and prominent sound events that often have very low power intensities [36]. Moreover, the dataset also provides ground truth information with temporal boundaries of starting and ending of the multiple events with respect to the SED system to be precisely designed for detecting single or multiple overlapping sound events with precise occurrence time. Despite the challenging facts, the TUT-SED2016Home dataset was tested for the first performance evaluation experiment for the Y-MCC method because the target sound events are highly important for understanding real-life indoor home event sounds. Moreover, it has a relatively small volume of audio recordings, potentially saving computing resources and time for evaluation.

4.1.1 MCC-based Class Mapping for TUT-SED2016 Home

The main two steps in MCC-based class mapping of the Y-MCC method consist of creating an MCC matrix and then generating class maps based on the MCC matrix. For the MCC matrix creation, two files as input are required to compute for an MCC matrix computation, estimated and reference files corresponding to the training portion of the TUT-SED2016Home dataset. The estimated files were produced in the Y-MCC Stage 2 by the YAMNet model according to the parameter settings, such as the number of classes taken from the YAMNet prediction, for the Y-MCC method top 5 high-probability classes were taken into a prediction vector with their time boundaries and written into an estimated file when all the frames of the input record were processed. In the next stage, when all estimated files were generated for all files in the training subset, the confusion matrix was created by comparing the pairs of estimated predictions and ground truth in the reference files to produce output as the four conditions of the confusion matrix (TP, FP, TN, FN), as discussed in the MCC method in Chapter 3.3.3. After that, the confusion matrix was concatenated for computing Matthews correlation coefficient to create an MCC matrix, with a dimension of 521×11, for the whole input dataset. Figure 4.1 presents the trend of the MCC matrix produced for the TUT-SED2016Home dataset. As can be seen, the general trend of the MCC score was widely spread, ranging in $MCC_{min,max} \in [-0.297, 0.525]$ over the 11 target classes of TUT-SED2016Home. The maximum correlation scores in class-wise, around half of all classes were obtained fairly well, with over 0.2, contrary to the other half, with below 0.2.

Looking more closely, Table 4.1 shows the MCC matrix in class-wise statistical descriptions. The highest MCC value was obtained by 'water tap running' at 0.525, followed by 'dishes' at 0.264 and 'object impact' at 0.244. On the other hand, the '(object) snapping' class showed very weak positive correlations with YAMNet classes at 0.056, seen from the maximum MCC value point of view. These class-wise highest and lowest maximum values indicated to determine the threshold range for the MCC matrix to be set for 11 stages with $MCC_{threshold} \in [0, 0.5]$ with 0.05 steps. As a result, 11 new class maps were created based on the MCC thresholds to map between 521 YAMNet classes and the 11 target classes predefined in the TUT-SED2016Home dataset.

The class correlations between the MCC score over 0.25 obtained by the TUT-SED2016Home and YAMNet classes are presented in Table 4.2, which was one of the 11 new class maps produced based on the 11 MCC threshold. It can be observed that the MCC-based class mapping method effectively resolved the uncertainty of mapping the various YAMNet classes to the target classes. For example, the 'water tap running' target class mapped with 8 YAMNet classes, including 'Sizzle,'

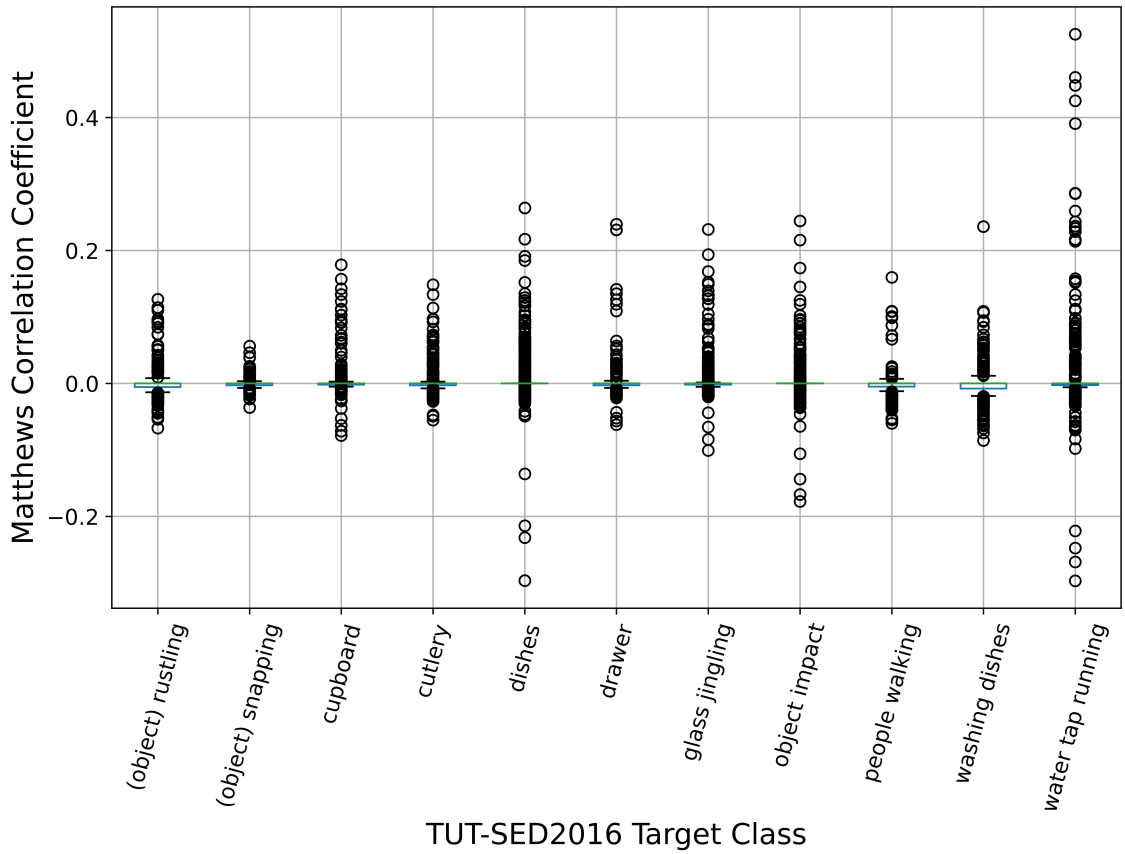

Figure 4.1 *The boxplot presents a trend of the correlation measured using Matthews correlation coefficient matrix between 521 YAMNet classes and 11 TUT-SED2016 target classes. The overall MCC minimum and maximum value ranged in* $MCC_{min,max} \in [-0.297, 0.525]$.

'Liquid,' 'Frying (food),' and 'Hiss,' with a significant level of positive correlation values. And the other 'dishes' target class mapped with the YAMNet's 'Dishes pots and pans' with a higher MCC score than the threshold of 0.25. In general, this mapping evidence could indicate the reliability of the MCC-based class mapping, which could be seen as improving the accuracy of finding closely correlated YAMNet classes among 521 for the 11 TUT-SED2016Home target classes.

4.1.2 Results

The previous studies of the SED system performance measured on the TUT-SED2016 dataset in the DCASE 2016 challenge have been reported as very challenging and insignificant [5, 47]. As discussed in Chapter 2.3.2 about the TUT-SED2016 dataset, the baseline system employed the GMM method with MFCC feature extraction performed low scores of F1 at 18.1% and ER at 0.95 in segment-based metrics [5, 47]. The ranked top one system obtained a significantly better error rate with 47.8% F1-score and 0.80 ER for the TUT-SED2016 dataset (31.0% F1 and 0.91 ER on the Home development dataset) which employed an RNN ar-

Table 4.1 *A table of the MCC matrix in class-wise statistical descriptions explains a correlation trend between the 521 YAMNet classes and the 11 target classes of the TUT-SED2016Home dataset. Average, minimum, maximum, and standard deviation abbreviations are denoted as Mean, Min, Max, and STD. The MCC strongest correlation was found from 'water tap running' with* 0.525 *in the TUT-SED2016Home target class.*

	MCC			
TUT-SED2016 Class	**Mean**	**Min**	**Max**	**STD**
(object) rustling	-0.0005	-0.067	0.126	0.019
(object) snapping	-0.0010	-0.036	0.056	0.007
cupboard	0.0026	-0.078	0.178	0.022
cutlery	0.0017	-0.055	0.149	0.018
dishes	0.0070	-0.296	**0.264**	0.038
drawer	0.0016	-0.062	0.239	0.024
glass jingling	0.0037	-0.101	0.232	0.027
object impact	0.0026	-0.177	**0.244**	0.029
people walking	-0.0024	-0.060	0.160	0.016
washing dishes	-0.0013	-0.086	**0.236**	0.022
water tap running	0.0093	-0.297	**0.525**	0.065

Table 4.2 *A MCC-based class map with 0.25 threshold contains YAMNet classes mapped to the target classes of the TUT-SED2016Home dataset. This example shows the reliability of the MCC-based class mapping to resolve the uncertainty of finding related classes among 521 YAMNet classes to 11 target classes.*

YAMNet Class	**TUT-SED2016 Class**	**MCC value**
Hiss	**water tap running**	**0.425**
Water	water tap running	0.269
Dishes pots and pans	**dishes**	**0.264**
Frying (food)	**water tap running**	**0.448**
Electric shaver electric razor	water tap running	0.286
Liquid	**water tap running**	**0.460**
Spray	water tap running	0.391
Stir	water tap running	0.285
Sizzle	**water tap running**	**0.525**

chitecture with Mel energy feature [5, 47, 75]. Additionally, it was reported that other systems in the same challenge task ranked from the top 2 to 9 also showed remarkably lower ER in [0.9, 0.97] (F1-score ranging in [23.9%, 42.9%]) than the top-ranked system [5]. As overall performance results of the Y-MCC system are given in Table 4.3, a statistical summary of the 1-second segment-based micro average over the MCC-threshold based 11 class maps using 5 audio record files of the evaluation dataset. In addition, Figure 4.2 illustrated the Y-MCC performance trend in the micro average metrics over 11 MCC-threshold-based class maps, which

indicated that the improving the performance over the threshold at the beginning steps. The results indicated that the lowest error rate performance was achieved with 0.88(±0.97) (corresponding F1 with 25.0%, at 0.25 MCC threshold) and the best F1-score at 34.5%(±9.5) (corresponding ER with 1.82, at 0.1 MCC threshold). In general, the Y-MCC performance results could be interpreted as compared to the previous studies in the DCASE2016 challenge, the Y-MCC method with the best F1-score at 34.5% achieved similar performance to the average F1-score at 35.2% of the top 9 systems measured on the TUT-SED2016Home dataset.

Table 4.3 *The table presents a statistical description of the 1-second segment-based overall micro average performance measured for the Y-MCC method using the TUT-SED2016Home dataset over the 11 MCC-threshold-based class maps.*

Statistics	Micro Average F1-score (%)	Precision (%)	Recall (%)	ER
Mean	22.6	44.1	28.2	1.43
Min	6.4	16.1	3.4	**0.88**
Max	**34.5**	**67.9**	**71.5**	3.72
STD	9.5	17.6	25.9	0.97

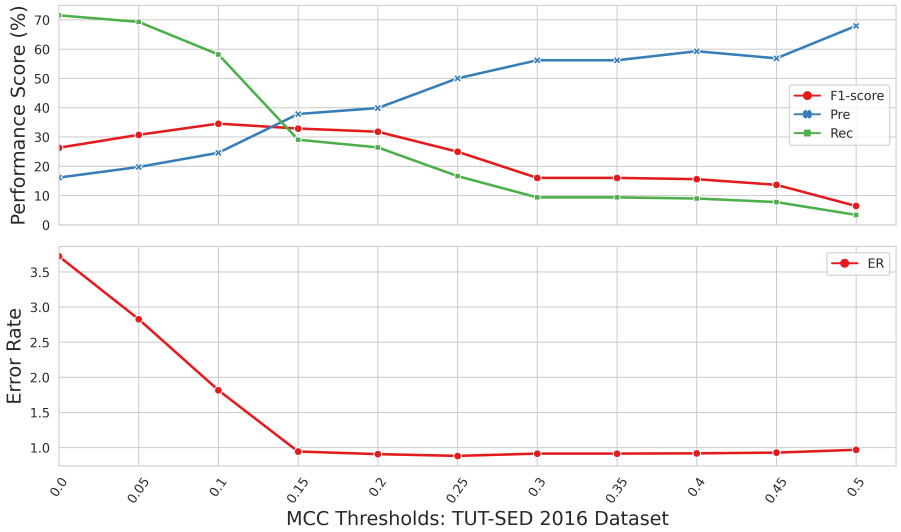

Figure 4.2 *The line plots show the Y-MCC overall micro average performance measured on the TUT-SED2016Home evaluation dataset over 11 steps of the MCC-threshold-based class maps. The general trend has shown that the performance improved over the MCC thresholds in terms of precision (TOP: blue line), with the highest F1-score of 34.5% at 0.1 MCC threshold (Top: red line), while the recall score steadily dropped (Top: green line). Additionally, a trend from the error rate (Bottom: red line) has shown remarkable drops and remained under 1 from the MCC threshold at 0.15 onward and the best ER of 0.88 at the 0.25 MCC threshold.*

More details of the Y-MCC class-wise performance are shown in Figure 4.3, and the maximum F1-score achieved by the 11 TUT-SED2016Home target classes are

listed in Table 4.4. The overall trend in the class-wise performance observed over the 11 MCC-threshold-based class maps that most of the target classes obtained some positive level of F1-score, which seemed difficult for an unbalanced dataset for four classes with the small number of sound events as reported in [5, 47]. The article described that in the baseline system, no true positive event detection was found for those small volume classes, including 'glass jingling' and '(object) snapping.' As expected, the MCC-threshold-based class maps helped to improve the Y-MCC performance for all the target classes even though the classes had small instances, which can be seen as a benefit of the pre-trained YAMNet with a large amount of AudioSet dataset. Furthermore, it can be seen in Figure 4.4 that most of the classes improved ER when the threshold increased, especially for '0: (object) rustling' dropped dramatically ER. On the other hand, with '10: water tap running' at the threshold zero, F1-score was highest at 58.5% F1 with 1.11 ER, afterward when thresholds increased, F1 slightly dropped but improved the precision and error rate resulted in the lowest error rate at 0.84 with precision at 61% and F1 at 50.9%. Moreover, other classes like '4: dishes' and '9: washing dishes' achieved over 40% F1-score, which can also be seen as a very significant score, considering the serious difficulty of the polyphonic with strongly labeled SED task using the unbalanced TUT-SED2016Home dataset. To summarize, the Y-MCC method successfully improved the class-wise performance scores for most TUT-SED2016Home target classes.

Table 4.4 *Y-MCC class-wise performance on the TUT-SED2016Home dataset presents the best F1-score obtained by the 11 target classes from 11 MCC-threshold-based class maps indicating the best F1-score by the '10: water tap running' with 58.5% at 0 MCC threshold.*

TUT-SED2016 Index_Class	Th	F1-score (%)	Precision (%)	Recall (%)	ER
0_(object) rustling	0.10	19.9	11.2	89.3	7.20
1_(object) snapping	0.05	6.7	11.1	4.8	1.33
2_cupboard	0.10	15.6	13.2	19.2	2.08
3_cutlery	0.05	8.5	5.2	22.9	4.94
4_dishes	**0.15**	**42.5**	**34.2**	**56.2**	**1.52**
5_drawer	0.00	16.5	16.1	17.0	1.72
6_glass jingling	0.20	38.9	33.3	46.7	1.47
7_object impact	0.05	37.0	26.8	59.9	2.04
8_people walking	0.10	23.3	13.4	90.6	5.97
9_washing dishes	**0.10**	**41.3**	**30.5**	**63.9**	**1.82**
10_water tap running	**0.00**	**58.5**	**46.7**	**78.2**	**1.11**

Consequently, the results suggested that the statistical method of Matthews correlation coefficient for the Y-MCC method improved the uncertainty of determining YAMNet classes toward 11 target classes predefined in the TUT-SED2016Home dataset. Based on the MCC-based class mapping approach, the 1-second segment-based micro average performance with 34.5% F1-score can be seen as a similar score

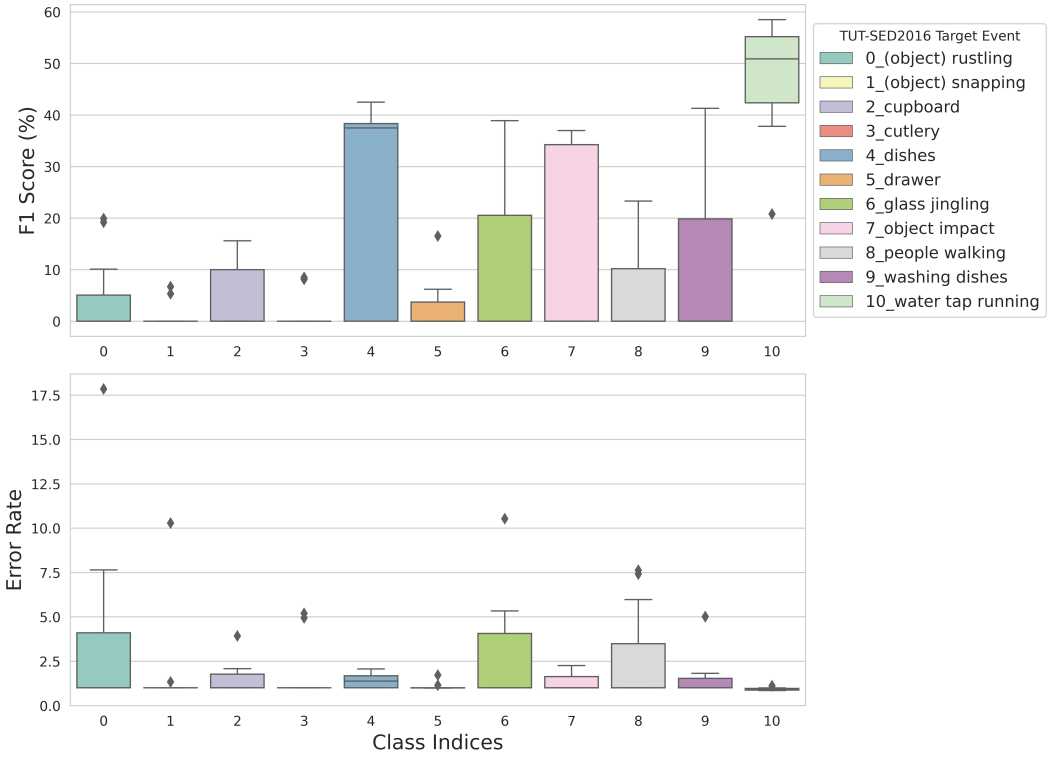

Figure 4.3 *The box plots present a trend of the class-wise performance metrics, Top: F1-score and Bottom: ER, of the Y-MCC method measured on the TUT-SED2016 dataset over 11 MCC-threshold-based class maps. Overall, it could be observed that all the target classes achieved some positive level of F1-score, even though the dataset was considered highly challenging for many known reasons.*

to an average top 9 performers of the DCASE 2016 challenge for the SED task. Obviously, the error rate seems higher for a SED system, but the classes like '10: water tap running' and '5: drawer' improved the error rate below 1 when the MCC threshold increased. This indicated that The Y-MCC performance in class-wise showed surprisingly good results considering the many challenging constraints discussed regarding the polyphonic SED task using the TUT-SED2016Home dataset. Furthermore, the correlation measurement based on Pearson showed a very strong relationship at 0.85 between the maximum F1 score and the Maximum MCC score in the 11 target classes. In conclusion, the observations of the Y-MCC performance results based on the statistical MCC methodology to map classes between YAMNet and TUT-SED2016 target classes could be moderately reliable for applying the system to the polyphonic SED task, particularly monitoring the home environment for the water tap running sound event.

4.2 Y-MCC Performance on ESC-50 Dataset

ESC-50 or ESC50 dataset is the class-wise well-balanced dataset with 50 target classes [60], described in Chapter 2.3.2. The dataset is used mainly for classification

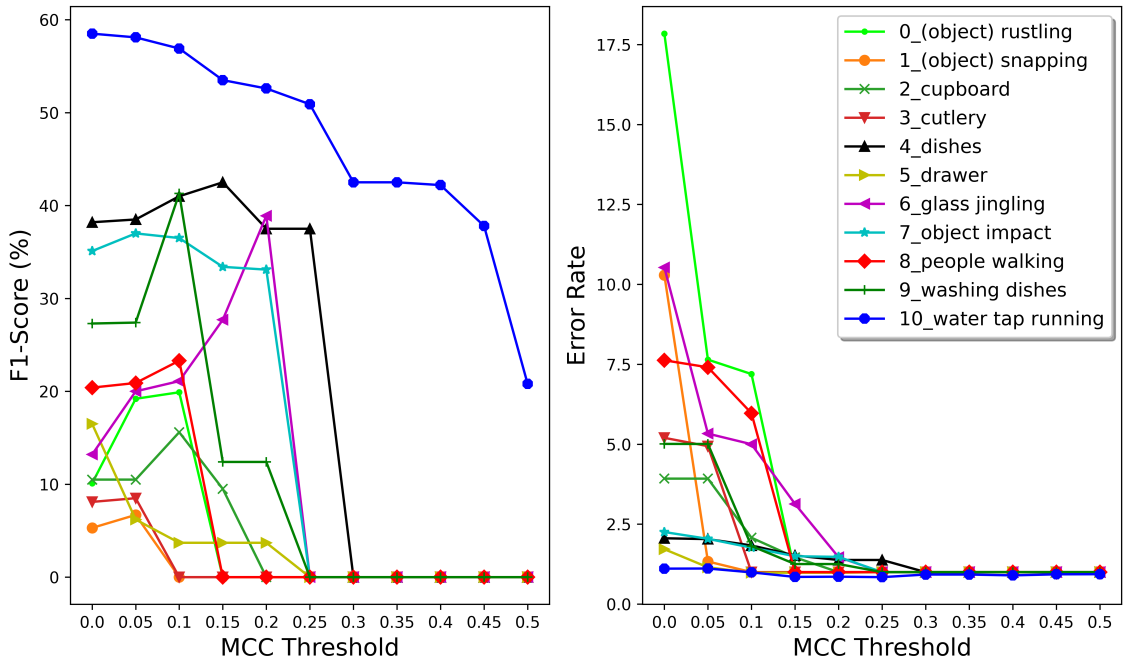

Figure 4.4 *The line plots present a trend of the class-wise performance metrics, Left: F1-score and Right: ER, of the Y-MCC method measured on the TUT-SED2016Home dataset over 11 MCC-threshold-based class maps. Overall, it could be observed that the Y-MCC method successfully improved most target class performance.*

tasks. However, it seems beneficial to evaluate the Y-MCC method because of the high relevance of the many target classes in homecare monitoring, especially sound groups of non-human speech sound and domestic urban sound highly relevant.

Using the ESC-50 dataset, the procedures to generate the MCC matrix between 521 YAMNet classes and 50 ESC-50 target sound event classes were done similarly to the TUT-SED2016Home dataset. The three main differences with the ESC-50 dataset were the fold-based audio file structure, a 5-second short-fixed size of audio clips, and monophonic reference. Therefore, the MCC matrix procedure and evaluation for the fold-based dataset have been implemented to handle 5-fold data subsets. Three out of five folds were used for generating the MCC matrix in 5-second 1200 audio clips. And the other two folds, fold 4 and 5, in a total of 5-second 800 audio clips, were used for the YMCC method evaluation.

4.2.1 MCC-based Class Mapping for ESC-50

Figure 4.5 illustrates the MCC correlation distribution dimension of 521×50 computed between 50 ESC-50 target classes and 521 YAMNet classes. As a result, The MCC correlation values spread in the range of $MCC_{min,max} \in [-0.068, 0.704]$ and over 24 ESC50 classes obtained over 0.4 MCC correlation values, which showed a considerably good correlation.

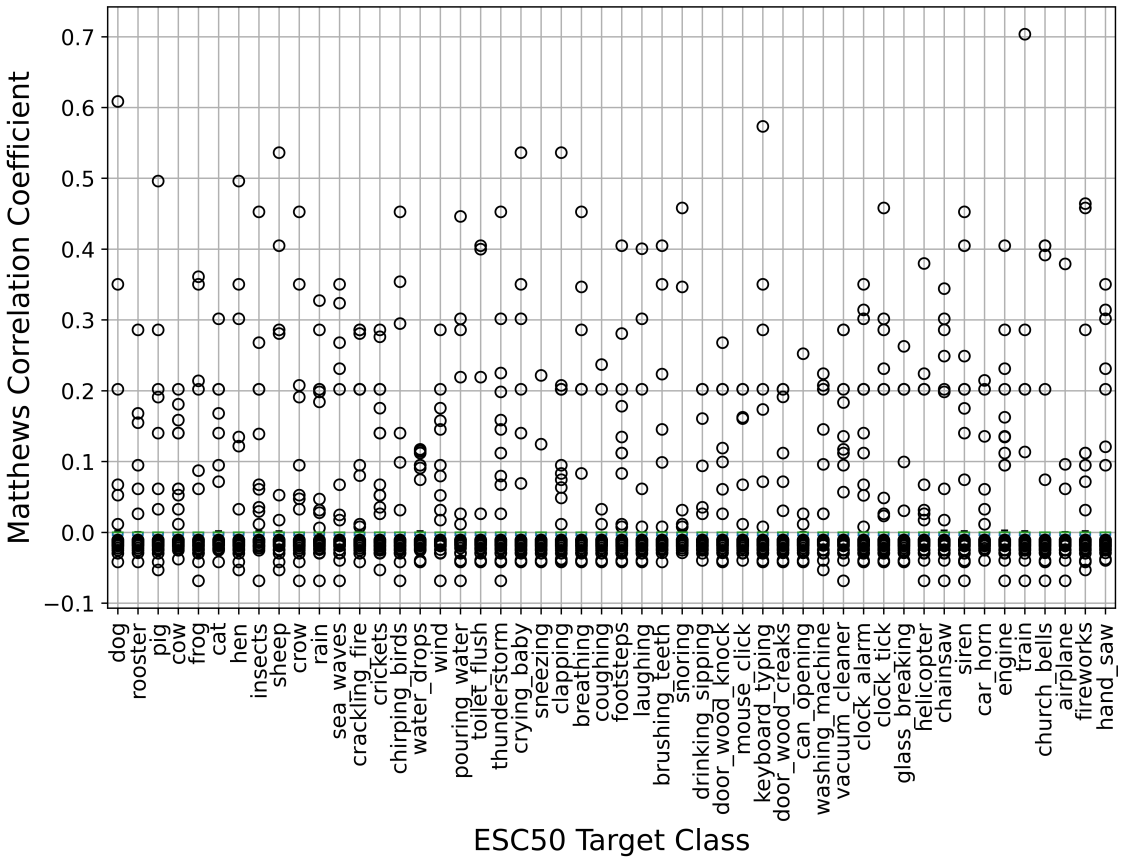

Figure 4.5 *The boxplot presents a trend of the MCC matrix between 521YAMNet classes and 50 ESC-50 target classes computed based on three folds of 1200 audio clips that were well-balanced between the classes. Among the ESC-50 50 target classes, 24 obtained an MCC value higher than 0.4.*

A more closely examined statistical description of the MCC matrix is summarised in Table 4.5 regarding the top 10 classes seen from the MCC maximum value. The highest three classes were obtained by 'train' at 0.704(±0.036) followed by 'dog' at 0.608(±0.033) and 'keyboard_typing' at 0.573(±0.035). Moreover, among the other classes, 'crying_baby,' 'clapping,' and 'snoring' yielded good MCC scores considered to be useful for homecare sound event detection. Conversely, the lowest MCC maximum value obtained by 10 classes is shown in Table 4.6. Among them, 'water_drops' was the lowest with MCC_{max}0.117(±0.013), followed by 'cow' at 0.202(±0.018), and 'drinking_sipping' at 0.202(±0.016).

The statistical analysis of the MCC matrix suggested that it could determine the MCC-based threshold for ESC-50 class mapping to be in the range of [0, 0.7] with 0.05 steps, considering the 'train' MCC maximum score at 0.704. Therefore the 15 MCC thresholds were used to create 15 MCC-threshold-based class maps between YAMNet 521 and ESC-50 50 classes. For instance, one of the MCC-threshold-based class maps based on the MCC threshold at 0.5 is presented in Table 4.7. It can

Table 4.5 *A list of the Top 10 ESC-50 classes measured from the statistical descriptions of the MCC maximum scores.*

ESC-50 Class	MCC Mean	MCC Min	MCC Max	MCC STD
train	**0.0003**	**-0.068**	**0.704**	**0.036**
dog	**0.0001**	**-0.042**	**0.608**	**0.033**
keyboard_typing	**0.0007**	**-0.042**	**0.573**	**0.035**
sheep	0.0005	-0.053	0.536	0.035
crying_baby	0.0005	-0.042	0.536	0.033
clapping	0.0013	-0.042	0.536	0.032
pig	0.0002	-0.053	0.496	0.029
hen	0.0002	-0.053	0.496	0.031
fireworks	0.0006	-0.053	0.464	0.033
snoring	**-0.0007**	**-0.029**	**0.458**	**0.026**

Table 4.6 *A list of the Low 10 ESC-50 classes measured from the statistical descriptions of the MCC maximum scores.*

ESC-50 Class	MCC Mean	MCC Min	MCC Max	MCC STD
water_drops	**-0.0011**	**-0.042**	**0.117**	**0.013**
cow	**-0.0003**	**-0.038**	**0.202**	**0.018**
drinking_sipping	**-0.0011**	**-0.040**	**0.202**	**0.016**
mouse_click	-0.0009	-0.040	0.202	0.017
door_wood_creaks	-0.0010	-0.042	0.202	0.017
car_horn	-0.0011	-0.040	0.214	0.015
sneezing	-0.0019	-0.042	0.222	0.012
washing_machine	-0.0007	-0.053	0.224	0.019
coughing	-0.0016	-0.042	0.237	0.015
can_opening	-0.0019	-0.042	0.252	0.012

be observed that the ESC-50 classes were reasonably associated with similar class labels of the YAMNet. Nevertheless, the new 15 class maps replaced the original YAMNet class map to evaluate the Y-MCC method for the ESC-50 dataset.

4.2.2 Results

As discussed previously, since the ESC-50 dataset released to the public with baseline system performance, it has been improved the classification accuracy dramatically with various machine learning models. Initially, the baseline performance based on a simple CNN model was reported at a 64.5% accuracy rate in 2015 [63]. Especially recent advanced models have reached over 90% accuracy; for instance, the performance of a model named with sequential learning on the ESC-50 dataset

Table 4.7 *A MCC-threshold-based class map created based on the threshold at 0.5 shows associated classes between the YAMNet and ESC-50.*

YAMNet Class	ESC-50 Class	MCC value
Whimper	**crying_baby**	**0.536**
Hands	**clapping**	**0.536**
Bow-wow	**dog**	**0.608**
Goat	**sheep**	**0.536**
Railroad car, train wagon	**train**	**0.704**
Computer keyboard	**keyboard_typing**	**0.574**

achieved a remarkable accuracy score at 94.1% [64]. The Y-MCC performance metrics with the ESC-50 dataset in a 5-second segment-based micro average are statistically described in Table 4.8 and Figure 4.6. It indicates the best F1 score in a micro average at 41.2% (0.62 ER, 0.4 MCC threshold) and the lowest ER at 0.6 (40.9% F1, 0.35 MCC threshold). The corresponding balanced accuracy score for 41.17% F1-score yielded 68.47%. Although these results seemed considerably lower than the recent state-of-the-art, the Y-MCC method achieved slightly higher than the CNN-based baseline model with a 4% improvement in terms of accuracy. The MCC-threshold trend seen from the figure shows, in most cases, the performance regarding precision increased when the MCC thresholds increased, whereas F1-score and recall increased up to 0.35 threshold. Nevertheless, it can be observed that the Y-MCC method with the lowest error rate of 0.6 at the 0.35 threshold seems to perform reasonably well. Consequently, the MCC-threshold-based class mapping used in the Y-MCC model against the ESC-50 dataset performed reasonably well at micro-average. Still, it showed room for improvement compared to the recent advanced classification models.

Table 4.8 *Table shows a statistical description of the Y-MCC performance measured against the ESC-50 dataset in the 5-second segment-based micro-average across 15 MCC-threshold-based class maps. The best performance was achieved in F-score at 41.2% (MCC-threshold at 0.4)and ER at 0.6 (MCC-threshold at 0.35).*

Statistics	F1-score (%)	Precision (%)	Recall (%)	ER
Mean	27.1	39.8	25.2	0.75
Min	3.8	32.4	2.0	**0.60**
Max	**41.2**	**52.6**	**40.4**	0.98
STD	14.3	5.9	15.2	0.15

Looking closely at the class-wise performance metrics of the Y-MCC method is illustrated in Figure 4.7, and the top-10 F1-score achieved by the ESC-50 target classes are listed in Table 4.9. The overall trend in the figure, computed over the 15 thresh-

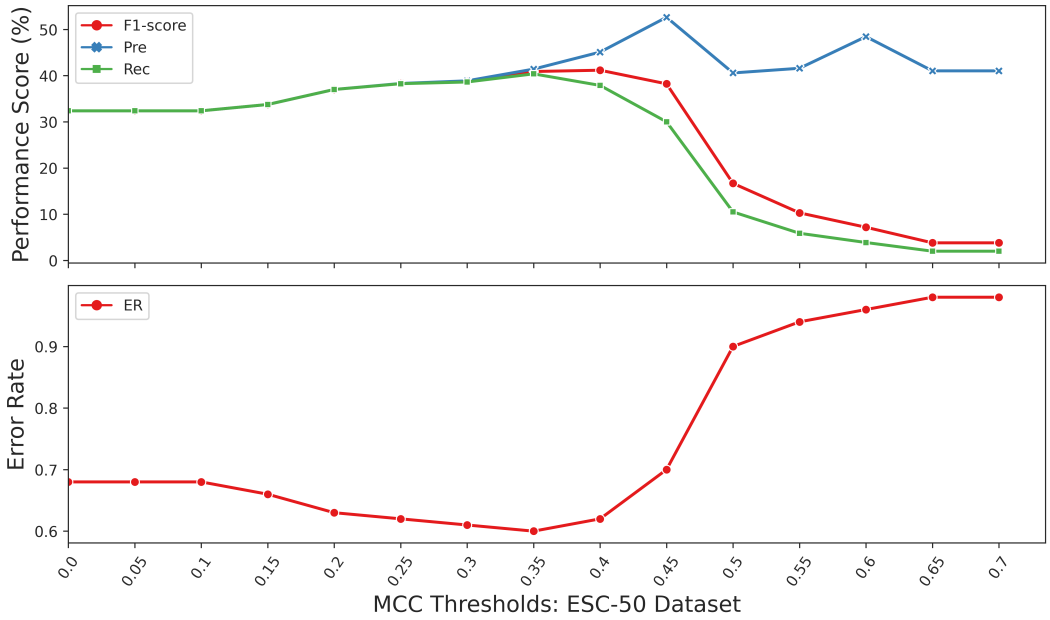

Figure 4.6 *The line plots present the growing trend of the Y-MCC method performance measured in the 5-second segment-based micro average on the ESC-50 dataset. The evaluation was performed over the 15 MCC-threshold-based class maps ranging [0, 0.7] with 0.05 steps. The performance metrics are shown on the top plot regarding F1, precision, and recall steadily increased from zero to the threshold of 0.4, reaching the F1-score peak at 41.2%. At the same time, the error rate (Down) declined from zero to 0.35 threshold, reaching its lowest point at 0.6.*

olds output, shows that 43 classes out of 50 obtained an F1 score higher than zero. Among them, the highest F1 score at 100% (0 ER) was achieved by '8_sheep' as a perfect score, followed by '27_brushing_teeth' and '16_wind' at 96.8% F1 and 0.06 ER for both classes. On the contrary, 7 ESC-50 target classes encountered problems producing meaningful F1 scores, such as '21_sneezing,' '33_door_wood_creaks,' and '48_fireworks' yielded zero F1-score. Despite the insignificant performance for approximately 14% of the total classes, the Y-MCC method achieved relatively high performance for most target classes including many important domestic sound event classes. Therefore, the class-wise performance results convinced the Y-MCC method successfully worked for the ESC-50 dataset in most of the 50 target classes.

A statistical method employing Matthews correlation coefficient for the Y-MCC noticeably reduced the uncertainty of determining YAMNet classes toward 50 target classes defined in the ESC-50 dataset. Based on the approach, the 5-second segment-based micro average performance with 64.47% accuracy (41.2% F1-score) was almost similar to the CNN baseline system at 64.5% of the dataset. Specifically, the top 10 F1-score classes obtained over 80% F1-score, including the 'sheep' target event with a 100% perfect detection score. Moreover, homecare-relevant sound classes like 'brushing_teeth,' 'glass_breaking,' 'toilet_flush,' and 'coughing' have

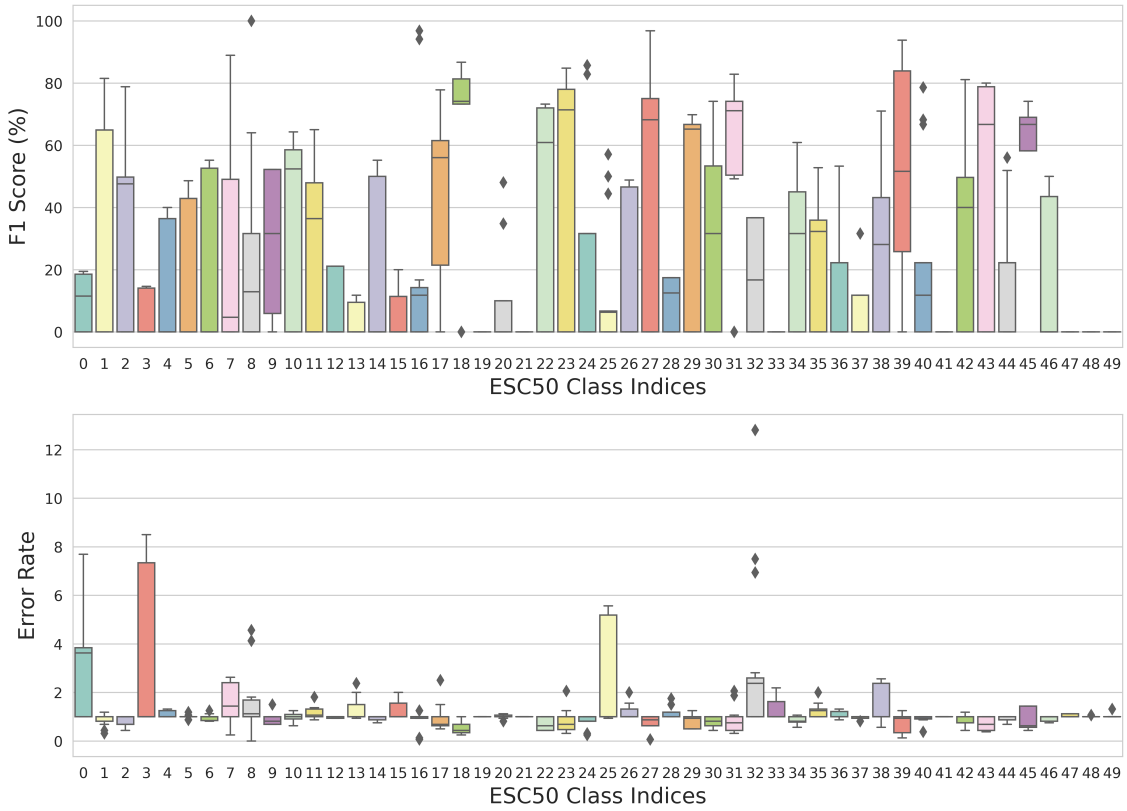

Figure 4.7 *The box plots present the Y-MCC method's positive trend of the class-wise performance metrics in F1-score (Top) and ER (Bottom) measured on the ESC-50 dataset over 15 MCC-threshold-based class maps. Overall, it can be observed that many classes achieved excellent performance, such as '8_sheep' achieved the perfect detection F1-score of 100% and zero ER, followed by '27_brushing_teeth' and '16_wind' at 96.8% F1-score and 0.06 ER.*

excellent performance over 85% F1-score. The top 10 classes showed a strong Person correlation score of 0.43, measured between the F1 and MCC scores. Although, some classes reported poor performance, which indicates room to improve the method. However, the observation of the class-wise performance results has shown convincing evidence that the Y-MCC method works effectively on the ESC-50 dataset even for the diverse sound sources, including human non-speech, domestic, and animal. These are essential sound events for homecare applications, such as 'coughing,' 'toilet flush,' and 'glass breaking.' Therefore, the performance results suggest that the Y-MCC method could be applied for homecare SED applications for those classes that achieved over 90% F1-score or 100% precision and recall, assuming that the sound characteristics of the application environment should be similar to the ESC-50 audio clips. Alternatively, as the YAMNet model utilized in [9] to detect segments with cough sound events for COVID-19 symptomatic detection, the Y-MCC method confirmed that the highly performed classes, especially classes achieved 100% precision like 'coughing' and 'sheep' could be used for the sound event detection in

Table 4.9 *A list summarizes the top 10 best class-wise performance metrics measured over the 15 threshold-based class maps for the ESC-50 dataset. Their F1 scores ranged from 83.9% to 100%, the perfect score achieved by '8: sheep' (0.4 MCC threshold) followed by '27: brushing_teeth' (0.35 MCC threshold) and '16: wind' (0.4 MCC threshold) with 96.8% F1-score and 0.06 ER for both classes. Moreover, the target class '7: insects' achieved 100% recall, and '24: coughing' performed 100% precision.*

ESC-50 Index_Class	Th	F1-score (%)	Precision (%)	Recall (%)	ER
8_sheep	**0.40**	**100.0**	**100.0**	**100.0**	**0.00**
27_brushing_teeth	**0.35**	**96.8**	**100.0**	**93.8**	**0.06**
16_wind	**0.40**	**96.8**	**100.0**	**93.8**	**0.06**
39_glass_breaking	0.45	93.8	93.8	93.8	0.13
7_insects	0.40	88.9	80.0	**100.0**	0.25
18_toilet_flush	0.35	86.7	92.9	81.2	0.25
39_glass_breaking	0.40	86.7	92.9	81.2	0.25
24_coughing	0.30	85.7	**100.0**	75.0	0.25
23_breathing	0.15	84.8	82.4	87.5	0.31
18_toilet_flush	0.40	83.9	86.7	81.2	0.31

segment level to assist the upstream model.

4.3 Y-MCC Performance on SINS Dataset

This chapter presents the Y-MCC performance evaluation using the SINS dataset, introduced in Chapter 2.3.2. The SINS dataset [39] plays a vital role in justifying the SED system's effectiveness for monitoring the homecare environment. The dataset with over 88 GB of audio recordings is divided into 4-fold subsets to facilitate cross-validation for generalizing machine learning methods. For the Y-MCC method evaluation, the MCC matrix was created using the fold1 subset of the SINS development dataset, which consists of the train set with 10-second 54964 audio recordings and the evaluation set with 18020 audio clips used for the performance evaluation of the Y-MCC method. This chapter mainly discusses the MCC-threshold-based class map generation between YAMNet and SINS target classes and assesses the Y-MCC performance measured on the SINS dataset.

4.3.1 MCC-based Class Mapping for SINS

In the first channel of the Y-MCC method, the MCC matrix was generated to create new class maps between 521 YAMNet and 9 SINS target classes. The MCC matrix using the SINS dataset was computed based on the confusion matrix comparing pairs of files, one from the ground truth of the fold1 train reference files and the other corresponding YAMNet's estimated files. Figure 4.8 illustrates the overall trend of the MCC matrix score in the range of $MCC_{min,max} \in [-0.369, 0.883]$ with a dimension of 521×9. Seen from the maximum MCC scores of the SINS target classes

presented in Table 4.10, it can be observed as most of the SINS classes obtained mild correlations (over 0.2 and below 0.4) with the YAMNet classes. The target class-wise, the highest correlation was shown by 'vacuum cleaner' at 0.883($\pm$0.043), whereas 'other' showed the lowest correlation at 0.098($\pm$0.008). This indicated to determine the MCC threshold for class map creation to be made in the range of $MCC_{threshold} \in [0, 0.5]$ with steps of 0.05. Therefore, 11 class maps were created based on the MCC threshold scores between YAMNet and SINS target classes and used in the second channel to measure the Y-MCC system performance.

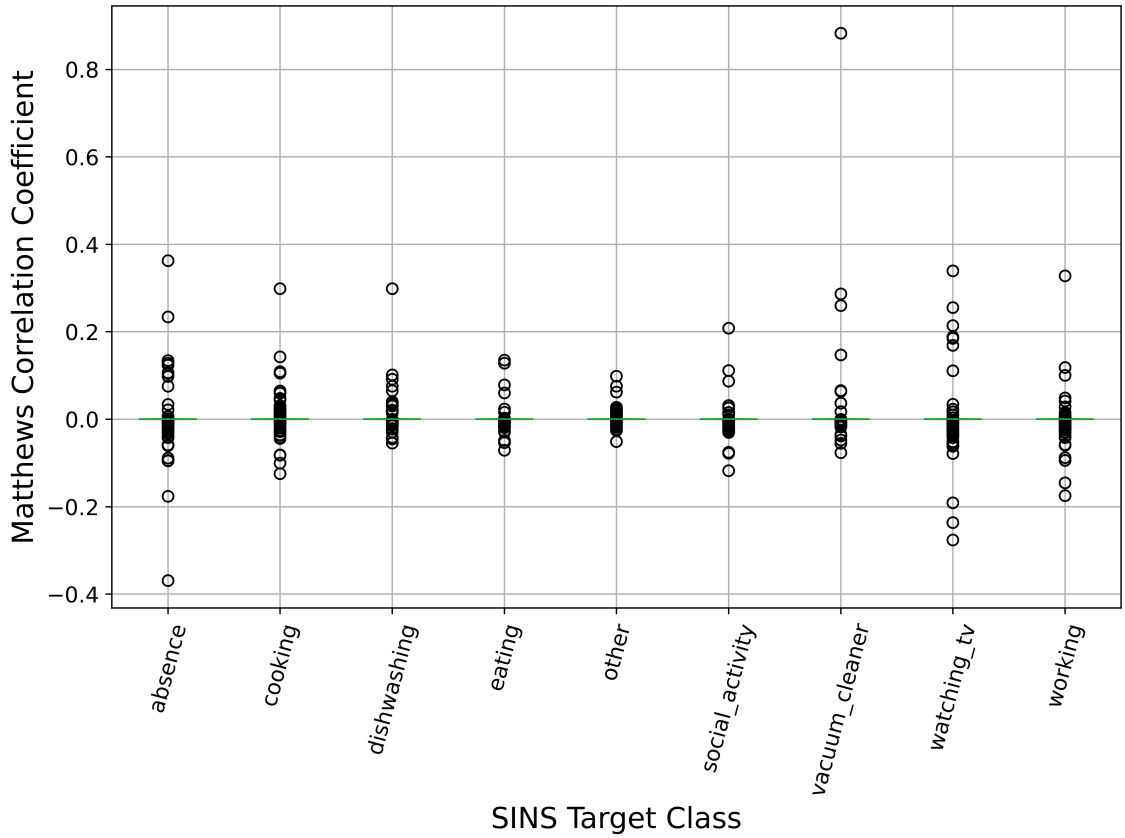

Figure 4.8 *The boxplot illustrates the general trend of the MCC matrix showing the correlation strength in the range of [-0.369, 0.883] between 521 YAMNet classes and 9 SINS target classes. Overall, as seen from the class-wise maximum MCC score, most of the target classes seemed to have a mild level of correlation under 0.4, except for the 'vacuum cleaner' at 0.883 as the highest.*

A closer look at a class mapping is presented in Table 4.11, which shows the MCC threshold at 0.25 produced 9 YAMNet classes mapped to 6 SINS target classes. It can be observed that there were multiple classes from YAMNet mapped to a SINS class. For example, the SINS 'vacuum_cleaner' were mapped with three YAMNet classes: 'Vacuum cleaner' at the highest MCC score of 0.883, 'Aircraft' at 0.26, and 'Tools' at 0.287. As the second highest MCC correlation, SINS 'absence' mapped

Table 4.10 *A statistical summary of the MCC metrics presents a degree of correlation between YAMNet 521 and SINS 9 target classes. The class-wise maximum MCC scores were used to determine the MCC threshold for MCC-based class map creation.*

	MCC			
SINS Class	**Mean**	**Min**	**Max**	**STD**
absence	-0.0001	-0.369	0.362	0.031
cooking	0.0010	-0.125	0.299	0.020
dishwashing	0.0011	-0.055	0.299	0.016
eating	-0.0002	-0.071	0.135	0.011
other	0.0004	-0.051	**0.098**	0.008
social_activity	-0.0004	-0.118	0.208	0.014
vacuum_cleaner	0.0025	-0.076	**0.883**	0.043
watching_tv	-0.0003	-0.276	0.339	0.032
working	-0.0011	-0.175	0.328	0.021

YAMNet 'absence' at 0.362. SINS 'watching_tv,' also multiply mapped with YAMNet 'Speech' at 0.339 and 'Music' slightly over the threshold at 0.255. Inevitably, the MCC-threshold-based class maps solved the uncertainty of the mapping problem between the YAMNet and SINS classes. Otherwise, it could be very difficult to estimate, for example, YAMNet 'Inside, small room' to map with SINS 'cooking' and YAMNet 'Snake' to SINS 'absence.' As a result, the MCC matrix and its

Table 4.11 *The table presents one of the class map examples based on the MCC threshold with 0.25. It was observed that 9 YAMNet classes were mapped to 6 SINS target classes. The strongest correlation was found from YAMNet 'Vacuum cleaner' mapped to SINS 'vacuum_cleaner' with 0.883. The second highest map was YAMNet 'Snake' to SINS 'absence' followed by YAMNet 'Speech' mapped to SINS 'watching_tv.'*

YAMNet Class	**SINS Class**	**MCC value**
Speech	**watching_tv**	**0.339**
Animal	working	0.328
Hiss	dishwashing	0.298
Snake	**absence**	**0.362**
Music	watching_tv	0.255
Aircraft	vacuum_cleaner	0.260
Vacuum cleaner	**vacuum_cleaner**	**0.883**
Tools	vacuum_cleaner	0.287
Inside, small room	cooking	0.299

statistical analysis provided information to determine the MCC-based threshold for class mapping to be made in the range of [0, 0.5] with 0.05 steps. Although 'vacuum_cleaner' was higher than the maximum threshold of 0.5 and mapped with only one YAMNet 'Vacuum_cleaner,' which indicated no need to extend the maximum threshold. Therefore the 11 MCC-based thresholds were used to create 11 class

maps to evaluate the Y-MCC performance in the next second channel. In general, the MCC statistical method suggested a way of solving the uncertainty of the class mapping and improving the reliability of the class mapping between the YAMNet and SINS target classes.

4.3.2 Results

In the DCASE 2018 challenge Task 5, the performance of the SED systems to classy target acoustic events defined in the SINS dataset has shown excellent class-wise F1-score performance [38]. The top-ranked 5 systems' performance in the challenge achieved class-wise macro F1-score averaged at 89.08%(±0.35), which exceeded nearly 5% higher than the baseline system with approximately 84% (refer to Chapter 2.3.2). However, the overall performance of the Y-MCC system was far lower than the baseline performance; the overall macro-average F1-score was 41.45%, a bit less than half of the baseline F1-score. A bar chart in Figure 4.9 illustrates the three class-wise F1-score comparisons on the SINS development dataset obtained by the Y-MCC method, the baseline system, and the average F1-score of the top 5 systems of the DCASE 2018 challenge.

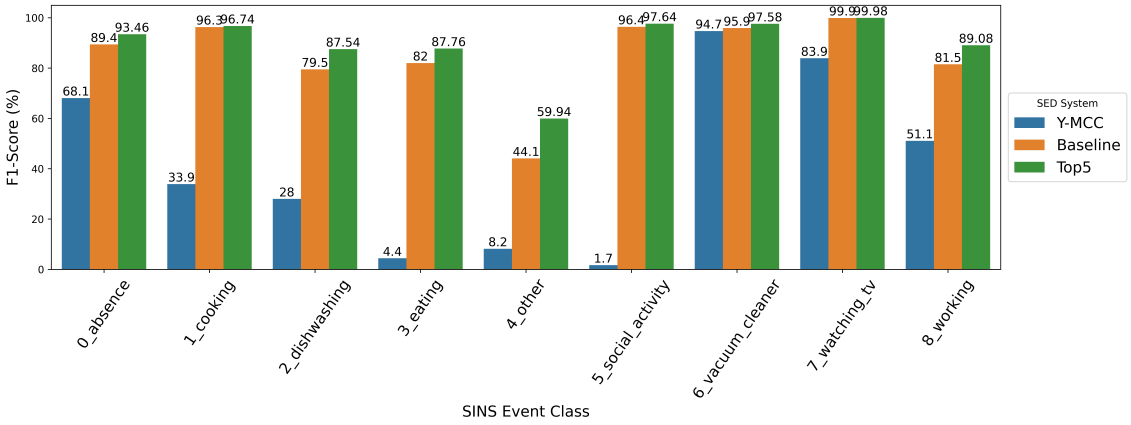

Figure 4.9 *The bar chart illustrates the class-wise performance of the Y-MCC in comparison with the baseline and top 5 systems (averaged) in DCASE2018 Challenge for the acoustic classification task using SINS development dataset. Overall the Y-MCC system performed far below the competing systems, but with '6: vacuum_cleaner', the Y-MCC system showed a significantly high F1-score at 94.7%.*

A closer look at the class-wise performance results shown in Table 4.12 provides important indications that the '6: vacuum_cleaner' class in the Y-MCC system could nearly beat the other competing systems with approximately 95% F1-score, 100% recall, and the lowest ER at 0.113. In contrast, the rest of the SINS target classes clearly indicated huge performance gaps. Additionally, three more classes of the Y-MCC system obtained over 50% F1-score, such as '7: watching_tv' at 83.9%, '0: absence' at 68.1%, and '8: working' at 51.1%; however, the competing

systems achieved higher than 80% F1-score. Similarly, '4: other' was the lowest performance obtained by the other competing systems, closer to 50%, 8.2% for the Y-MCC system as the second lowest on its own. To summarize, the findings of the '6: vacuum_cleaner' and '7: watching _tv' classes from the Y-MCC system measured on the SINS development dataset could be reliable for homecare applications with relatively good scores, whereas most of the other classes might not suggest using directly due to considerably low performance.

Table 4.12 *Table presents the class-wise performance metrics of the Y-MCC system compared to the TOP-ranked 5 and the baseline system published in the DCASE 2018 challenge for Task 5. Considering the F1-score and ER obtained by the Y-MCC system, '6: vacuum_cleaner' and '7: watching_tv' achieved significantly high F1-score over 80% and less than 0.4 ER, which could be considered highly reliable detection by the Y-MCC system.*

		Y-MCC				Baseline	Top5
SINS Index_Class	**Th.**	**F1(%)**	**Pre(%)**	**Rec(%)**	**ER**	**F1(%) [38]**	**F1(%) [38]**
0_absence	0.10	68.1	56.5	85.5	0.80	89.4	93.46
1_cooking	0.05	33.9	20.9	90.2	3.52	96.3	96.74
2_dishwashing	0.15	28.0	56.9	18.5	0.96	79.5	87.54
3_eating	0.05	4.4	68.4	2.3	0.99	82.0	87.76
4_other	0.00	8.2	28.1	4.8	1.08	44.1	59.94
5_social_activity	0.00	1.7	69.2	0.9	1.00	96.4	97.64
6_vacuum_cleaner	**0.30**	**94.7**	**89.9**	**100.0**	**0.11**	**95.9**	**97.58**
7_watching_tv	**0.30**	**83.9**	**75.0**	**95.3**	**0.37**	**99.9**	**99.98**
8_working	0.30	51.1	47.8	54.9	1.05	81.5	89.08

In addition, the overall micro average performance results summarized in Table 4.13 and Figure 4.10 provide information about a trend of the Y-MCC system performance measured on over 11 MCC-threshold-based class maps generated based on the evaluation subset of fold1 SINS development dataset. It can be observed that the performance of the Y-MCC improved when the threshold steps moved upward up to 0.3 as a peak, afterward, it declined noticeably. Based on the 11 MCC-threshold-based class maps ranging in [0, 0.5] with 0.05 steps, the best performance yielded with the 0.3 MCC thresholds at 59.46% F1-score with the lowest ER at 0.41, which was approximately 5% higher and 0.09 lower than the 0.25 threshold. The findings clearly suggest that the MCC-threshold-based class mapping method used in the Y-MCC system effectively improved the system's performance.

To summarise, the main goal of the Y-MCC system evaluation was to determine whether the system could yield meaningful performance results toward the highly unbalanced but relatively important indoor home-based sound event classes defined in the SINS dataset. The performance results of the Y-MCC system indicated in the overall micro and macro average metrics might be insufficient against the baseline and the top 5-ranked systems released in the DCASE 2018 challenge task. However,

Table 4.13 *Table shows a statistical description of the performance of the Y-MCC method using the SINS dataset in the 10-second segment-based micro average across 11 thresholds-based class maps ranging in [0, 0.5]. The best performance was achieved in F-score at 59.5% and the lowest error rate at 0.41.*

Statistics	Micro Average F1-score (%)	Micro Average Precision (%)	Micro Average Recall (%)	Micro Average ER
Mean	36.8	50.6	36.2	0.64
Min	2.6	33.0	1.3	**0.41**
Max	**59.5**	**59.8**	**59.1**	0.99
STD	23.2	6.4	23.8	0.24

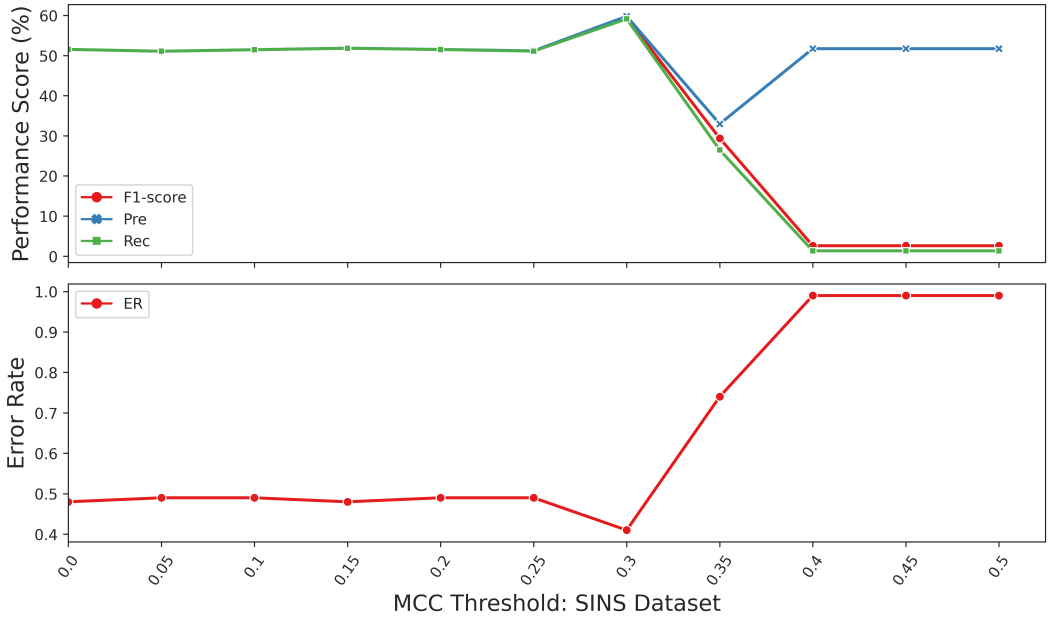

Figure 4.10 *The line plots present a trend of the overall micro averaged performance metrics of the Y-MCC system measured on the SINS development dataset. Obviously, the trend showed that over the 11 MCC-threshold-based class maps, the performance of F1-score and ER improved steadily up to the peak at 0.3 thresholds, afterward declining the performance.*

the class-wise performance results revealed that the two SINS target classes, such as 'vacuum_cleaner' and 'watching_tv,' might be significant considering the class level metrics of F1-score over 80% and ER below 0.4. Moreover, these findings were supported by the Pearson correlation score of 0.8, meaning there was a very significant correlation score between that MCC value and F1-score. Therefore, the MCC-threshold-based class mapping method used in the Y-MCC system showed its effectiveness toward the SINS dataset for resolving the uncertainty of the class map, simultaneously improving the performance.

5 Conclusion

Recent prior publications and acoustic recognition systems based on the YAMNet pre-trained model, which trained using the large-scale of the AudioSet dataset to predict 521 acoustic event classes, have successfully demonstrated the benefits of utilizing the YAMNet CNN-based machine learning model for their applications even in the healthcare field. As a common use case of the YAMNet model utilization suggested in [9], the research has employed the YAMNet classification inferences in the pre-processing stage of their classification system, named ViT, to identify cough or non-cough segments in the audio recordings. The ViT system achieved over 97% accuracy in classifying the COVID-19 symptomatic cough evaluated on the three public COVID-19 datasets. On the other hand, the feature extraction method offered by the YAMNet pre-trained model has also been explored to generalize classification systems suffering from limited training datasets. This approach has been applied in numerous studies, including classifying Alzheimer's dementia using YAMNet's feature extraction on the speech event [10] and detecting abnormal respiratory sounds using wheeze and crackle sound events [42]. Moreover, as a non-healthcare field published in [11], extensively investigated the YAMNet model by optimizing to detect numerous target classes defined in the commonly used public audio datasets: UrbanSound 8K (10 classes), ESC-10 (10 classes), and Air Compressor dataset (8 classes). However, YAMNet-based studies, including these previous pieces of literature, have failed to address the impact of class ambiguity between the YAMNet 521 classes associated with their target event classes, which could significantly affect the performance of sound classification or detection systems.

In this study, we investigated the uncertainty of the YAMNet class map based on the statistical methodology of the Matthews correlation coefficient called the Y-MCC method, primarily aiming to improve the performance of the sound event detection system. Additionally, the Y-MCC system has been designed to adapt the YAMNet model to become stretchable for both classification and polyphonic SED tasks to support innovative homecare applications. The findings of this study indicate that the Y-MCC method has achieved higher performance with advanced steps of the MCC-threshold-based class maps on the three carefully selected public sound datasets with predefined sound classes relevant to monitor the homecare environment. The overall micro-averaged performance of the Y-MCC method evaluated for the three chosen datasets: TUT-SED2016Home, ESC-50, and SINS dataset, as presented in Figure 5.1, has revealed that the large-scale of the SINS dataset has obtained the best F1-score of 59.46% and the lowest error rate of 0.41 with

the MCC-threshold-based at 0.3 class map, which is significantly better than the other two datasets: TUT-SED2016Home polyphonic real-life dataset and ESC-50 class-wise equally balanced monophonic dataset. Moreover, the general trend of performance improvement has been observed from all three datasets when the MCC threshold moved upward up to some threshold steps by reducing error rates remarkably, specifically for the TUT-SED2016Home dataset.

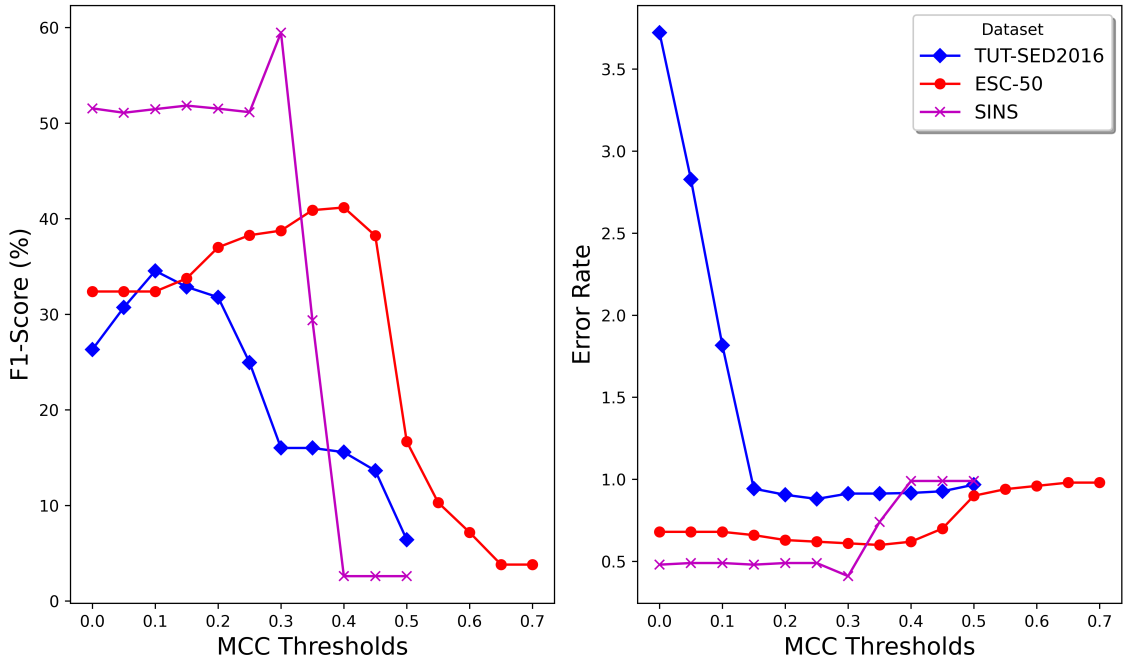

Figure 5.1 *The line plots illustrate the overall micro average metrics in F1-score and ER of the Y-MCC performance evaluated on the three datasets: TUT-SED2016Home, ESC-50, and SINS datasets. The main trend of the three datasets can be seen as the MCC-threshold-based method effectively improved the F1-score (Left) and ER (Right) for all three datasets.*

In addition, the Y-MCC class-wise performance, especially the top 3 best classes from the three datasets compared in Figure 5.2. It can be observed that four classes have F1-score higher than nearly 95% with ER lower than 0.11: three classes from ESC-50 and one from SINS. Furthermore, the outstanding 100% F1-score obtained by the '8: sheep' sound event from the ESC-50 dataset shows convincing evidence that the Y-MCC has yielded the perfect score. Apart from that, the class-level F1-score achieved by '8: vacuum_cleaner' from the SINS dataset at 94.7% is approximately close to the competing systems as the baseline and top 5 systems of the DCASE2018 Challenge task for monitoring indoor home sound activities. On the contrary, the Y-MCC performance on the TUT-SED2016Home dataset might look insignificant. Still, it could be considered an average score obtained by the top 9 systems ranked in the DCASE 2016 Challenge task for sound event detection in real-life audio. Therefore, with this compelling evidence for improvement of the

overall and class-wise performance, the Y-MCC approach has been successful for the sound event detection system by statistically resolving the inaccurate classes in the top probability predictions provided by the YAMNet pre-trained model.

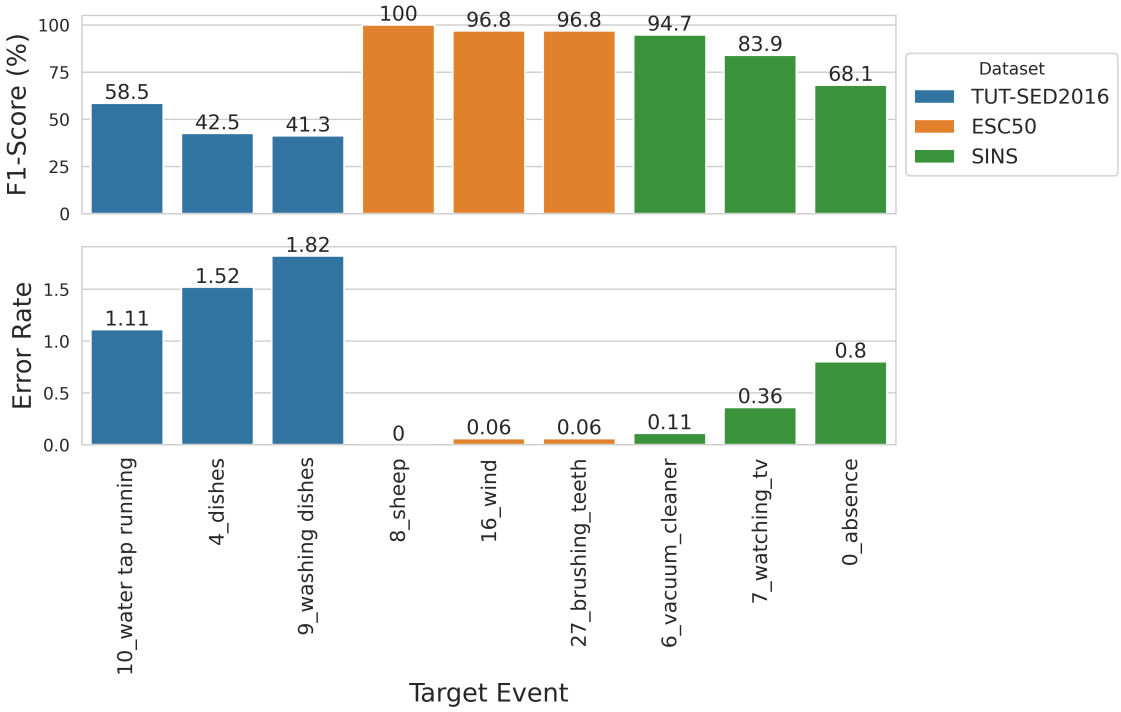

Figure 5.2 *Class-wise top 3 best performance evaluated by the Y-MCC system on the three datasets, TUT-SED 2016, ESC-50, and SINS, are presented in terms of class-level maximum F1-score (Top) and corresponding ER (Down), which indicate the compelling evidence of the Y-MCC method successfully performed SED tasks specifically those classes over 90% F1-score, including '8: sheep' at 100% F1-score and zero ER in ESC-50 dataset.*

To our knowledge, this thesis is the first investigation of the erroneous YAMNet predictions for the predefined classes in the three benchmark datasets, which mainly contain domestic indoor home sound events. This study was carried out to assess the potential benefit of the pre-trained YAMNet model that could be adapted to the sound event detection system for facilitating smart homecare applications. However, this method has several weaknesses notified by no detection on some classes, 7 out of the 50 target classes in the ESC-50 dataset, and the low performance on some essential sound activities necessary to support elderly homecare, particularly related to human non-speech sound activities. Therefore, future work should include a follow-up study to strengthen the Y-MCC method to perform the commercial level of the overall detection score, especially for essential sound activities considered important to monitor the health of elderly people in their homecare environment.

References

[1] S. Krstulović, "Audio Event Recognition in the Smart Home," in *Computational Analysis of Sound Scenes and Events*, T. Virtanen, M. D. Plumbley, and D. Ellis, Eds., Cham: Springer International Publishing, 2018, pp. 335–371.

[2] E. Rudnicka, P. Napierała, A. Podfigurna, B. Męczekalski, R. Smolarczyk, and M. Grymowicz, "The World Health Organization (WHO) approach to healthy ageing," *Maturitas*, vol. 139, pp. 6–11, 2020.

[3] S. Spasova, R. Baeten, and B. Vanhercke, "Challenges in long-term care in europe," *Eurohealth*, vol. 24, no. 4, pp. 7–12, 2018.

[4] D. D. Furszyfer Del Rio, B. K. Sovacool, N. Bergman, and K. E. Makuch, "Critically reviewing smart home technology applications and business models in Europe," *Energy Policy*, vol. 144, p. 111 631, 2020.

[5] 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 Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 2, pp. 379–393, 2018.

[6] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "FSD50K: An Open Dataset of Human-Labeled Sound Events," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 829–852, 2022.

[7] R. Y. M. Li, H. C. Y. Li, C. K. Mak, and T. B. Tang, "Sustainable Smart Home and Home Automation: Big Data Analytics Approach," *International Journal of Smart Home*, vol. 10, no. 8, pp. 177–198, 2016.

[8] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson, "CNN architectures for large-scale audio classification," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, USA: IEEE Press, 2017, pp. 131–135.

[9] N. Sobahi, O. Atila, E. Deniz, A. Sengur, and U. R. Acharya, "Explainable COVID-19 detection using fractal dimension and vision transformer with Grad-CAM on cough sounds," *Biocybernetics and Biomedical Engineering*, vol. 42, no. 3, pp. 1066–1080, 2022.

[10] Y. Zhu, X. Liang, J. A. Batsis, and R. M. Roth, "Exploring Deep Transfer Learning Techniques for Alzheimer's Dementia Detection," *Frontiers in Computer Science*, vol. 3, 2021.

[11] E. Tsalera, A. Papadakis, and M. Samarakou, "Comparison of Pre-Trained CNNs for Audio Classification Using Transfer Learning," *Journal of Sensor and Actuator Networks*, vol. 10, no. 4, p. 72, 2021.

[12] H. Soini, P. Routasalo, and H. Lagström, "Characteristics of the Mini-Nutritional Assessment in elderly home-care patients," *European Journal of Clinical Nutrition*, vol. 58, no. 1, pp. 64–70, 2004.

[13] P. Sundaravadivel, E. Kougianos, S. P. Mohanty, and M. K. Ganapathiraju, "Everything You Wanted to Know about Smart Health Care: Evaluating the Different Technologies and Components of the Internet of Things for Better Health," *IEEE Consumer Electronics Magazine*, vol. 7, no. 1, pp. 18–28, 2018.

[14] H. M. Do, M. Pham, W. Sheng, D. Yang, and M. Liu, "RiSH: A robot-integrated smart home for elderly care," *Robotics and Autonomous Systems*, vol. 101, pp. 74–92, 2018.

[15] A. Banjar, H. Dawood, A. Javed, and F. Hassan, "Fall event detection using the mean absolute deviated local ternary patterns and BiLSTM," *Applied Acoustics*, vol. 192, p. 108 725, 2022.

[16] J. Vanus, J. Belesova, R. Martinek, J. Nedoma, M. Fajkus, P. Bilik, and J. Zidek, "Monitoring of the daily living activities in smart home care," *Human-centric Computing and Information Sciences*, vol. 7, no. 1, p. 30, 2017.

[17] L.-N. Kwon, D.-H. Yang, M.-G. Hwang, S.-J. Lim, Y.-K. Kim, J.-G. Kim, K.-H. Cho, H.-W. Chun, and K.-W. Park, "Automated Classification of Normal Control and Early-Stage Dementia Based on Activities of Daily Living (ADL) Data Acquired from Smart Home Environment," *International Journal of Environmental Research and Public Health*, vol. 18, no. 24, p. 13 235, 2021.

[18] M. Hartmann, U. S. Hashmi, and A. Imran, "Edge computing in smart health care systems: Review, challenges, and research directions," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 3, e3710, 2022.

[19] T. Virtanen, M. D. Plumbley, and D. Ellis, Eds., *Computational Analysis of Sound Scenes and Events*. Cham, Switzerland: Springer International Publishing, 2018.

[20] D. C. Peterson, V. Reddy, and R. N. Hamel, "Neuroanatomy, Auditory Pathway," in *StatPearls*, Treasure Island (FL): StatPearls Publishing, 2023.

[21] I. Nelken, J. Bizley, S. A. Shamma, and X. Wang, "Auditory Cortical Processing in Real-World Listening: The Auditory System Going Real," *Journal of Neuroscience*, vol. 34, no. 46, pp. 15 135–15 138, 2014.

[22] R. Munkong and B.-H. Juang, "Auditory perception and cognition," *IEEE Signal Processing Magazine*, vol. 25, no. 3, pp. 98–117, 2008.

[23] N. Staeren, H. Renvall, F. De Martino, R. Goebel, and E. Formisano, "Sound categories are represented as distributed patterns in the human auditory cortex," *Current Biology*, vol. 19, no. 6, pp. 498–502, 2009.

[24] J. P. Rauschecker and B. Tian, "Mechanisms and streams for processing of "what" and "where" in auditory cortex," *Proceedings of the National Academy of Sciences*, vol. 97, no. 22, pp. 11 800–11 806, 2000.

[25] J. G. Betts, P. Desaix, E. Johnson, J. E. Johnson, O. Korol, D. Kruse, B. Poe, J. A. Wise, M. Womble, and K. A. Young, *Anatomy and Physiology*. Texas, USA: OpenStax, p. 1420.

[26] J. O. Pickles, *Introduction to the Physiology of Hearing*. Bradford, UK: Brill, 2012.

[27] F. Alías, J. C. Socoró, and X. Sevillano, "A Review of Physical and Perceptual Feature Extraction Techniques for Speech, Music and Environmental Sounds," *Applied Sciences*, vol. 6, no. 5, p. 143, 2016.

[28] R. M. Rangayyan, *Biomedical Signal Analysis*, Second edition. New Jersey, USA: John Wiley & Sons, 2015.

[29] S. M. Alessio, *Digital Signal Processing and Spectral Analysis for Scientists* (Signals and Communication Technology). Cham: Springer International Publishing, 2016.

[30] G. Heinzel, A. Rudiger, and R. Schilling, "Spectrum and spectral density estimation by the Discrete Fourier transform (DFT), including a comprehensive list of window functions and some new flat-top windows.," Internal Report, Max-Planck-Institut fur Grav-itationsphysik, Hannover, Tech. Rep., 2002.

[31] N. Turpault, R. Serizel, J. Salamon, and A. P. Shah, "Sound Event Detection in Domestic Environments with Weakly Labeled Data and Soundscape Synthesis," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, New York University, 2019, pp. 253–257.

[32] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.

[33] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "Librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, vol. 8, 2015.

[34] T. Heittola, Annamaria Mesaros, T. Virtanen, and M. Gabbouj, "Supervised model training for overlapping sound events based on unsupervised source separation," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, Vancouver, BC, Canada, 2013, pp. 8677–8681.

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

[36] E. Cakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional Recurrent Neural Networks for Polyphonic Sound Event Detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1291–1303, 2017.

[37] 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 Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 6, pp. 992–1006, 2019.

[38] G. Dekkers, L. Vuegen, T. van Waterschoot, B. Vanrumste, and P. Karsmakers, "DCASE 2018 Challenge - Task 5: Monitoring of domestic activities based on multi-channel acoustics," KU Leuven, Tech. Rep., 2018. arXiv: 1807.11246.

[39] G. Dekkers, S. Lauwereins, B. Thoen, M. W. Adhana, H. Brouckxon, B. Van den Bergh, T. van Waterschoot, B. Vanrumste, M. Verhelst, and P. Karsmakers, "The SINS database for detection of daily activities in a home environment using an Acoustic Sensor Network," in *Detection and Classification of Acoustic Scenes and Events 2017*, DCASE Workshop, 2017.

[40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd International Conference on Learning Representations (ICLR 2015)*, 2015.

[41] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv:1704.04861 [cs]*, 2017.

[42] U. Tiwari, S. Bhosale, R. Chakraborty, and S. K. Kopparapu, "Deep Lung Auscultation Using Acoustic Biomarkers for Abnormal Respiratory Sound Event Detection," in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 1305–1309.

[43] H. Xue and F. D. Salim, "Exploring Self-Supervised Representation Ensembles for COVID-19 Cough Classification," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 1944–1952.

[44] B. M. Rocha, D. Filos, L. Mendes, I. Vogiatzis, E. Perantoni, E. Kaimakamis, P. Natsiavas, A. Oliveira, C. Jácome, A. Marques, R. P. Paiva, I. Chouvarda, P. Carvalho, and N. Maglaveras, "A Respiratory Sound Database for the Development of Automated Classification," in *Precision Medicine Powered by pHealth and Connected Health*, N. Maglaveras, I. Chouvarda, and P. de Carvalho, Eds., ser. IFMBE Proceedings, Singapore: Springer, 2018, pp. 33–37.

[45] A. Menon, K. Mehrotra, C. K. Mohan, and S. Ranka, "Characterization of a Class of Sigmoid Functions with Applications to Neural Networks," *Neural Networks*, vol. 9, no. 5, pp. 819–835, 1996.

[46] S. Raschka and V. Mirjalili, *Python Machine Learning - Second Edition*. Birmingham: Packt Publishing, 2017, vol. 2nd ed.

[47] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *2016 24th European Signal Processing Conference (EUSIPCO)*, 2016, pp. 1128–1132.

[48] R. Serizel, V. Bisot, S. Essid, and G. Richard, "Acoustic Features for Environmental Sound Analysis," in *Computational Analysis of Sound Scenes and Events*, T. Virtanen, M. D. Plumbley, and D. Ellis, Eds., Cham: Springer International Publishing, 2018, pp. 71–101.

[49] E. Çakir and T. Virtanen, "End-to-End Polyphonic Sound Event Detection Using Convolutional Recurrent Neural Networks with Learned Time-Frequency Representation Input," in *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1–7.

[50] E. Cakir, E. C. Ozan, and T. Virtanen, "Filterbank learning for deep neural network based polyphonic sound event detection," in *2016 International Joint Conference on Neural Networks (IJCNN)*, 2016, pp. 3399–3406.

[51] G. Richard, S. Sundaram, and S. Narayanan, "An Overview on Perceptually Motivated Audio Indexing and Classification," *Proceedings of the IEEE*, vol. 101, no. 9, pp. 1939–1954, 2013.

[52] J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," *Mathematics of computation*, vol. 19, no. 90, pp. 297–301, 1965.

[53] M. T. Heideman, D. H. Johnson, and C. S. Burrus, "Gauss and the History of the Fast Fourier Transform," *Archive for History of Exact Sciences*, vol. 34, no. 3, pp. 265–277, 1985.

[54] M. Kahrs and K. Brandenburg, *Applications of Digital Signal Processing to Audio and Acoustics*. New York, NY, USA: Springer, 1998.

[55] S. S. Stevens, J. Volkmann, and E. B. Newman, "A Scale for the Measurement of the Psychological Magnitude Pitch," *The Journal of the Acoustical Society of America*, vol. 8, no. 3, pp. 185–190, 1937.

[56] B. Nasersharif and A. Akbari, "A framework for robust mfcc feature extraction using snr-dependent compression of enhanced mel filter bank energies," in *Ninth International Conference on Spoken Language Processing*, 2006.

[57] A. Kumar, M. Khadkevich, and C. Fügen, "Knowledge Transfer from Weakly Labeled Audio Using Convolutional Neural Network for Sound Events and Scenes," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 326–330.

[58] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[59] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proceedings of the 22nd ACM International Conference on Multimedia*, ser. MM '14, Association for Computing Machinery, Orlando, Florida, USA, 2014, pp. 1041–1044.

[60] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proceedings of the 23rd ACM International Conference on Multimedia*, ser. MM '15, New York, NY, USA, 2015, pp. 1015–1018.

[61] Q. Kong, C. Yu, Y. Xu, T. Iqbal, W. Wang, and M. D. Plumbley, "Weakly Labelled AudioSet Tagging With Attention Neural Networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 11, pp. 1791–1802, 2019.

[62] 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 *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 776–780.

[63] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, Boston, MA, USA, 2015, pp. 1–6.

[64] A. Kumar and V. Ithapu, "A Sequential Self Teaching Approach for Improving Generalization in Sound Event Recognition," in *Proceedings of the 37th International Conference on Machine Learning*, PMLR, 2020, pp. 5447–5457.

[65] A. R. Webb, K. D. Copsey, and G. Cawley, *Statistical Pattern Recognition*. Hoboken: John Wiley and Sons, Incorporated, 2011.

[66] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for Polyphonic Sound Event Detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016.

[67] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, p. 6, 2020.

[68] S. Boughorbel, F. Jarray, and M. El-Anbari, "Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric," *PloS One*, vol. 12, no. 6, e0177678, 2017.

[69] A. Mesaros, T. Heittola, T. Virtanen, and M. D. Plumbley, "Sound Event Detection: A tutorial," *IEEE Signal Processing Magazine*, vol. 38, no. 5, pp. 67–83, 2021.

[70] B. W. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *Biochimica et Biophysica Acta (BBA) - Protein Structure*, vol. 405, no. 2, pp. 442–451, 1975.

[71] R. Boethling, "Comparison of ready biodegradation estimation methods for fragrance materials," *Science of The Total Environment*, vol. 497-498, pp. 60–67, 2014.

[72] P. Baldi, S. Brunak, Y. Chauvin, C. A. F. Andersen, and H. Nielsen, "Assessing the accuracy of prediction algorithms for classification: An overview," *Bioinformatics*, vol. 16, no. 5, pp. 412–424, 2000.

[73] D. Chicco, M. J. Warrens, and G. Jurman, "The Matthews Correlation Coefficient (MCC) is More Informative Than Cohen's Kappa and Brier Score in Binary Classification Assessment," *IEEE Access*, vol. 9, pp. 78 368–78 381, 2021.

[74] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[75] S. Adavanne, G. Parascandolo, P. Pertilä, T. Heittola, and T. Virtanen, "Sound Event Detection in Multichannel Audio Using Spatial and Harmonic Features," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*, Tampere University of Technology. Department of Signal Processing, 2016, pp. 6–10.