Kirigami: Lightweight Speech Filtering for Privacy-Preserving Activity Recognition using Audio

SUDERSHAN BOOVARAGHAVAN, Carnegie Mellon University, United States
HAOZHE ZHOU, Carnegie Mellon University, United States
MAYANK GOEL, Carnegie Mellon University, United States
YUVRAJ AGARWAL, Carnegie Mellon University, United States

Audio-based human activity recognition (HAR) is very popular because many human activities have unique sound signatures that can be detected using machine learning (ML) approaches. These audio-based ML HAR pipelines often use common featurization techniques, such as extracting various statistical and spectral features by converting time domain signals to the frequency domain (using an FFT) and using them to train ML models. Some of these approaches also claim privacy benefits by preventing the identification of human speech. However, recent deep learning-based automatic speech recognition (ASR) models pose new privacy challenges to these featurization techniques. In this paper, we systematically evaluate various featurization approaches for audio data, assessing their privacy risks through metrics like speech intelligibility (PER and WER) while considering the utility tradeoff in terms of ML-based activity recognition accuracy. Our findings reveal the susceptibility of these approaches to speech content recovery when exposed to recent ASR models, especially under re-tuning or retraining conditions. Notably, fine-tuned ASR models achieved an average Phoneme Error Rate (PER) of 39.99% and Word Error Rate (WER) of 44.43% in speech recognition for these approaches. To overcome these privacy concerns, we propose Kirigami, a lightweight machine learning-based audio speech filter that removes human speech segments reducing the efficacy of ASR models (70.48% PER and 101.40% WER) while also maintaining HAR accuracy (76.0% accuracy). We show that Kirigami can be implemented on common edge microcontrollers with limited computational capabilities and memory, providing a path to deployment on a variety of IoT devices. Finally, we conducted a real-world user study and showed the robustness of Kirigami on a laptop and an ARM Cortex-M4F microcontroller under three different background noises.

CCS Concepts: • **Security and privacy** → **Privacy-preserving protocols**; • **Human-centered computing** → *Ubiquitous and mobile computing*;

Additional Key Words and Phrases: Privacy, Acoustics, Internet of Things, Ubiquitous Sensing

1 INTRODUCTION

Audio-based ambient sensing approaches are increasingly prevalent in various application domains, such as personal health monitoring [21, 26, 31], ambient environmental sensing [2, 5, 20, 29] and energy efficiency optimization [8, 40] showcasing its growing significance in enhancing our overall quality of life. However, the increased reliance on audio data in these applications has raised substantial privacy concerns, especially considering the inherently sensitive nature of audio data and its potential to capture human speech. For instance, users of smart speakers have shown dissatisfaction with the storage of sound on servers or sharing data with third parties [12, 27]. To address these concerns, a common strategy is to apply a combination of audio data featurization and speech filtering approaches, preferably at the edge, to reduce privacy concerns while still ensuring the utility [8, 25, 26, 27, 29]. These methods attempt to process the raw audio signal locally on the device and extract useful information from the audio signal while impeding speech information from leaving the device. While these approaches have shown promising results in protecting user privacy by evaluating the intelligibility of the processed audio, the emergence of deep-learning-based Automatic Speech Recognition (ASR) [1, 18, 36] models poses new privacy challenges. While a human might not be able to decipher a featurized audio file, a machine learning model trained/tuned on featurized data might be able to recognize the speech content. ASR

Authors' addresses: Sudershan Boovaraghavan, sudershan@cmu.edu, Carnegie Mellon University, Pittsburgh, United States; Haozhe Zhou, haozhezh@andrew.cmu.edu, Carnegie Mellon University, Pittsburgh, United States; Mayank Goel, mayankgoel@cmu.edu, Carnegie Mellon University, Pittsburgh, United States; Yuvraj Agarwal, yuvraj@cs.cmu.edu, Carnegie Mellon University, Pittsburgh, United States.

ACM 2474-9567/2024/3-ART36
https://doi.org/10.1145/3643502

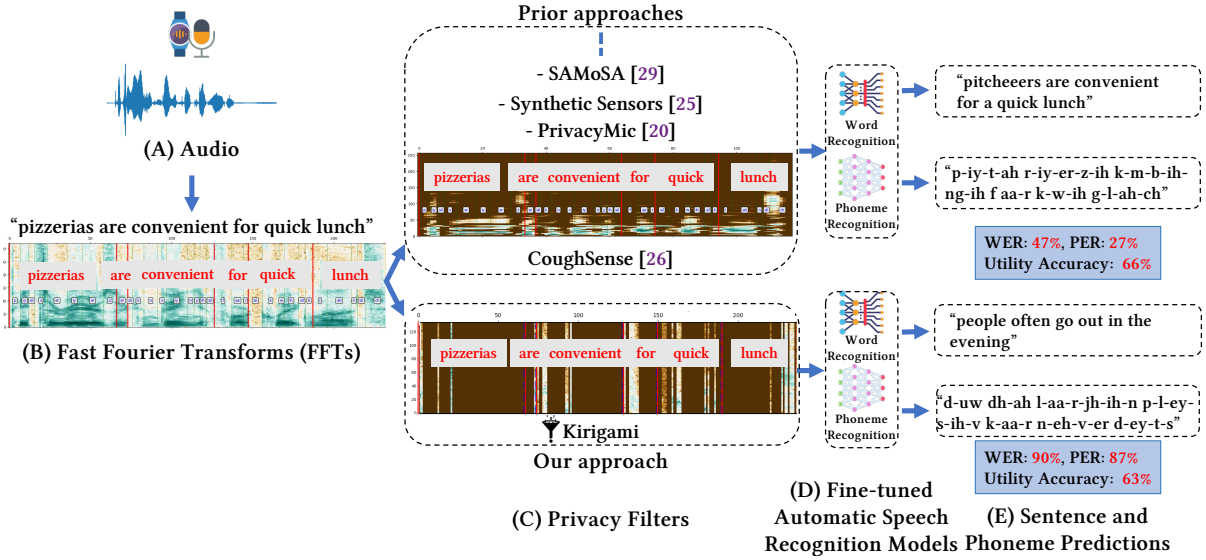


Fig. 1. Ambient sensing applications that use microphones break down raw audio data (A) into its component frequencies, FFTs (B), for meaningful feature extraction. Prior approaches have suggested filters (C) to eliminate speech while retaining other data for privacy preservation. However, our study reveals that Automatic Speech Recognition (ASR) algorithms (D) can be fine-tuned to recognize speech and phoneme information to an extent (E), demonstrating the limitation of such featurization techniques. Our novel approach, Kirigami, mitigates the privacy risks of speech inferences from ASR-based models by identifying and filtering likely speech segments on the edge while maintaining the accuracy of audio-based human activity recognition tasks.

models, such as Whisper [36] or Wav2Vec [1], are trained on thousands of hours of multilingual speech data to improve their robustness to different accents, background noise, and diverse languages. These modern ASR systems are not limited to recognizing speech from raw audio but can be tuned to recognize speech content specifically from the transformed audio, which was previously thought to be safe from privacy breaches, as we show in this paper. As the outputs of prior approaches are not examined on modern fine-tuned ASR models, the privacy implications remain unclear.

This paper evaluates the privacy risks posed by recent ASR models on prior audio filtering and on-the-edge techniques. We analyzed four prior approaches representing different types of privacy-focused filtering techniques on audio data [26], [20], [25] and [29]. We replicated these approaches and passed their featured data respectively through fine-tuned ASR models, such as Wav2Vec2.0 [1] and Whisper AI [36]. We found that even after applying the filters from prior approaches, some of the speech-related information, such as residual parts of phonemes [3] (unit of sound that can distinguish words), were still present, and the ASR models were able to reveal *some* information about the original speech as shown in Figure 1. To contextualize the privacy risks of existing audio filtering approaches, we asked 10 participants to answer questions about the speech content of the recovered speech information and found that the majority of the participants reasonably identified the topic or content of speech, further highlighting the privacy risks. Until we can run a powerful ASR model such as Whisper AI on the edge to remove speech-based audio, there is an urgent need for a lightweight approach to filter out sensitive audio at the edge.

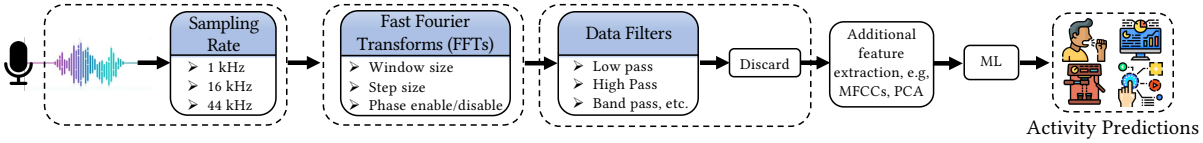


Fig. 2. An architecture of an audio-based activity recognition approach that takes featurized audio as inputs for applications such as cough recognition or event detection

To overcome these challenges, we present Kirigami, a lightweight edge-compatible speech filter that effectively removes probable speech content while preserving non-speech content to maintain high utility value for activity recognition applications. Unlike existing solutions that may still reveal residual speech information, Kirigami takes a more conservative approach of completely discarding likely audio content on the edge. For this reason, we believe that Kirigami will remain effective even as ASR models become increasingly sophisticated in the future. Furthermore, we demonstrate Kirigami's effective adaptation to real-world environments through an innovative background masking method, enhancing its ability to filter ambient sounds before processing speech events. Moreover, Kirigami allows adaptability through custom post-filter featurization methods, allowing users to users to customize the filter for specific application needs using techniques like Log-Mel Spectrogram and Mel-Frequency Cepstral Coefficients. This flexibility enhances Kirigami's applicability across a range of scenarios. Our results demonstrate that Kirigami is highly effective in suppressing human speech inference even when using fine-tuned ASR models. Kirigami can run on low memory, is computationally efficient, and has been tested and verified on embedded hardware platforms, making it a viable solution for real-world IoT use cases.

Overall, we make the following contributions:

- We systematically examine the privacy risks associated with various existing privacy-focused featurization approaches. We show that, on average, fine-tuned ASR models can recognize speech with a 39.99% Phoneme Error Rate (PER) and 44.43% Word Error Rate (WER) for these approaches, posing significant privacy risks.
- We conducted a user study to contextualize the privacy implications of PER and WER values. On average, 90% of the participants were able to infer the speech topic when the predicted sentence had a WER below 80%. Additionally, 80 % of participants could deduce the sentence topic from its phoneme prediction up to 60% PER, revealing privacy risks in current audio filtering. Based on our user study results, we derive a privacy cut-off of PER at 60% and WER at 80%, above which little information about speech can be retrieved
- We present Kirigami[1], a lightweight machine learning-based audio speech filter, which removes likely human speech segments while preserving other sounds to maintain high activity recognition accuracy. We evaluate Kirigami with fine-tuned state-of-the-art ASR models and show that our Kirigami filter achieves, on average, 70.48% PER and 101.40% WER while ensuring 76.0% accuracy on activity recognition applications. In addition, Kirigami can be implemented on inexpensive and resource-constrained microcontrollers, making it deployable on a wide range of IoT edge devices.
- Finally, we conducted a real-world user study to assess the robustness of Kirigami in filtering speech in environments with diverse background noise while also ensuring the preservation of activity recognition accuracy, thereby demonstrating the consistent performance of Kirigami in the real world.

2 BACKGROUND

In this section, we introduce the common audio-based ambient sensing solutions and discuss their architecture, with a focus on their feature engineering approaches to denature the audio data.

[1] www.github.com/synergylabs/kirigami

2.1 Microphone for Activity Recognition

Microphones are widely used in ambient sensing to support various applications surrounding Human Activity Recognition(HAR), such as health monitoring [26, 28, 42, 43], monitoring the number of people present in a building, and identification of room occupancy and activities of people [8, 24, 39]. In addition, audio capture can be utilized for assistive services, particularly for populations with hearing disabilities, where it can be used for audio scene analysis, audible event alerts, and new wearable devices that work in conjunction with microphones inside smart buildings [33]. In such applications, the collected ambient audio data is often denatured to ensure sensitive information, such as speech, is not collected or sent to the cloud.

A typical ambient sensing solution with a microphone consists of three main sub-components: *audio sub-sampling*, *audio featurization*, and *data filters*, as illustrated in Figure 2. The audio sub-sampling component records the time domain ambient audio at different configurable sampling rates based on the application. This raw audio data is then passed through an audio featurization algorithm to convert them into frequency domain signals to reduce the data dimensionality and in some cases to denature the data. In general, Fast Fourier Transforms (FFTs) are used to convert the raw audio signal into a frequency domain representation critical for extracting useful information from the audio signal. Next, various data filtering approaches, including low pass filters, can be employed to eliminate mid to low-frequency bands containing speech information or background noise from the processed signals. The filtered FFT data is then subject to further feature extraction using various audio processing techniques such as Mel Frequency Cepstral Coefficients (MFCCs), filter banks, or spectral features. These extracted features are then used by ML models to classify and recognize audio based events or activities.

2.1.1 Privacy v.s. Utility Tradeoffs for Speech Filters. While featurized data can be sent to a cloud backend, where speech can then be filtered, it is generally considered better to do this on the edge [4] for privacy to prevent speech data from being sent in the first place. In addition, any filtering approach needs to balance privacy (i.e. detecting actual speech segments) and utility (i.e. avoiding filtering non-speech segments) to be useful for real-world activity recognition. In general, audio-based speech filtering approaches can be categorized into four types: *time domain based*, *frequency domain based*, *feature-based*, and *model-based*.

Time domain-based filtering involves techniques such as reducing the audio signal's sampling rate or calculating statistical values such as minimum, maximum, std-dev for a time-period of values. While doing so may be effective at protecting speech privacy, it reduces the utility for downstream HAR applications since sub-sampling can remove important features of the audio signal (e.g. high frequency signals) and not provide enough information to detect activities. Feature-based filtering, extracts specific speech features to speech, such as a spectral envelope or harmonic structure, and use them to remove speech segments. This approach can still affect utility, as it may filter out non-speech sounds. Alternatively, a model-based approach uses ML models to recognize and filter speech. This approach can be highly effective but is computationally expensive, and not available on edge devices with limited computational power and storage.

2.2 Phonemes in Audio

A phoneme is a perceptually distinct unit of sound, that can distinguish one word from another in a specified language. When speaking, various phonemes can be produced by adjusting the air passage in the vocal tract. Consonant sounds result from restricting the airflow, such as using different lip, tongue, or teeth positions, whereas vowel sounds occur when the airflow is less restricted and the mouth is more open [30]. Understanding the phoneme structure of a language is crucial in various areas, such as linguistics, speech recognition, and natural language processing. Typically, the English language is composed of 44 phonemes, including 24 consonants and 20 vowels [3]. The 39-phoneme set illustrated in Figure 3, derived from the Carnegie Mellon Pronouncing Dictionary (CMUdict) [10], is widely employed in various ASR and NLP applications. Figure 4 shows the unique frequency spectrum signature that indicates the corresponding phoneme. For example, the frequency spectrum

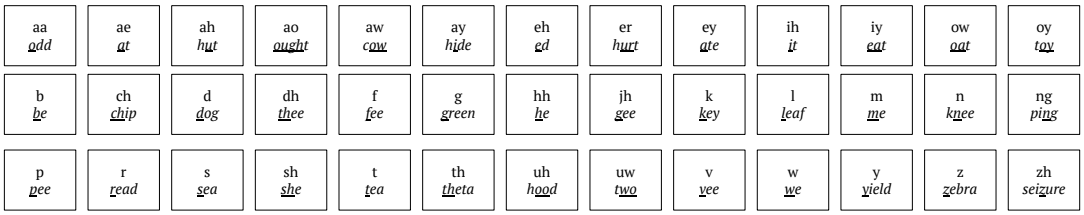


Fig. 3. The 39-phoneme table of CMUdict[10]

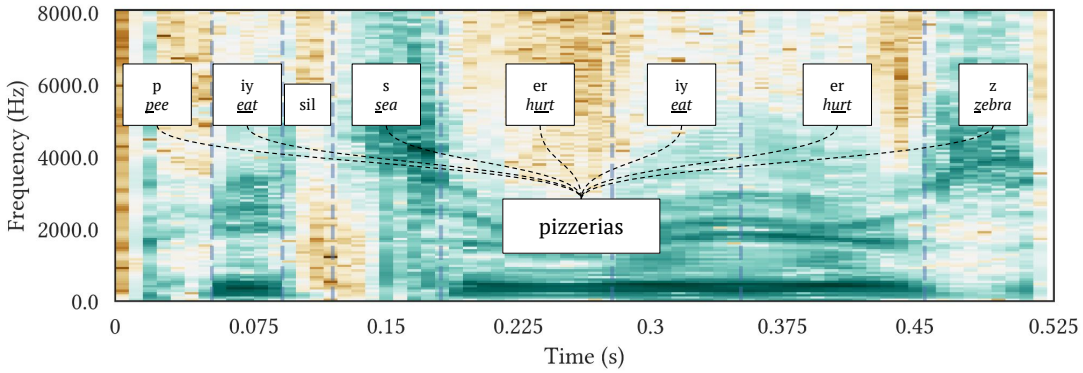


Fig. 4. The phoneme and grapheme of an example word "pizzerias".Phonemes are the individual speech sounds composing words, while graphemes are the corresponding letters or letter groups representing those sounds.

for the sound *z* shows that almost all of the frequency spectrum values are activated in comparison to the other phoneme information. Overall, phonemes as features play an important role in speech recognition tasks.

2.3 Deep-learning-based Automatic Speech Recognition Models

Recently, there have been significant strides in Automatic Speech Recognition (ASR) models making them even more accurate. Deep learning models, such as recurrent neural networks (RNNs), convolutional neural networks (CNNs), and transformer models, have been trained on large speech corpora and have demonstrated state-of-the-art performance on a variety of benchmark datasets [15, 32]. More recently, large ASR models such as Wav2Vec [1] have been trained on several hundred thousand hours of multilingual speech data, increasing their robustness to different accents, background noise, and diverse languages. Leveraging the advantage of pretraining, these models can quickly learn general representations of speech patterns and phonetic features (see Fig 4), even with low-fidelity or denatured data, as well as handle variations in speaker accent and speech rate. For example, Whisper [36] combines an encoder-decoder model with a context network to improve the modeling of long-term dependencies in speech. Wav2Vec uses contrastive learning to train a model to distinguish between a correctly aligned speech segment and a randomly sampled corrupted segment [1]. Similarly, other transformer-based models such as BERT [9] and GPT models [14] can also be fine-tuned to achieve impressive results for many down-stream tasks, including ASR [19, 44]. These advancements pose a significant challenge to edge speech featurization, and filtering approaches as modern ASR systems can be fine-tuned to potentially identify speech content even from transformed audio data, as we show in this paper.

2.4 Threat Model

Based on our review of the relevant literature on audio privacy techniques used in several activity recognition applications [5, 20, 29, 42], we consider a sophisticated adversary (following the Dolev-Yao model [11]) who has full knowledge of the audio featurization methods used by the device or application. We also assume the adversary wants to extract speech from the featurized and transformed data sent by a device that senses audio. We assume that the adversary does not have direct access to compromise the device itself (i.e., it cannot change its firmware). We also assume that the adversary has the knowledge to replicate the same audio transformation on the speech datasets to use as training data. The adversary can also try to invert any transformation using approximation techniques, for example, using inverse Fourier Transforms or inverse Principal Component Analysis (PCA). The adversary has access to public speech datasets, such as TIMIT [15] and Librispeech [32], to public ASR models [37], and can even fine-tune these models. Finally, we assume that the adversary has access to featurized/filtered audio. In a real-world scenario, an adversary capable of launching such an attack can be, for example, an application that uses audio data to identify activities such as cough, an honest-but-curious audio-to-HAR cloud API service provider, or an external hacker. Based on these assumptions, we consider three potential adversarial scenarios:

S1 - Scenario requiring low effort: An adversary downloads a pre-trained ASR model (no fine-tuning). Then, they try and reverse-engineer the featurized and filtered audio data using an inverse PCA or inverse FFT. The resulting data is in a format that the various ASR models expect, and the adversary passes the data to them to infer speech segments.

S2 - Scenario requiring moderate effort: An adversary downloads a pre-trained ASR model. They fine-tune the ASR model by replicating the same audio featurization techniques used on an annotated speech dataset to create a training set. The adversary also creates a pipeline to transform the featurized audio data into the same shape as the ASR model requires. During training, the adversary re-trains a small subset of the layers on the ASR model with pre-trained weights loaded while freezing the gradients of the rest of the model. Finally, the adversary processes the target audio data into the same shape dimensions as the original ASR model to infer speech.

S3 - Scenario requiring high effort: An adversary downloads a pre-trained ASR model. To fine-tune the ASR model, the adversary replicates the audio featurization techniques on an annotated speech dataset to create a training set. The adversary also creates a pipeline that can transform the featurized audio data into the same shape as the ASR model requires. The adversary trains *all the layers* on the ASR model with pre-trained weights loaded. Finally, the adversary processed the target audio file into the same shape dimensions as what the original network was trained on.

3 RELATED WORK

Prior works that use audio for activity recognition have proposed different methods to protect audio privacy while preserving the utility of detecting activities.

Table 1. Summary of evaluated speech filtering approaches

Filtering Approach	Fourier Transform Configuration		Filter Type	Filter Summary
	Window-Size	Stride-Size		
CoughSense [26]	512	256	Feature	STFT concatenation (150ms), PCA (10 components)
Synthetic Sensors [25]	256	128	Frequency Domain	Reduced FFT (10 windows/s)
PrivacyMic [20]	256	128	Frequency Domain	Low Pass Filter (<300 Hz)
SAMoSA [29]	600	30	Time Domain	Subsampling (1kHz)

3.1 Audio Privacy Filters for Activity Recognition

Researchers have proposed various audio filtering methods to remove speech information from audio, including data degradation techniques to sample FFTs at a lower rate or completely drop FFT data from a certain frequency band. These approaches aim to protect user speech data while allowing other audio data that are useful for activity recognition, but their efficacy varies significantly, and their limitations must be considered. Table 1 shows the summary of different speech filtering approaches presented in the prior work.

Coughsense [26] is one of the earliest works to propose a cough detection system that utilizes a low-cost microphone to detect coughs accurately in real-time. They reduced speech intelligibility by aggressively aggregating 150ms of sound and extracting ten principal components from principal component analysis on cough sounds. They showed that their system could classify coughs with high accuracy. Iravantchi *et al.* [20] proposed a daily activity recognition system that utilizes inaudible frequencies in the audio signals to preserve privacy. Such an approach requires special microphone sensors that capture ultrasonic and infrasonic sounds and the usual microphone that collects sounds in the audible range. Filters can be implemented on microphone hardware to filter out frequencies from 300 Hz to 8kHz.

Another line of work focuses on more generic transformations to hide speech in audio. Chen *et al.* [7] suggested a method to filter speech from audio by replacing the vocal tract transfer function of vowel regions in audio with the transfer function from prerecorded vowels. SoundShredding [23] proposed a privacy-preserving audio transformation in which the order of frames from MFCC features is randomized. The commonly used method of evaluation in these approaches includes recruiting participants to listen to the processed audio clips and examine if any speech content can be picked or to rate the extent of clarity of the audio clips. Another approach is to pass through an existing speech-to-text service such as Google Speech Recognition. While the ability to recognize speech was shown to be limited after using these transformations was evaluated under human listening experiment [7, 20, 26, 29] or passing through existing speech-to-text API [20, 29], the threat from recent powerful machine-learning-based ASR models was not considered.

4 FEASIBILITY OF INFERRING SPEECH FROM FILTERED OR FEATURIZED AUDIO

Two reasons to evaluate the feasibility of ASR models to infer speech from featurized audio are: (a) our observation of the residual phoneme information available in the output filtered data from prior approaches, and (b) the ability of the deep learning-based ASR models to be fine-tuned and learn from featurized data. In prior approaches, a speech filter such as a time or frequency-domain filter is applied to an audio signal. These approaches often remove certain frequency components associated with human speech. However, not all phoneme information is removed, and some residual information may remain in the filtered signal. Figure 5 shows the spectrogram of the data after prior filtering approaches are applied. This residual phoneme information can be seen in the form of different acoustic characteristics, such as spectral shapes specific to spoken words or phonemes. For example, the FFT output after applying the CoughSense filter [26], for the phoneme *"iy"* has unique spectral patterns still present around the 1 kHz frequency range. Similarly, for PrivacyMic [20], we can see that unique spectral patterns are present around the 250 Hz frequency range for the same *"iy"* phoneme. An adversary can exploit this residual phoneme information by using ASR models. An ASR model trained on raw audio can still be fine-tuned for featurized audio and does not need to be trained from scratch. These models can learn complex patterns and representations from data through training over a large speech data set such as TIMIT[15].

Next, we describe all ASR models that we evaluated and detail the procedure to mimic speech inference attacks. We tested two kinds of models based on the type of inferences the models made: phoneme and word. We summarize the ASR models in Table 2.

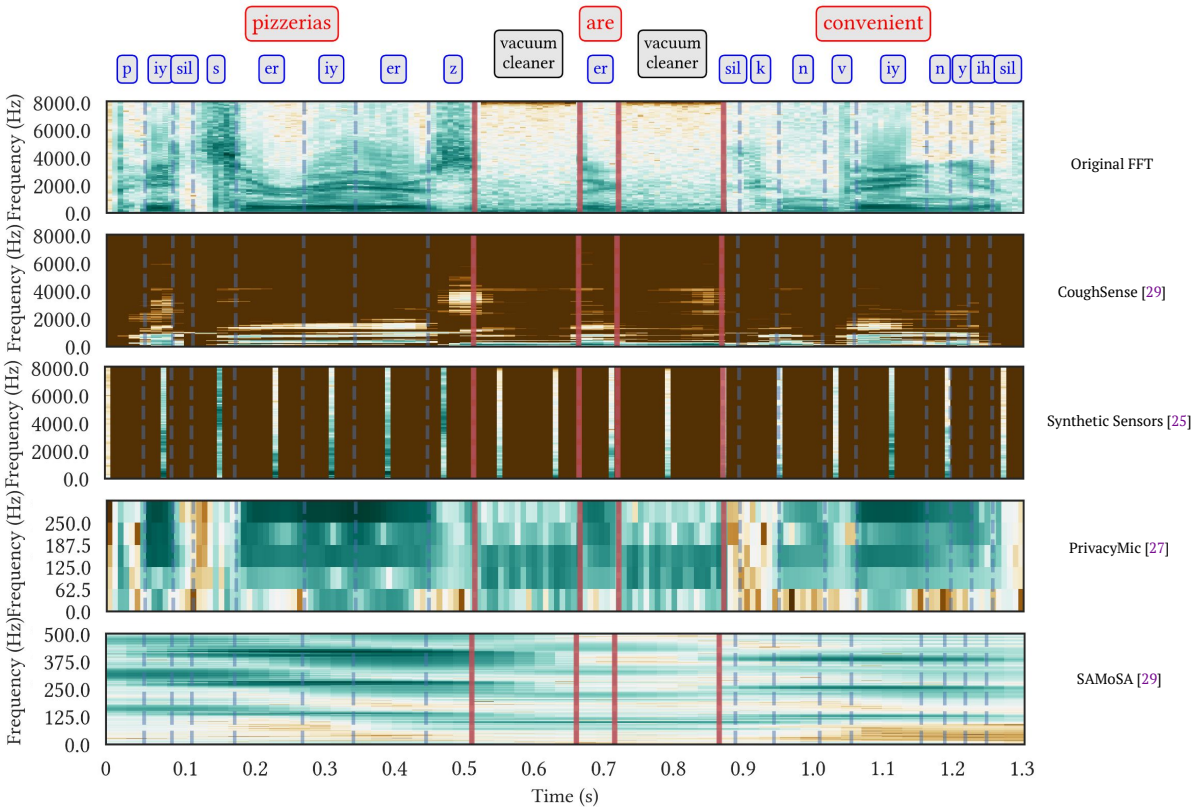


Fig. 5. FFT Spectrogram of several privacy filters proposed by prior work

Table 2. The list of evaluated ASR models against audio privacy filters

Attack Model	Inference Type	Pre-Training (Hours)	Fine-Tuning (Hours)	# Parameters	Metric
CRDNN	Phoneme	5 (labeled)	5 (labeled)	10M	PER
Wav2Vec Transducer	Phoneme	53.2k (unlabeled) + 960(labeled)	5 (labeled)	318M	PER
Whisper AI (Pretrained)	Word	680k (labeled)	0	769M	WER
Whisper AI	Word	680k (labeled)	5 (labeled)	769M	WER

4.1 Fine-Tuning Phoneme-based Speech Inference Models

Phoneme prediction models can be utilized to infer phonemes from featurized audio data. As opposed to word-level speech recognition, phoneme recognition offers the benefit of having considerably fewer prediction targets (e.g. 39 phonemes in CMU-Dict [10]), alleviating the concerns about the size of the vocabulary [6]. Moreover, we speculate that in the case only part of a word can be inferred, phoneme-based models might provide a chance for the human attacker to infer the complete word based on the context with only the predicted parts.

4.1.1 Convolution Recurrent Deep Neural Network (CRDNN). The CRDNN model combines Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Multi-Layer Perceptrons (MLPs). The CNN layers extract features from spectrograms of raw audio, while the RNN layers allow the network to find sequential

information for phoneme prediction. Connectionist Temporal Classification (CTC) [18] is integrated into the architecture, allowing the model to handle varying lengths of input and output sequences without requiring explicit alignment.

We adopted the CRDNN model to infer phonemes from filtered speech data based on implementation from SpeechBrain [37]. We convert the audio features obtained from different audio filtering techniques, through rescaling and cropping, into spectrogram-shaped representations with 40 features at each time window.

Table 3. Sample speech inference results and Phoneme Error Rate (PER) from the CRDNN model.

Original Segmented Phonemes	Privacy Filters	Predicted Segmented Phonemes	PER
pizzerias are convenient for quick lunch p-iy s-er-iy-er-z er-k-n v-iy-n-y-ih f-aa-r-k-w-ih k-l ah-n-ch	CoughSense [26]	p-iy-t-ih-ay iy ih-z-ih-k-m b-ih-n-y-ih f r-ay-k-w-ih m-ah-n-s	44.12%
	Synthetic Sensors [25]	s-t-ih-r-iy ih-z-k-ah-m p-iy-n f-aa-r-k-w-ih-l-ah-n s	50.00%
	PrivacyMic [20]	p-iy p er-r-iy-ih-z-ih k-m-b-ih-l-ah f-aa-r-t-r-ae n-ah-jh	52.94%
	SAMoSA [29]	dh-ah-p-r-aa p-er-d-ih-s p-l-ih-n t-ih-k ih-n-d-ih-s t-r-ey dh-ah-p-r-aa p-er	97.05%
december and january are nice months to spend in miami d-ih-s-eh-m b-er-ng-jh-y-ae-n y-uw-er ih-n ay-s-m-ah-n-th s-t-ih-s p-eh-n-ih-n m-ay-ae m-iy	CoughSense [26]	d-ih-s-ih-m-er-z eh-n-iy er m-aa-s m-ah-n s-d-ih-s p-ah n-ih-m aa-ih n-iy	45.45%
	Synthetic Sensors [25]	dh-ih-s-ih m b-er-ih-z eh m r-eh r-ih-n-ay-s m-ah-n t s p-r-ih-n-ih m-ay-ih-n m-iy	43.18%
	PrivacyMic [20]	dh-ah s-ih-ng-g-er-n jh-eh-n-er-l-iy ih-m-ay-s m-ih-n-s t-ih-s-p-ih-n-ih-ng l-ay-b-l-iy	45.45%
	SAMoSA [29]	dh-ih-s-p-aa-r k-ih-n-t-ih k-s-p-er d-ih-s-t-r-ey dh-ih-s-p-eh-r-ih k-ih-n-t-ih k-ih-n	77.72%
basketball-can-be-an-entertaining-sport b-ae-s-k-ih b-aa-l-k-ih-n b-iy ih-n eh-n-t-er ch-ey-n-ih-ng s-p-aa-r	CoughSense [26]	b-r s-t-ih b-r-k-ih-n b-iy-ih-n t-ih k-ey-m-ih n-s p-r-ay	32.43%
	Synthetic Sensors [25]	dh-ae-f-ih-l-aa k-ih-n m-ey-n er s t-ey n-ih-ng-z b-aa r	48.65%
	PrivacyMic [20]	dh-eh-s-t-ih b-ae-k-ih-n w-ah-n ih-n-t-er t-uw ih-n-iy s-t-r-ey s	54.05%
	SAMoSA [29]	dh-ih-s-p-aa-r t-ih s-p-l-ih-n t-r-iy-k ih-n-d-ih-s t r-iy-k-ih-n	72.97%

4.1.2 Transducer with Pretrained Wav2Vec 2.0. Transducer models, also known as RNN-T (Recurrent Neural Network Transducer) models, are a type of end-to-end ASR system that directly map input speech features to target text without requiring any explicit alignment between them [17]. These models consist of an encoder, a decoder, and a joint network, which together predict the output sequence in an autoregressive manner. Wav2vec 2.0 is a self-supervised pretraining method that learns powerful speech representations from raw audio waveforms by exploiting the temporal structure of the data [1]. Studies have shown that a pre-trained Wav2vec model leads to better performance than using handcrafted features, such as the Mel-frequency cepstral coefficients (MFCCs) or filter banks, as the ASR models can benefit from the rich and expressive features that wav2vec learns from large amounts of unlabeled audio data [13, 37, 41].

We used an implementation of the Transducer model from SpeechBrain [37] and fine-tuned the model starting from an existing checkpoint trained using the TIMIT dataset [15]. We used the Wav2Vec2-Large-LV60 to extract features from audio inputs, which contains 317M parameters and is pre-trained on 53.2k hours of unlabeled audio data and 960 hours of speech data [1]. While the resulting filtered audio from many audio privacy-focused featurization techniques transforms audio into the frequency domain, with the Wav2Vec 2.0 encoder, the Transducer model takes waveforms as inputs. To make the model compatible with the spectrogram-like shape resulting from different featurization approaches, we applied the Inverse Fast Fourier Transform (IFFT) to obtain a waveform representation from the spectrogram-shaped representations. All weights of the model are fine-tuned using the TIMIT dataset.

4.1.3 Phoneme Post-Processing. Directly interpreting the phoneme outputs might still be challenging for inexperienced adversaries. To enhance the speech inference practicality and to better understand the privacy implications of the tested audio filtering techniques, we perform the following post-processing on the phonemes output. Our first step is to segment the phoneme predictions into groups, in which each group of phonemes likely represents a word. We trained a bidirectional LSTM sequence tagging model to segment the phonemes using the ground truth TIMIT phonemes and words, which achieved 98.6% per-tag accuracy. In addition, we used a heuristic-based approach that breaks at 'sil' (silence) phonemes unless there are less than four consecutive predicted non-silence phonemes in prior, which is likely due to prediction error. After segmenting the entire phoneme sequence prediction, we used Pincelate [35], an open-source tool that performs phoneme-to-grapheme and grapheme-to-phoneme conversion, to spell the probable word for each segment of phonemes. Although the

Table 4. Sample speech inference results and Word Error Rate (WER) from the Whisper model

Original sentence	Privacy Filters	Whisper (Fine-tuned)	WER
pizzerias are convenient for quick lunch	CoughSense [26]	pitcheeers are convenient for a quick lunch	33.3%
	Synthetic Sensors [25]	his barriers continued to overlap	100%
	Privacy Mic [20]	peculiar is a conveyor for a quick lunch	83.3%
	SAMoSA [29]	people often go for in quick evening	83.3%
december and january are nice months to spend in miami	CoughSense [26]	decembers are nice mountains to spend in miami	40%
	Synthetic Sensors [25]	december and jan are make sure you save money to visit my website	90%
	Privacy Mic [20]	the figure here may attach to the spring and water	100%
	SAMoSA [29]	decide and jan are moving to the may	70%
basketball can be an entertaining sport	CoughSense [26]	basket bowl can be an immediate sport	50%
	Synthetic Sensors [25]	basketball can be found in video game	83.3%
	Privacy Mic [20]	bask be an enter	66.7%
	SAMoSA [29]	ballers are on an extreme sport	66.7%

spelling of the sentence is imperfect due to the phoneme inference errors, it still provides hints to infer the speech content. Table 3 shows the inferred segmented phonemes using the CRDNN model. In some cases, even after filtering, the inferred phonemes can sound very similar to the original sentences. For example, when using the CoughSense [26] filtering approach, the CRDNN model was still able to capture *p-iy-t-ih-ay iy ih-z*, which is very close to the pronunciation of the word *pizzerias*. When the PER value increases, transcribing the words is not as straightforward as one needs to spell the words and consider which phonemes might be incorrect predictions. The output *dh-ah s-ih-ng-g-er-n* for PrivacyMic [27], for instance, still sounds similar to the word *December*, but directly identifying the word without knowing the original sentence can be challenging.

4.2 Fine-Tuning Word-based Speech Inference Models

Whisper is a Transformer-based encoder-decoder model, more commonly known as a sequence-to-sequence model [36]. Unlike Wav2Vec, which was primarily trained on unlabelled data in an unsupervised manner, Whisper was trained on 680k hours of labeled speech data, such as LibriSpeech [32] using extensive supervision with 769M parameters. We used the pre-trained Whisper *medium* checkpoint and then fine-tuned the model to the different types of audio features obtained from the prior filtering techniques discussed in the previous section. Since the Whisper model expects log-Mel spectrogram as input, we convert the audio features obtained from different audio filtering techniques to log-Mel spectrograms. We then use this information to fine-tune the model. During the fine-tuning step, Whisper's parameters are updated to match the specific characteristics of the target word prediction, such as its phonetic spectral properties.

Table 4 shows examples of sentence predictions from Whisper pre-trained and fine-tuned ML models when we apply different filtering techniques. In certain cases, the predicted sentences are similar to the original sentences. For example, in certain instances, even after applying the CoughSense [26] filtering approach, the fine-tuned whisper model successfully predicted all the words except for "*pitcheeers*" in the example "pizzerias are convenient for quick lunch." Furthermore, it is observed that as the WER value increases, the distinction between WER values becomes less clear in terms of what information they may reveal. For instance, although an example sentence prediction from PrivacyMic has 90% WER, the prediction from Synthetic Sensors with the same WER still reveals some of the original speech information (such as words like "december" and "jan"). In addition, we found WER may appear high for short sentences, due to the number of words normalizing it, and thus predicting only a few incorrect words will be enough to raise WER to a high value.

4.3 Need for PER and WER contextualization

Notably, as the above results show PER and WER values by themselves are only part of the story in terms of understanding the potential privacy concerns with the parts of the original speech that may still be reconstructed using ASR approaches. In addition, all the words in a sentence are not the same in terms of what they reveal

about the conversation and different sentences with similar WER/PER values may lead to less (or more) privacy concerns. Finally, the data and examples for different featurization approaches mentioned in this section are merely illustrative to show what is possible by re-tuning some of the ASR models. In Section 6.2 we provide a detailed evaluation of Kirigami as compared with various prior approaches on a larger corpus of speech data in terms of average PER and WER values. Furthermore, to contextualize different PER and WER ranges in terms of what they can still reveal, we performed a separate user study, the results of which are reported in Section 6.4.

5 KIRIGAMI: LIGHTWEIGHT SPEECH FILTER

As shown in the previous section, prior approaches on preserving user privacy are susceptible to inferring speech with the latest state-of-art fine-tuned ASR models. A key reason for this is that these approaches focused on degrading data or utilizing feature-reduction strategies to filter potential speech segments. However, modern ASR models such as Whisper [36] are trained on a broad spectrum of acoustic features and linguistic contexts that can take advantage of any residual speech information, such as phonemes, making them less susceptible to conventional privacy-preserving techniques. More importantly, as the development and optimization of ASR models progress in the future, their reliance on any residual speech segments to enhance ASR performance increases, highlighting the necessity for new strategies in preserving privacy. Consequently, our approach focuses on the detection and removal of data segments containing speech-related information, including phonemes. This ensures a more robust mechanism to safeguard user privacy within the evolving realm of ASR technology. Our design of Kirigami is based on a set of key insights. First, the detection of speech information (phonemes) can be modeled as a binary classification task, for which shallow machine learning models may suffice in terms of reasonable accuracy. Second, these shallow ML models can be deployed on a wide variety of hardware as they are memory and computationally efficient. Third, the Kirigami filter can promptly discard detected speech segments at the edge to safeguard speech privacy, allowing full-featured FFT data to pass through when non-speech segments are detected, thereby optimizing utility performance.

5.1 Machine Learning-based Kirigami Speech Filter

Our proposed solution of a speech filter on the edge involves constructing a lightweight yet efficient real-time speech detector. The overall objective of the speech detector is to classify each time frame of the Short-Time Fourier-transformed (STFT) audio data as either speech or non-speech. Formally, let $X \in \mathbb{R}^d$ be a d-dimensional vector representing a time frame of STFT data. For example, $d = 128$ when the window size of the STFT is 256. The task of the speech detector is to learn a mapping $f : \mathbb{R}^d \rightarrow 0, 1$, where $f(X) = 1$ indicates speech and $f(X) = 0$ indicates non-speech. Once a time frame is identified as likely speech (i.e., $f(X) = 1$), that particular time frame is discarded.

In Kirigami, we use a Logistic Regression (LR) model for real-time speech detection. Logistic Regression is a well-established shallow ML model typically used for binary classification tasks and is also resource-efficient. To build an LR model, we first normalize the STFT features using the L-1 norm across all frequency components for each time frame. The normalization step ensures that the influences of volume variation from the audio signal are reduced.

Thus we formally represent the Logistic Regression model used for binary speech detection as follows:

$$g(X) = \sigma(W \cdot \frac{X}{||X||_1} + \beta_0) \tag{1}$$

, where $||X||_1 = \sum_{i=1}^{d} |x_i|$ is the L-1 norm of X, W and β_0 are the weight coefficients and intercept learned from the training data, and σ is the logit function:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{2}$$

The decision $f(X)$ of whether a time frame should be removed is based on comparing the model prediction against a threshold value τ, which can be represented as:

$$f(X) = \begin{cases} 1 & \text{if } g(X) \geq \tau \\ 0 & \text{if } g(X) < \tau \end{cases} \tag{3}$$

In our training process, the LR model was developed using the TIMIT [15] dataset for speech data and the ESC50 [34] environmental sound dataset for non-speech data. We opted for the TIMIT dataset for speech data due to its inclusion of multiple hours of phonetically transcribed sentences, enabling in-depth analysis and modeling of speech sounds. Additionally, we opted for the ESC50 dataset for non-speech data, aligning with our objective of recognizing events and activities. The sound samples from ESC50 provide valuable training data for the LR model to effectively distinguish and preserve sounds associated with various activities. To enhance the diversity of our dataset, we created an additional speech dataset where we overlay sounds from ESC50 on top of the TIMIT speech audio. This augmentation aims to enrich the training data, enabling the LR model to better generalize and perform effectively across a range of real-world scenarios. We apply STFT to the audio samples from these sources to transform them into the frequency domain (FFTs). Subsequently, we label each time frame as positive (i.e., speech) or negative (i.e., non-speech) depending on the source. We balanced the dataset to have an equal number of speech and non-speech samples. In total, our dataset comprises 20000 samples, which are randomly split into three subsets: 80% for training, 10% for validation, and 10% for testing. Through supervised training, the LR model learns to classify each time frame as speech or non-speech, thereby removing the time frames that are likely speech. Overall, our Kirigami LR model (using $\tau = 0.5$) achieved a speech recognition accuracy of 76.44%, indicating the effectiveness of our method in accurately identifying and classifying speech segments. It's important to note that we are not aiming for perfect classification accuracy, and our goal was to make the model configurable to balance privacy or utility requirements, depending on the use case. This adaptability allows for a nuanced and tailored approach, where the balance between accuracy and the desired outcome can be fine-tuned to align with the overarching goals of the application or system. We elaborate further in Section 5.2 on how the Kirigami LR model can offer sufficient privacy protection with appropriate threshold values while preserving adequate utility value.

5.2 Configuring Privacy vs. Utility Tradeoffs

Figure 6 provides an illustration of the trade-off between privacy and utility as we configure the Kirigami filter to have different values for τ. In the figure, the first part contains pure speech ($t = 0s$ to $t = 1.86s$), speech data overlaid with a vacuum cleaner sound ($t = 1.86s$ to $t = 3.72s$), and a vacuum cleaner sound ($t = 3.72s$ to $t = 5.6s$).

The value of the threshold τ, configurable to be between 0 and 1, plays a crucial role in determining the model's inclination towards either preserving privacy or maintaining utility. A value closer to 0.5 leans toward balancing both. Given that the Kirigami filter is an ML model, there are instances where it is incorrect, which leads to either some speech data being leaked or some audio event data being filtered. For example, for the LR0.5 configuration, some frames with the word "quick" are mistakenly classified as non-speech (a false-negative). As the threshold changes from 0.5 to 0.1, the model becomes more conservative and prioritizes privacy protection. For example, for the LR0.1 configuration, all the segments with the word "quick" are now detected correctly as speech, but towards the end, numerous segments with the vacuum cleaner sound alone are incorrectly filtered out as speech (false positives), which can affect the utility of activity detection. The optimal threshold depends on the specific use case and the application requirement. In situations where privacy is crucial, such as if the sensor is installed in a private office, a lower threshold would be more suitable. Conversely, in less sensitive contexts (e.g., shared spaces) or where the accuracy of activity detection is more important (HAR for fall detection scenario), a threshold closer

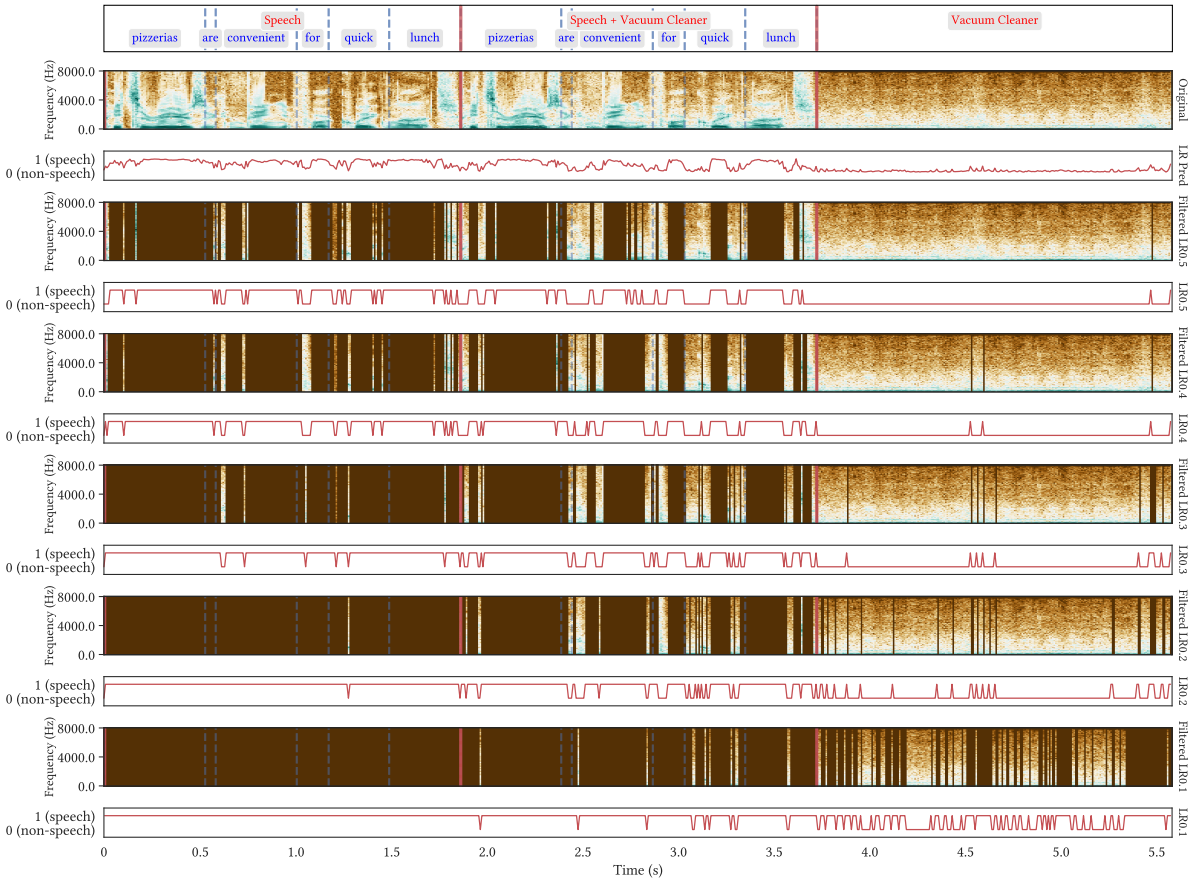


Fig. 6. Illustration of Kirigami's Logistic Regression model for phoneme prediction. The LR Pred line graph shows the predictions from 0 to 1, while LR0.5 to LR0.1 shows predictions at threshold values indicating speech (1) or non-speech (0). Spectrograms showcase original and filtered audio, demonstrating a balance between privacy (speech filtering) and utility (activity recognition).

to 0.5 may be more appropriate. We further quantify the impact of different thresholds on privacy vs utility and discuss its implications as compared to various prior approaches in Section 6.3 and 6.4.

5.3 Kirigami Speech Filter on the Edge

A key goal in developing the Kirigami filter was to ensure its feasibility of deployment on edge, which typically implies operating in resource-constrained environments. For instance, popular ARM Cortex M class microcontrollers commonly found in IoT devices have around 128KB RAM 100-150MHz CPUs, and thus cannot run deep learning-based ASR models such as Whisper to filter speech. With all its configurable threshold parameters, this meant that the Kirigami filter needed to be implemented in environments with frugal memory resources and limited computing capabilities.

We quantized the Kirigami LR model to reduce its memory footprint, ensuring more efficient storage and processing on resource-constrained devices. This involved converting the model's floating-point parameters to integers, a process that conserves memory and contributes to improved computational efficiency. Overall, our

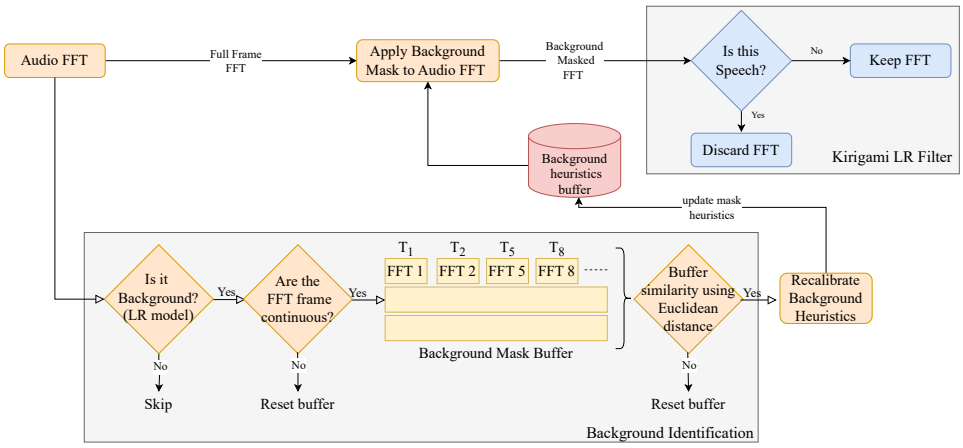


Fig. 7. Flowchart depicts the adaptive background masking process in conjunction with Kirigami's speech filter. The process involves background detection, buffer comparison, heuristic calculation, and the generation of a background mask to filter out background frequencies. The resulting background-filtered FFT enhances Kirigami's speech filter for improved accuracy by eliminating background noise.

Kirigami approach requires the storage of 129 weight values, including the intercept, and the computation of one normalization, one dot product, and one logit function. We implemented the Kirigami LR model on the popular edge microcontroller ARM Cortex-M4F with 256 KB RAM and 1 MB flash and measured the memory consumption and latency. Our measured memory footprint of the quantized Kirigami model coefficients was 518 bytes (< 1 KB) and a total of 2.1 KB (< 3 KB) for the entire Kirigami filter, including the model intermediate weight calculations. The end-to-end latency for prediction of an FFT sample is approximately 0.71 ms, demonstrating that the Kirigami filter is not only resource-efficient in terms of memory consumption but also exhibits low latency, making it well-suited for deployment in real-time applications on edge devices with limited computational resources. We implemented the Kirigami LR model in both C and Python to ensure comprehensive compatibility of the Kirigami filter across different device environments and run efficiently on devices with limited computational power and memory. We further evaluate the real-world accuracy performance of our edge Kirigami filter in Section 6.6.

5.4 Adapting the Kirigami Speech Filter for Real-World Environments

A key design goal of the Kirigami filter is to robustly filter out speech in real-world environments. Our initial hypothesis to achieve this was to train the Kirigami filter on a custom dataset where activities of interest are overlaid with speech events. We formed this dataset by augmenting environmental sounds from ESC50 dataset of activities and ambient sound with the TIMIT speech audio, using this dataset to train the Kirigami model. This approach allowed us to simulate real-world conditions where people speak with other activities and events happening in the background. The Kirigami speech filter, while effective in controlled environments, faced several challenges in these real-world situations, primarily due to the presence of background noise, diverse acoustic landscapes, and variability in ambient noise levels.

Thus, we aimed to identify background or ambient sounds in the environment and filter those out before identifying and filtering speech events. However, real-world environments exhibit a dynamic spectrum of ambient noise, with levels that fluctuate based on factors such as location, time of day, and environmental conditions. For example, workshop environments may feature machinery traffic sounds, while quieter office spaces may still

have variable noise from air conditioners, HVAC, or occasional people chatting in the surroundings. Moreover, background noises, which may be constant or intermittent, span a wide spectrum of frequencies and intensities.

To overcome these challenges, we present an adaptive background masking process combined with Kirigami's speech filter. We continuously collect background noise profiles from the environment, estimate a mask for these background noises, and apply this mask to filter out the noise. As shown in figure 7, our approach consists of the following steps: (1) background identification, (2) creating a background mask buffer, and (3) background mask generation and filtering. Once the noise is filtered, the data is sent to our Kirigami LR model for speech filtering. The background identification step uses a Logistic Regression model to predict whether the input FFT represents background noise versus foreground speech or activity of interest. This model is trained on datasets containing a mixture of background noises of typical environments from Microsoft Scalable Noisy Speech Dataset (MS-SNSD) [38] and foreground activities and speech from TIMIT [15] and ESC-50 [34]. We attempted to further increase the real-world fidelity of noise mixtures by overlaying the foreground speech and activities with background noise at various signal-to-noise ratios and various pitches of background noises. Second, these predicted FFT data are added to the background noise mask buffer, which maintains multiple buffers of continuous background FFT data. This buffer imposes conditions on the temporal continuity of background FFT samples, ensuring that the FFT frames within the buffer are contiguous. Once multiple of these buffers are filled up, the similarity across different buffers is gauged using the Euclidean distance metric If the buffers are similar, the process generates a background mask. This process ensures the reliability and accuracy of the captured background profile, enabling adaptation to diverse environmental settings. The background mask generation process relies on a method called spectral gating [22]. This technique involves estimating a background threshold (or gate) for each frequency band within the collected background profile, calculated using the mean and standard deviation over frequency. This threshold is then used to compute a mask, which gates noise below the frequency-varying threshold. During the background masking phase, we initiate the process by establishing a gain control for each frequency band. If a frequency surpasses the previously determined threshold, the gain is set to 0 dB; otherwise, the gain is reduced (*e.g.*, to -18 dB) to mitigate background noise. Following this, we use frequency smoothing to ensure that individual frequencies are neither excessively suppressed nor boosted in isolation. We then direct the background-masked FFT to the Kirigami speech filter, which is now potentially less susceptible to the influence of background noise. By incorporating an adaptive algorithm that responds to fluctuations in ambient noise, we enhance the Kirigami filter's versatility in handling variable acoustic environments, ensuring reliable speech recognition performance across diverse real-world scenarios. We evaluate the robustness of our approach in the real world in Section 6.6.

6 EVALUATION

This section evaluates the effectiveness of state-of-the-art speech recognition systems in recovering speech text from the prior privacy-focused featurization approaches. In addition, we evaluate Kirigami's ability to identify phonemes from audio data. Overall, our evaluation aims to answer the following questions:

- RQ1: How accurately do modern ASR-based systems identify speech contents from audio featurized using prior approaches?
- RQ2: How robust is Kirigami 's filter to ASR-based attacks, and how does Kirigami's filtering approach affect the utility?
- RQ3: How accurately does Kirigami's filter perform in real-world environments?

6.1 Evaluation Setup

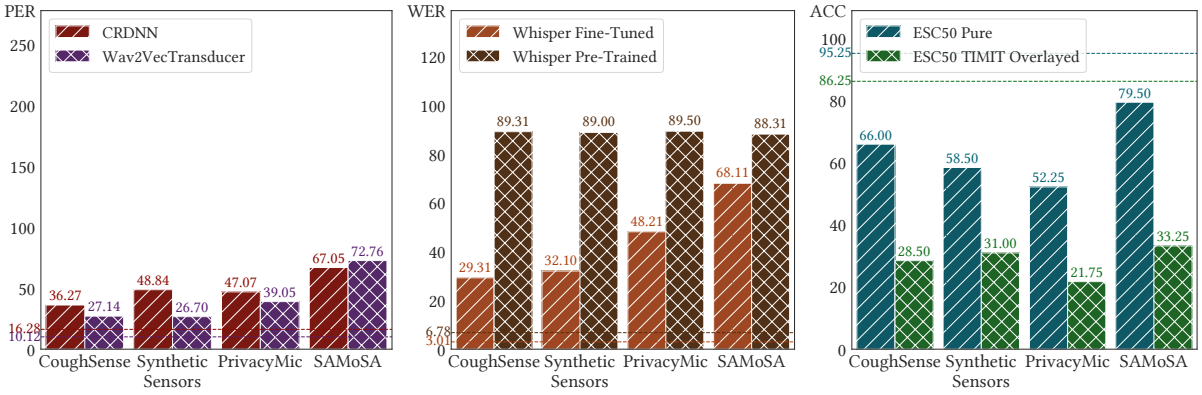
Dataset: We utilize the TIMIT [15] dataset to evaluate the feasibility of inferring speech from featurized data, fine-tuning the ASR-based models, and building the Kirigami filter. The TIMIT dataset contains a total of 5 hours

of English speech with 4,620 phonetically transcribed sentences, with approximately ten sentences per speaker. Each sentence is segmented into phonetic units, such as phonemes and words, allowing for detailed analysis and modeling of speech sounds. To evaluate the utility of Kirigami filter and the prior filtering approaches, we use the ESC-50 [34] dataset. The dataset contains 2000 environmental sound recordings from 50 classes involving various sound types, including animal sounds, natural sounds, human non-speech sounds, *etc.* To match the scope and difficulty of the application scenarios in prior audio privacy filtering approaches, we selected ten classes: toilet flush, sneezing, clapping, breathing, coughing, footsteps, laughing, brushing teeth, snoring, drinking, door knock, washing machine, vacuum cleaner, clock alarm, and clock tick. Finally, we also created an overlay of the TIMIT dataset with the ESC-50 dataset to evaluate privacy and utility performance in a noisy environment. For speech inference evaluation, the overlaid data is produced by overlaying a random sound file from ESC-50 on top of each speech audio file from TIMIT. The resulting audio file contains the same speech content and length as the original. Similarly, for utility evaluation, we overlaid a random speech audio file from TIMIT on each sound file from ESC-50. We match the loudness of two different audio files based on the loudness level in decibels relative to full scale (dBFS).

Speech Inference Evaluation: To examine the extent to which modern ASR models can infer speech content information from prior audio privacy filter approaches, we implemented each privacy filter approach and evaluated the speech inference performance. The approaches that we included in our evaluation are CoughSense [26], Synthetic Sensors [25], PrivacyMic [27], and SAMoSA [29] as all these works indicate privacy as a primary factor in their filter design process. We applied each of these privacy filter approaches to obtain a dataset of filtered audio samples. We used four different configurations of ASR models: CRDNN, Wav2VecTransducer, Whisper Pre-Trained, and Whisper Fine-Tuned. The training set of filtered audio samples is used to fine-tune the model weights, which facilitates the ASR model to adapt to filtered audio samples and learn suitable new feature extraction and prediction mechanisms. The models are trained to start from existing checkpoints for optimal speech inference performance.

Privacy Performance Measures: The performance of speech inference is measured in Phoneme Error Rate (PER) and Word Error Rate (WER). PER and WER are defined as the number of insertions, deletions, and substitutions normalized by the length of the target sentence. PER measures the number of incorrect phoneme predictions produced by the ASR model, while WER measures the rate at which words are predicted incorrectly. The evaluation of speech inference measured in PER and WER is conducted on both TIMIT as the pure speech dataset and overlaid dataset, although we adopt the PER and WER on pure speech data as the primary measure of privacy protection.

Utility Performance Measures: Utility, often emerging as an opposing goal to privacy, also needs to be assessed to understand the effectiveness of a privacy filter. An effective filter ideally shall achieve high PER and WER while having minimal loss in the utility performance compared to non-filtered audio. We adopted the Audio Spectrogram Transformer (AST) [16], a state-of-the-art audio classifier and one of the best performers on the ESC50 dataset, to evaluate the accuracy of inferences as the utility performance. We used a 5-fold cross-validation standard to the ESC50 dataset and calculated the classification accuracy of the 10 classes selected from the ESC50 dataset on, both, the pure environmental sound and the sound overlaid with speech. Unlike speech inference where the privacy leakage on pure speech is the primary concern, the classification accuracy on both pure and noisy data is crucial for consistent performance across different environmental conditions. Taking all these measures together allows us to assess the trade-off between privacy protection and utility preservation for each filtering approach.



(a) Phoneme Error Rate (PER) (b) Word Error Rate (WER) (c) Classification Accuracy

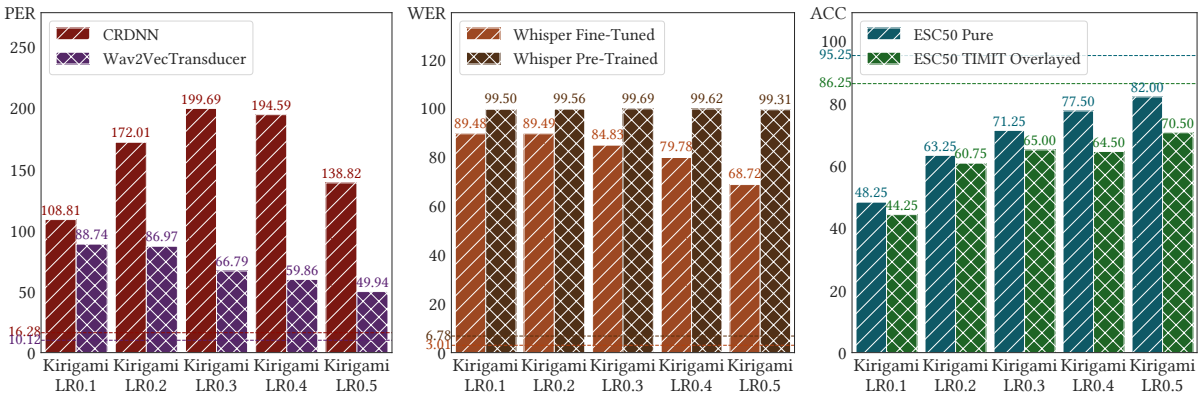
Fig. 8. Results of (a) phoneme-based speech inference, (b) word-based speech inference, and (c) activity classification accuracy on prior filtering approaches.

6.2 RQ1: Feasibility of Speech Inference from Prior Speech Filtering Approaches

We fine-tune the ASR models by applying each audio featurization approach to the training set of the TIMIT dataset. The ASR models learn to infer speech from these featurized audio samples through fine-tuning. Finally, the performance of speech inference from all ASR models is evaluated using the PER and WER metrics on the pure speech Timit dataset and Timit speech data overlaid ESC-50 activity data.

Phoneme-based Speech Inference: Fig. 8(a) summarized the experiment results of speech inference on featurized audio data using four prior approaches to audio speech filtering using the CRDNN and Wav2VecTransducer models. Overall, these results demonstrated a concerning level of privacy risks in audio privacy filtering techniques. CoughSense [26], Synthetic Sensors [25], and PrivacyMic [27] showed PER of 27.14%, 39.05%, and 26.70% respectively on pure speech sounds, proving the feasibility in inferring phonemes from the filtered speech data. Wav2VecTransducer outperforms CRDNN in inferring phonemes on these three approaches except on SAMoSA. SAMoSA [29], with simple downsampling and a large FFT window size approach, exhibits adequate protection on speech. We conjectured that this protection might be due to the length of FFT windows measured in time (600ms) far exceeding the time to speak a phoneme in most cases. To help assess the audio filtering effectiveness in comparison to complete audio data, we included our baseline approaches using FFT data from 256/128 windows and step sizes without any filtering, achieved PER of 16.28% and 10.12% using CRDNN and Wav2VecTransducer models, respectively. In Figure 8 (a) and (b), this baseline is shown as dashed lines.

Word-based Speech Inference: We also compare the Word Error Rates (WER) of the prior audio filtering techniques using fine-tuned and pre-trained Whisper models to assess the efficacy of the word-based speech inference models. Figure 8(b) shows the WER of fine-tuned whisper for prior filter approaches. CoughSense, Synthetic Sensors, PrivacyMic, and SAMoSA, showed WER of 29.31%, 32.10%, 48.21%, and 68.11%, respectively, on pure speech sounds showing that fine-tuned whisper models can recognize speech content even after the data is filtered by the prior approaches. In addition, we also see that using an off-the-shelf pre-trained Whisper model has higher WER, which means the prior filter approaches are still resilient to the pre-trained Whisper model. The weakest filter among the four is CoughSense [26] as it has the lower WER scores (29.31%), meaning the inference obtained from the fine-tuned model provides enough information about the original speech content. As



(a) Phoneme Error Rate (PER) (b) Word Error Rate (WER) (c) Classification Accuracy

Fig. 9. Results of (a) phoneme-based speech inference, (b) word-based speech inference, and (c) activity classification accuracy on Kirigami filtering approaches.

a point of comparison, our baseline approaches using FFT data with 256/128 windows and step sizes without any filtering yielded WER of 3.01% and 6.78% with fine-tuned and pre-trained Whisper models, respectively. Overall, these results demonstrate the potential of fine-tuning the ASR model to effectively infer speech content from filtered audio, highlighting its capability to overcome the prior speech filter approaches and provide accurate word-based speech recognition.

6.2.1 Utility Impact: Figure 8(c) shows the audio classification accuracy on 10 activity classes on the ESC50 dataset [34] over the four prior approaches. The baseline configuration (no filter) achieved 95.25% and 86.25%. Out of the four prior approaches, the best performer is SAMoSA [29], which achieved 79.50% accuracy in classifying pure activity sounds. The classification performance for the other three approaches is significantly lower than the baseline configuration. Notably, all four approaches showed significant performance drops on overlaid sounds. We hypothesized that this performance drop might be caused by the always-on manner of these filtering approaches, which degrades the expressivity of the audio data and makes activity and speech sound less distinguishable when overlayed. The drop is especially pronounced for SAMOSA (a drop of 46.25%). We posit this drop to SAMOSA's very low default sampling rate (1 kHz). This approach works well when the signal is clean, but the performance plummets significantly when the speech and ambient sounds are overlaid.

6.3 RQ2 : Performance of Kirigami filters

Fig. 9 summarizes the performance of the Kirigami filter with different configurations of threshold values. Overall, Kirigami filters showed superior protection for speech privacy compared to the prior approaches, especially for configurations that lean towards privacy, such as LR0.1 and LR0.2.

Phoneme-based Speech Inference: As shown in Fig 9 (a), the CRDNN model produces almost complete noise, with PER values above 100%, for any of the 5 Kirigami filter configurations. For the Wav2VecTransducer model, as the threshold value moves from 0.5 to 0.1, the difficulty of inferring phonemes, as measured by the PER produced by the Wav2VecTransducer model, increased as expected.

Word-based Speech Inference: We also see similar trends in word-based models (Fig. 9 (b)). When the Kirigami filters are applied to the Whisper pre-trained models, we see that they have a higher WER score of more than

90%. For fine-tuned models, LR0.3, LR0.2, and LR0.1 all achieved above 80% WER. In addition, we see that as we change the threshold configuration of Kirigami from being privacy-preserving (threshold = 0.1) to providing higher utility (threshold = 0.5), the WER values decrease from 89.48% to 68.72% for fine-tuned whisper model. Even the lowest WER 68.72%, which is produced by LR0.5 that leans more towards utility, is already higher than the WER for SAMoSA [29], the best-performing privacy filtering technique out of the four prior filters.

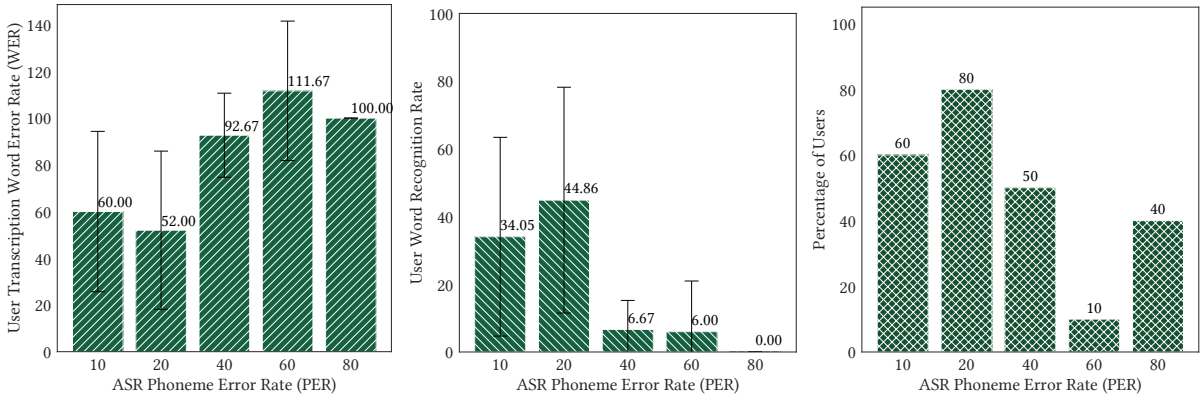
Utility Impact: Out of the five configurations, LR0.5 achieved the best classification accuracy at 82.00% for pure activity sounds and 70.50% for overlaid sounds, which also outperforms all 4 prior filtering approaches that we evaluated. As the Kirigami filter is configured to be more privacy-sensitive, the classification accuracy slightly drops. Even at LR0.2, the accuracy for both pure and overlaid sounds is still above 60%. Another notable difference from prior approaches is that for all Kirigami filters, the negative impacts from overlaid sound are very moderate, at most 11.50% for LR0.5. This advantage of Kirigami, as we hypothesized, is because Kirigami keeps the complete FFT values at the pauses of speech in the overlaid sound, which provides adequate information for the activity recognition. This highlights another advantage of Kirigami filters as to not only protect speech privacy but also maintain utility value even when activities are performed when speech is present.

6.4 PER v.s WER Contextualization

While Phoneme Error Rate (PER) and Word Error Rate (WER) are widely used in speech recognition literature, it is difficult to contextualize their privacy implications. For instance, one could ask at what level of PER or WER is an audio featurization technique safe or risky. In addition, it remains a question of what information can be inferred at different PER and WER values. To understand the practical implications of speech inference, we conducted a IRB-approved user study to contextualize how much information can users decipher from the inferred phonemes and words.

Questionnaire Design: We randomly selected ten sentences from the TIMIT dataset [15] that independently convey a complete meaning. For instance, the sentence *pizzerias are convenient for quick lunch* conveys a statement about pizzerias and lunch. Each sentence was subjected to speech inference predictions through various privacy filters, and the PER and WER were measured for each. For each one of the ten selected sentences, we randomly picked five different predictions that fall into five ranges of PER or WER values. Using these sentences, we created a pool of 50 scenarios, half of which are phoneme-based speech inferences, and the other half are from word-based models. In each scenario, we ask the participants five questions, including transcribing the sentence, identifying words from the original audio, choosing the most likely speech topic, choosing the most likely speech content, and rating the similarity of the prediction to the original sentence. In phoneme scenarios, we presented the segmented phonemes (e.g., *p-iy-ch-er-r-iy-z ih-k-n-v-iy-n y-ih f-aa-r-ah k-w-ih-l-aa ch*), spelling prediction (e.g., *peceruries enchant ye fara Quilla ch*), as well as a reference phoneme pronunciation table. For word model predictions, we only show the sentence prediction (e.g., *combine play them grams a large bowl*).

Study Procedure: In total, we recruited 10 participants (seven females and three males) from the university with an average age of 24.7, ranging between 22 and 28 years. Out of the ten participants, two participants self-identified themselves as having linguistic backgrounds. Before the study began, we introduced participants to phonemes and speech inferences. Then, we went over two example scenarios, one from a phoneme model and the other from a word model. We guided participants through the process of answering five questions for the phoneme scenario and five questions for the word scenario. We demonstrated how to transcribe the words and infer speech content based on the phonemes and word predictions that contain errors, as well as addressed any confusion that participants may have had regarding our study. Once the study began, each participant was randomly assigned ten scenarios, one for each sentence. The ten scenarios that a participant answers all have distinct PER and WER ranges.



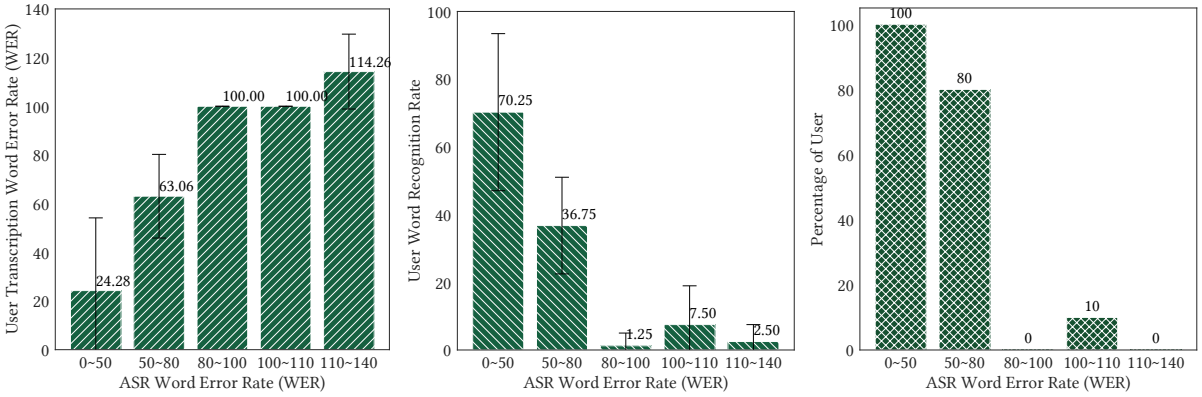
(a) User Transcription WER (b) User Word Recognition Rate (c) Topic Inferred User Percentage

Fig. 10. Results of user transcribing sentence (a), recognizing words (b), and inferring speech topic (c) at various PER from ASR models

Study Results: We summarized the results of the user study in Fig. 10 and 11. Figure 10 suggests a steady increase in the difficulty of recognizing words from phonemes as PER increases. Participants, on average, recognize around 40% of the words when phoneme prediction PER is around 10 ∼ 20%. The majority of participants (50 ∼ 80%) are able to infer the topic of the sentence from its phoneme prediction until 60% PER, especially when the user has some prior knowledge of the context or has relevant options shown to them. Although on 80% PER, 40% of participants selected the correct topic, we believe the participants answered the topics correctly by chance after we manually examined the questions and phoneme predictions that these participants received. Therefore, we consider 60% as a suggested threshold of PER, above which minimal information can be obtained from the model inference. For WER, the difficulty in transcribing and recogizing words increases as WER increases. Figure 11 shows sharp turning point at 80∼100%, after which it becomes very challenging for participants to recognize any words or topics. Therefore, we suggest an 80% WER threshold as the point at which the model inference can provide only limited information. However, it is recommended to consider both PER and WER together for better privacy assurance. This study provides useful insights into the performance evaluation of ASR systems and can guide future research in this field.

6.5 Comparison of Kirigami and Prior Speech Filtering Approaches

Figures 12 (a) and (b) show the comparison of the privacy and utility tradeoffs of prior and Kirigami speech-filtering approaches based on two benchmark datasets, pure Timit speech data for speech inference (to assess privacy risks), ESC50 activity recognition dataset (to assess utility benefits for activity recognition) and Timit speech overlaid on ESC50 dataset (to assess utility benefits for activity recognition in noisy data) for phoneme and word based ASR models. Using the scatter plot we can assess the effectiveness of the prior approaches and the Kirigami filters in preserving privacy while preserving the utility for activity recognition. In the scatter plot, the x-axis represents the level of privacy achieved by each approach, with greater distances from the base indicating higher privacy protection. The y-axis represents the level of utility achieved, indicating the effectiveness or performance of the activity recognition tasks. The privacy metrics PER and WER values for the filtering approaches, picking the lowest PER values and lowest WER values (most privacy-invasive) among the ASR models. Based on user study results, a threshold of 60% for PER and 80% for WER is established as the point



(a) User Transcription WER (b) User Word Recognition Rate (c) Topic Inferred User Percentage

Fig. 11. Results of user transcribing sentence (a), recognizing words (b), and inferring speech topic (c) at various WER from ASR models

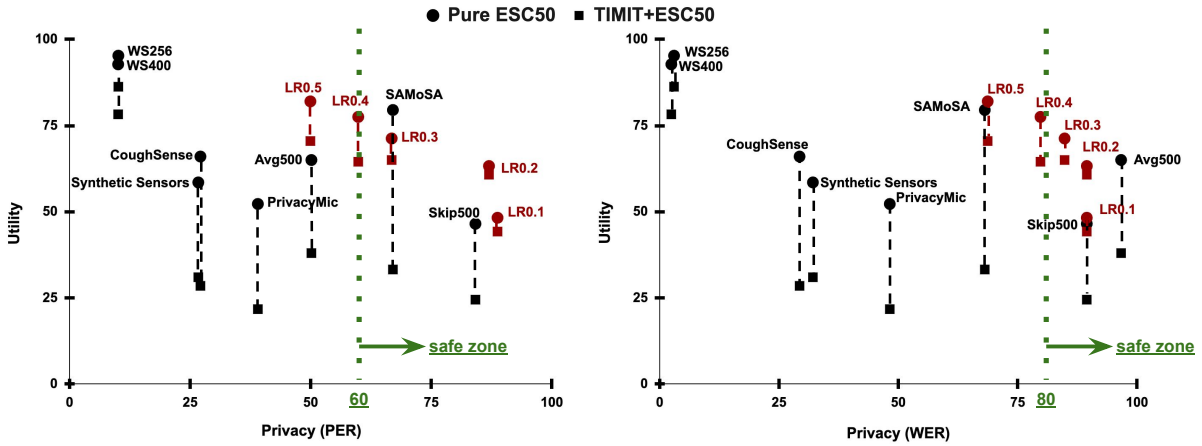


Fig. 12. Scatter Plot of Privacy and Utility Trade-Off of Different Audio Featurization Techniques. The vertical dashed lines represent the drop in accuracy for the utility measure in the presence of simultaneous speech and ambient sounds (overlaid sounds). Informed by our user study, we consider regions more than 60% PER and 80%WER as safe zones as little information from speech can be inferred. The ideal region is the top-right corner as it maximizes error in reproducing the spoken content and accuracy for the end goal task. For both plots, several Kirigami configurations are near that corner. In (a), SAMoSA [29] is close to the corner too, but the drop off in utility due to the presence of overlaid sounds is substantial. In comparison, Kirigami filters are more immune to noisy environments.

at which the filtering approach is deemed safe. The utility metric is picked based on this high accuracy achieved after the filtering technique using the ESC50 dataset. Ideally, the desired positioning of the filtering approaches on the scatter plot is in the top right quadrant. This indicates achieving the highest utility while simultaneously providing the highest privacy guarantees.

Figure 12(a) shows the most privacy-preserving filter for phoneme-based ASR models is the Kirigami's LR 0.2, while the least privacy-preserving filter apart from the baseline (WS256, WS400) is CoughSense [26] and Synthetic sensors [25]. While other approaches, such as SAMoSA [29], have high utility accuracy(76%) for pure ESC50 dataset, and their approach is in the PER safe zone, their utility accuracy (26%) drops for activity recognition when noise data is present (Timit speech overlay on ESC50). Based on this, the most ideal approach is Kirigami's LR 0.2 primarily due to higher PER numbers and better utility accuracy for both pure ESC 50 and Timit speech overlay on ESC50 datasets. Figure 12(b) shows the most privacy-preserving filter word-based ASR models have both Kirigami's LR 0.2 and Kirigami's LR 0.1, least privacy-preserving is CoughSense [26]. We also see that most of the prior speech-filtering approaches are in the unsafe zone, including SAMoSA and PrivacyMic, indicating that these approaches are ineffective in preserving privacy. Considering both privacy and utility aspects, Kirigami's LR 0.2 filter emerges as the most suitable choice due to its higher PER numbers and better utility accuracy for both the pure ESC50 and Timit speech overlay on ESC50 datasets. It strikes a balance between privacy preservation and utility enhancement. Our Kirigami's LR 0.2 filter offers a compelling solution, providing a high level of privacy preservation while maintaining satisfactory utility accuracy. However, as we delve into the extensive real-world study, the story takes an unexpected turn. Contrary to our initial expectations, the effectiveness of the Kirigami's LR 0.2 filter, while effective in controlled environments, was significantly impacted by the presence of real dynamic background noise, diverse acoustic landscapes, and fluctuations in ambient noise levels. To overcome this, we present an adaptive background masking process combined with Kirigami's speech filter as mentioned in Section 5.4 and evaluate Kirigami's effectiveness in discarding speech.

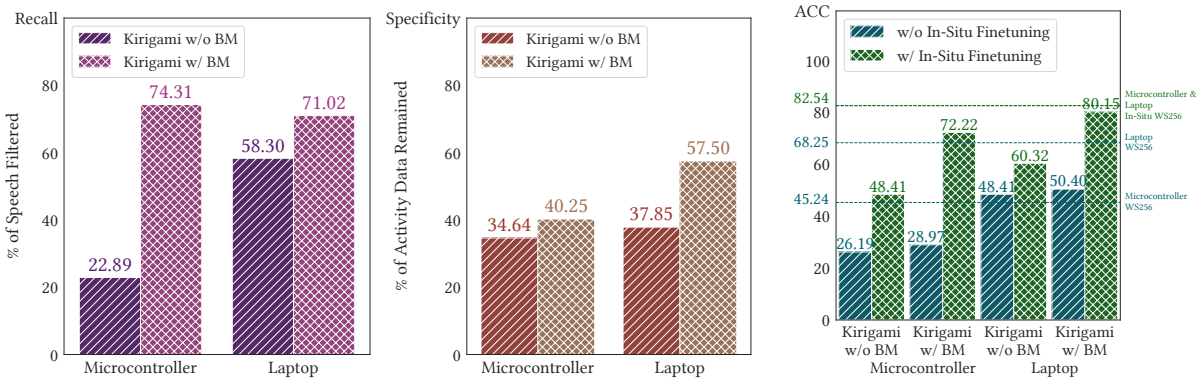
6.6 RQ3: Evaluation of Kirigami filter in the Real World

We conducted a user study to evaluate the robustness of our Kirigami filters for speech recognition in real-world environments beyond using audio datasets. In addition, we evaluate speech recognition accuracy in different locations with varying background noises and characterize the Kirigami filter's performance.

Scenarios Definition: We define three scenarios to characterize distinct speech and activity patterns in real-world settings. In scenario 1, to showcase the Kirigami's robustness to the duration of speech, participants are given a script containing randomly selected short and longer-duration sentences from the TIMIT dataset [15], each independently conveying complete meanings. They are asked to speak three short sentences, averaging 15 to 20 seconds each, and three longer sentences, taking approximately 1 minute, repeating each sentence three times. In scenario 2, participants are tasked with speaking short sentences at varying distances (1, 2, and 3 feet) from the source microphone. In Scenario 3, evaluating Kirigami's utility preservation, participants engage in diverse activities, including sporadic actions like clapping or typing, continuous actions like vacuum running, and human voice-based activities such as coughing or laughing. Each of these scenarios is repeated in three locations: a lab, a makerspace, and a conference room. We chose these locations to include a diverse range of background noise profiles.

Study Procedure: For this study, we recruited 7 participants (4 Females, 3 Males) ranging from 22 to 28 years old (Average = 24.7 years). We capture audio data from two input sources:(1) raw audio data from the Laptop (Macbook) microphone and (2) featurized FFT data from a microphone connected to a microcontroller (ARM Cortex-M4F with 256 KB RAM and 1 MB flash). Each participant is provided with a script and a set of sentences. To emulate a real-world setting, we only put constraints on the entire scenario's speech start and end time. However, participants can speak the sentences in any manner they see fit. For each scenario, we evaluate Kirigami's LR filter with and without a background mask, denoted as *Kirigami w/ BM* and *Kirigami w/o BM* and calculate speech and activity recognition accuracy.

Study Metrics: To measure the real-world performance of Kirigami filters to detect speech in the real world accurately, we use recall – percentage of speech removed when speech happens and specificity – percentage



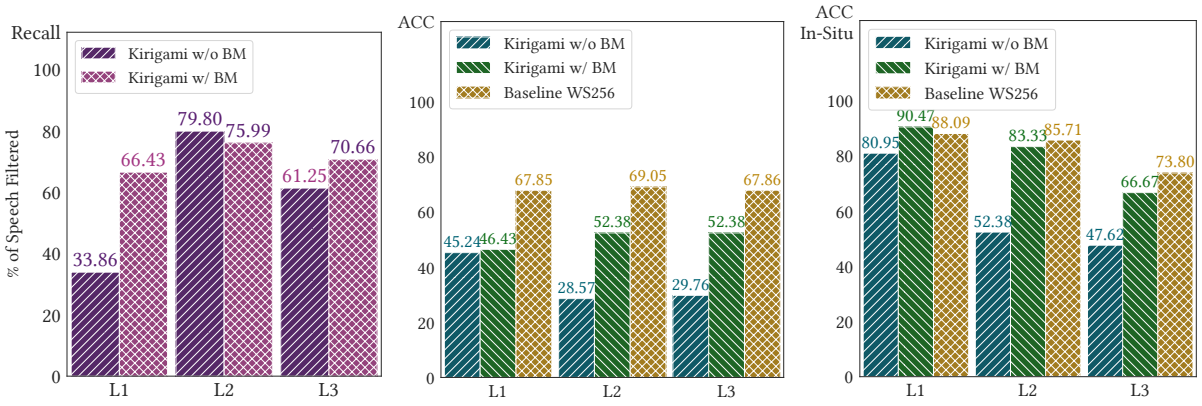
(a) Percentage of speech data filtered (b) Percentage of activity data retained (c) Activity recognition accuracy

Fig. 13. Comparing the performance of Kirigami LR filter with and without background mask from Microcontroller and Laptop. The figure shows the percentage of (a) speech-filtered and (b) activity data retained and overall activity recognition accuracy.

data available for utility-based models. We selected these two metrics due to their ability to evaluate filter performance independent of the composition ratio between speech and non-speech durations in real-world scenarios. Depending on the application, the ratio of speech and non-speech data can be different from the ratio in our user study or dataset. For this reason, using accuracy as the metric in our case would be strongly affected by the composition ratio of speech in our user study and might not be an informative indicator of real-world performance. To measure the activity recognition performance, similar to before, we use an AST-based model and calculate the classification accuracy in the real world. We further fine-tuned the global model by taking partial activity data as training samples from the user study to show an increase in classification accuracy.

Overall User Study Results: Figure 13 shows the performance comparison between the Kirigami filter with and without the background mask when the filter runs on a Microcontroller or Laptop. This result was obtained after the study was conducted among diverse participants and conducted in different locations in the building (L1, L2, L3) with varying background noise.

In Figure 13 (a), we see that the Kirigami filter with the background mask (BM) consistently outperforms its counterpart without BM, both on micro-controller and laptop platforms. The filtered speech data percentage is notably higher with BM (Micro: 74.31%, Laptop: 71.02%) compared to without BM (Micro: 22.89%, Laptop: 58.30%). Similarly, the presence of BM results in a greater retention of activity data (Micro: 40.25%, Laptop: 57.50%) compared to without BM (Micro: 34.64%, Laptop: 37.85%), as depicted in Figure 13 (b). This observation suggests that the speech recognition performance of the Kirigami LR without BM is significantly impacted by the diverse array of background noises present in real-world scenarios. Furthermore, our analysis indicates that Kirigami with BM achieves higher activity recognition scores, particularly for the laptop (without fine-tuning: 60.32%, with in-situ fine-tuning: 80.15%), surpassing the scores obtained without BM (without fine-tuning: 48.41%, with in-situ fine-tuning: 60.32%). A similar trend is observed for the microcontroller, where Kirigami with BM (without fine-tuning: 28.97%, with in-situ fine-tuning: 72.22%) consistently outperforms Kirigami without BM (without fine-tuning: 26.19%, with in-situ fine-tuning: 48.41%). In summary, our findings demonstrate the consistent and robust performance of the Kirigami filter with the background mask in speech filtering across diverse



(a) Percentage of speech data filtered (b) Activity recognition accuracy (c) Activity recognition accuracy of model fine-tuned with in-situ data

Fig. 14. Comparing the performance of Kirigami LR filter on Laptop with and without background mask at various locations: L1-Lab, L2-Makerspace, L3-Conference Room.

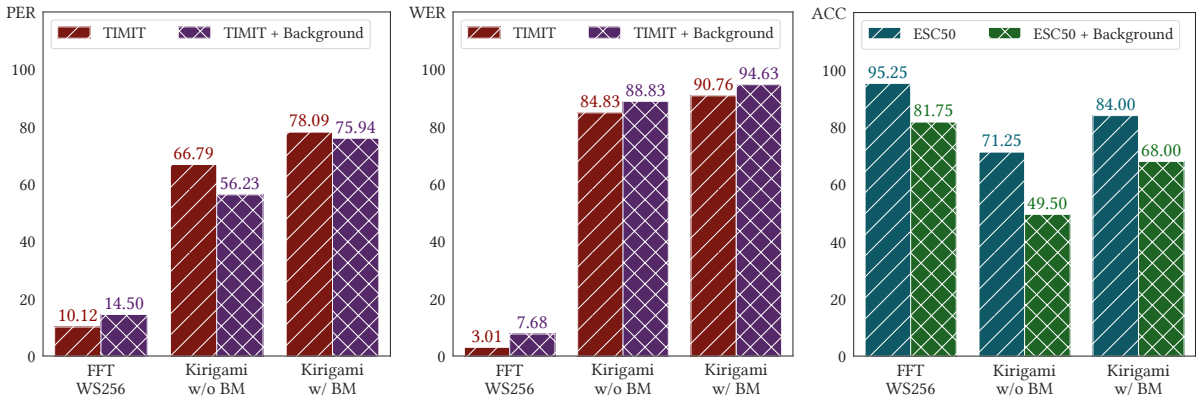
environments. In contrast, the Kirigami filter without the background mask experiences significant performance variations.

Evaluating Kirigami's Performance Across Different Environments: To examine the resiliency of Kirigami LR filter to various background profiles in real-world scenarios, we conducted the user study in three distinct locations characterized by diverse background settings in our campus building. Location L1 represented a laboratory setting, a shared space with multiple individuals, featuring a moderate-level background noise generated by continuous HVAC running and occasional conversations in the surroundings. Location L2, a maker space, exhibited a background profile dominated by low-frequency noise from machinery. In contrast, Location L3, a conference room with an open window, presented an external noise profile, including vehicle honks and birds chirping. Location L3 was the noisiest background environment, while L2 was deemed the least noisiest.

Figure 14 shows the comparison of Kirigami filter performance with and without BM in different locations. In general, the Kirigami filter without BM exhibits less accurate speech removal and demonstrates inconsistency across diverse locations. For instance, in location L1, a laboratory environment, the percentage of filtered speech decreases to 33.86% without BM, while the Kirigami filter with BM removes 66.43% of speech data. But in location L2, which is the quieter environment, we see that both Kirigami filter with and w/o filters filter have comparable speech percentage filtered showcasing Kirigami w/o BM performance changes in different locations. However, in comparison, Kirigami w/ BM speech filtering accuracy is consistent in different locations (L1: 66.43%, L2: 75.99%, and L3: 70.66%) while ensuring the utility of the data preserved is also high across different locations.

Evaluating Speech Inference and Activity Recognition using Kirigami with BM: Speech inference models often experience a decline in recognition accuracy when operating in noisy environments. Similarly, Kirigami filters, operating in a detect-and-remove manner, might also have to face challenges in filtering out speech in noisy environments. Therefore, we conducted empirical tests to confirm that Kirigami w/ BM maintains its robustness under ASR models when exposed to a noisy dataset. This investigation aims to assess the net impact of noisy data on speech inference and activity recognition.

In this evaluation, we use the TIMIT dataset and the same 10 classes ESC50 and various background environmental noises (*TIMIT + Background*) and (*ESC50 + Background*)/ The constructed dataset retains the same



(a) Phoneme Error Rate (PER) (b) Word Error Rate (WER) (c) Classification Accuracy

Fig. 15. Results of (a) phoneme-based speech inference, (b) word-based speech inference, and (c) activity classification accuracy on prior filtering approaches on the clean and noisy dataset.

structure as the original TIMIT and ESC50 datasets, except various background noises are overlaid. As before, we fine-tuned the CRDNN model, Wav2VecTransducer, and Whisper AI models to infer speech after filtering using Kirigami w/o BM and Kirigami w/ BM. We fine-tuned AST models for activity recognition using the filtered audio. In addition, we include the baseline model using an FFT of 256/128 windows and step sizes, without Kirigami filters, to gauge the impact of background noise on the ASR model alone.

Fig. 15 summarizes the performance of the Kirigami filter with and without BM under clean and noisy audio data. We include the summary of evaluation results for space limitation from only the strongest performers: Wav2VecTransducer and Fine-Tuned Whisper AI models. Overall, Kirigami w/ BM showed superior protection for speech privacy and preserved more utility values for activity recognition models than Kirigami w/o BM. As seen in Fig. 15 (a) and (b), Kirigami w/ BM remain high PER and WER across clean and noisy datasets, demonstrating its reliable privacy protection under noisy environments, while Kirigami w/o BM suffers from a degradation in PER. As seen in Fig. 15 (c) showed that Kirigami w/ BM outperforms Kirigami w/o BM on both the original ESC50 and noisy datasets. The superior performance of Kirigami w/ BM, even for the clean case, is possibly due to its capability to suppress intrinsic background noise in the original ESC50 dataset.

7 DISCUSSION AND LIMITATIONS

Evaluation or Privacy Filtering Models: Replicating and testing each proposed technique individually is time-consuming and expensive in terms of cloud computing credits needed, especially with model re-tuning or re-training. To address this, we carefully selected at least one prior work representing each type of privacy filtering that we identified. While this selection provides valuable insights into the performance of different filtering approaches, it may not encompass the entire spectrum of privacy filtering techniques available. Future research could explore a broader range of filtering techniques to gain an even more comprehensive understanding of their effectiveness and trade-offs. We believe that Kirigami's edge filtering approach to detect and filter speech-like segments will still remain superior to other approaches in terms of privacy.

Alternative ML Model for Filtering: We focused on using LR as the primary ML model used by our Kirigami filter rather than exploring other alternatives. While our results show that our LR-based Kirigami filter is quite

effective, other ML models specifically designed for edge devices, such as TinyML or lightweight recurrent neural network (RNN) models, could offer additional benefits and trade-offs. Our goal was to use a resource-frugal shallow model that could run on a wide range of IoT devices, but we leave the investigation of these alternative ML models as a future exploration.

User Study Validity: We acknowledge the smaller size of our participant pool for the user study as a limitation.A larger and more diverse sample size would further enhance the validity and generalizability of the study results. A larger sample would also provide a broader representation of user preferences, behaviors, and perceptions, leading to more robust conclusions.

Additional Metrics for Speech Privacy: Our study highlights an important consideration regarding the use of Phoneme Error Rate (PER) and Word Error Rate (WER) as metrics for evaluating speech privacy. While PER and WER are commonly used metrics for assessing the performance of automatic speech recognition (ASR) systems, they are not specifically designed for privacy evaluation. Although we measured and reported the "safe zones" based on our user study, indicating areas where privacy is preserved, it is important to note that these safe zones are not guaranteed to be completely safe from privacy risks. Our findings suggest that while PER and WER are useful in determining the privacy characteristics of audio featurization, they should be complemented with additional privacy evaluation measures to provide a more comprehensive assessment of speech privacy. Further research into specialized metrics or evaluation methodologies for speech privacy would contribute to the development of more reliable and robust privacy evaluation frameworks.

8 CONCLUSION

Deep learning-based automatic speech recognition (ASR) has posed new challenges to privacy-focused audio featurization techniques. Such a risk exists primarily because modern ASR systems can be tuned to recognize speech content specifically to these audio featurization techniques. We aim to systematically characterize various featurization techniques on audio data, particularly those that extract statistical and spectral features using Fast Fourier Transforms (FFTs), and evaluate the privacy risks and utility tradeoffs. We first explore different FFT-based featurization approaches proposed in prior works that aim to remove sensitive information from raw audio while providing utility to activity recognition tasks. We then study the recent advancements in deep learning-based automatic speech recognition (ASR) and their potential impact on these edge audio featurization techniques. We also investigate the utility of different featurization approaches in generating discernible features for machine learning prediction. We then propose Kirigami, a general-purpose edge audio speech filter resilient to various speech recognition or audio reconstruction techniques while being feasible to implement on edge devices with limited computational power. We plan to open-source our Kirigami codebase for researchers and practioners to use and build upon.

ACKNOWLEDGMENTS

This work was partially supported by NSF Award SaTC-1801472 and the CMU's CyLab Security and Privacy Institute. We gratefully acknowledge a gift by JP Morgan Chase to support research on smart buildings at Carnegie Mellon. We want to thank Chris Harrison, Vimal Mollyn, and Riku Arakawa for their invaluable feedback on the early revisions of the paper. We also thank our anonymous reviewers for their constructive feedback on our paper.

REFERENCES

[1] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. Wav2vec 2.0: a framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33, 12449–12460.

[2] Yang Bai, Li Lu, Jerry Cheng, Jian Liu, Yingying Chen, and Jiadi Yu. 2020. Acoustic-based sensing and applications: a survey. *Computer Networks*, 181, 107447.

[3] Aldo Luiz Bizzocchi. 2017. How many phonemes does the english language have? *International Journal on Studies in English Language and Literature (IJSELL)*, 5, 10, 36–46.

[4] Jeremy A Blumenthal, Meera Adya, and Jacqueline Mogle. 2008. The multiple dimensions of privacy: testing lay expectations of privacy. *U. Pa. J. Const. L.*, 11, 331.

[5] Sudershan Boovaraghavan, Chen Chen, Anurag Maravi, Mike Czapik, Yang Zhang, Chris Harrison, and Yuvraj Agarwal. 2023. Mites: design and deployment of a general-purpose sensing infrastructure for buildings. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 7, 1, Article 2, 32 pages. DOI: 10.1145/3580865.

[6] Herve A Bourlard and Nelson Morgan. 1994. *Connectionist speech recognition: a hybrid approach*. Vol. 247. Springer Science & Business Media.

[7] Francine Chen, John Adcock, and Shruti Krishnagiri. 2008. Audio privacy: reducing speech intelligibility while preserving environmental sounds. In *Proceedings of the 16th ACM international conference on Multimedia*, 733–736.

[8] Bhawana Chhaglani, Camellia Zakaria, Adam Lechowicz, Jeremy Gummeson, and Prashant Shenoy. 2022. Flowsense: monitoring airflow in building ventilation systems using audio sensing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6, 1, 1–26.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[10] The CMU Pronouncing Dictionary. [n. d.] *http://www.speech.cs.cmu.edu/cgi-bin/cmudict*.

[11] D. Dolev and A. Yao. 1983. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29, 2, 198–208. DOI: 10.1109/TIT.1983.1056650.

[12] Julia C Dunbar, Emily Bascom, Ashley Boone, and Alexis Hiniker. 2021. Is someone listening? audio-related privacy perceptions and design recommendations from guardians, pragmatists, and cynics. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5, 3, 1–23.

[13] Zhiyun Fan, Meng Li, Shiyu Zhou, and Bo Xu. 2020. Exploring wav2vec 2.0 on speaker verification and language identification. *arXiv preprint arXiv:2012.06185*.

[14] Luciano Floridi and Massimo Chiriatti. 2020. Gpt-3: its nature, scope, limits, and consequences. *Minds and Machines*, 30, 681–694.

[15] John S Garofolo. 1993. Timit acoustic phonetic continuous speech corpus. *Linguistic Data Consortium, 1993*.

[16] Yuan Gong, Yu-An Chung, and James Glass. 2021. Ast: audio spectrogram transformer. *arXiv preprint arXiv:2104.01778*.

[17] Alex Graves. 2012. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*.

[18] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, 369–376.

[19] Wen-Chin Huang, Chia-Hua Wu, Shang-Bao Luo, Kuan-Yu Chen, Hsin-Min Wang, and Tomoki Toda. 2021. Speech recognition by simply fine-tuning bert. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 7343–7347.

[20] Yasha Iravantchi, Karan Ahuja, Mayank Goel, Chris Harrison, and Alanson Sample. 2021. Privacymic: utilizing inaudible frequencies for privacy preserving daily activity recognition. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–13.

[21] Sunder Ali Khowaja, Aria Ghora Prabono, Feri Setiawan, Bernardo Nugroho Yahya, and Seok-Lyong Lee. 2018. Contextual activity based healthcare internet of things, services, and people (hiotsp): an architectural framework for healthcare monitoring using wearable sensors. *Computer Networks*, 145, 190–206.

[22] Davi Miara Kiapuchinski, Carlos Raimundo Erig Lima, and Celso Antônio Alves Kaestner. 2012. Spectral noise gate technique applied to birdsong preprocessing on embedded unit. In *2012 IEEE International Symposium on Multimedia*, 24–27. DOI: 10.1109/ISM.2012.12.

[23] Sumeet Kumar, Le T Nguyen, Ming Zeng, Kate Liu, and Joy Zhang. 2015. Sound shredding: privacy preserved audio sensing. In *Proceedings of the 16th international workshop on mobile computing systems and applications*, 135–140.

[24] Gierad Laput, Karan Ahuja, Mayank Goel, and Chris Harrison. 2018. Ubicoustics: plug-and-play acoustic activity recognition. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (UIST '18). Association for Computing Machinery, Berlin, Germany, 213–224. ISBN: 9781450359481. DOI: 10.1145/3242587.3242609.

[25] Gierad Laput, Yang Zhang, and Chris Harrison. 2017. Synthetic sensors: towards general-purpose sensing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 3986–3999.

[26] Eric C Larson, TienJui Lee, Sean Liu, Margaret Rosenfeld, and Shwetak N Patel. 2011. Accurate and privacy preserving cough sensing using a low-cost microphone. In *Proceedings of the 13th international conference on Ubiquitous computing*, 375–384.

[27] Nathan Malkin, Joe Deatrick, Allen Tong, Primal Wijesekera, Serge Egelman, and David Wagner. 2019. Privacy attitudes of smart speaker users. *Proceedings on Privacy Enhancing Technologies*, 2019, 4, 250–271.

[28] Cecilia Mascolo. 2020. Listen to your health: reflections on mobile health diagnostics through audio signals. In *Proceedings of the 21st International Workshop on Mobile Computing Systems and Applications*, 1–1.

[29] Vimal Mollyn, Karan Ahuja, Dhruv Verma, Chris Harrison, and Mayank Goel. 2022. Samosa: sensing activities with motion and subsampled audio. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6, 3, 1–19.

[30] Joseph D O'Connor. 1980. *Better English Pronunciation*. Cambridge University Press.

[31] Madhurananda Pahar, Igor Miranda, Andreas Diacon, and Thomas Niesler. 2022. Automatic non-invasive cough detection based on accelerometer and audio signals. *Journal of Signal Processing Systems*, 94, 8, 821–835.

[32] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 5206–5210.

[33] Yi-Hao Peng et al. 2018. Speechbubbles: enhancing captioning experiences for deaf and hard-of-hearing people in group conversations. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (CHI '18). Association for Computing Machinery, Montreal QC, Canada, 1–10. ISBN: 9781450356206. DOI: 10.1145/3173574.3173867.

[34] Karol J. Piczak. 2015. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*. ACM Press, Brisbane, Australia, (Oct. 13, 2015), 1015–1018. ISBN: 978-1-4503-3459-4. DOI: 10.1145/2733373.2806390.

[35] Pincelate. [n. d.] *https://github.com/aparrish/ pincelate/*.

[36] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2212.04356*.

[37] Mirco Ravanelli et al. 2021. Speechbrain: a general-purpose speech toolkit. *arXiv preprint arXiv:2106.04624*.

[38] Chandan KA Reddy, Ebrahim Beyrami, Jamie Pool, Ross Cutler, Sriram Srinivasan, and Johannes Gehrke. 2019. A scalable noisy speech dataset and online subjective test framework. *Proc. Interspeech 2019*, 1816–1820.

[39] Qun Song, Chaojie Gu, and Rui Tan. 2018. Deep room recognition using inaudible echos. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2, 3, 1–28.

[40] The Verge. 2023. A new hack can turn an Echo into a live microphone. https://www.theverge.com/2017/8/1/16079044/amazon-echo-hack-microphone-listen-in-mark-barnes. (2023).

[41] Yingzhi Wang, Abdelmoumene Boumadane, and Abdelwahab Heba. 2021. A fine-tuned wav2vec 2.0/hubert benchmark for speech emotion recognition, speaker verification and spoken language understanding. *arXiv preprint arXiv:2111.02735*.

[42] Xuhai Xu, Ebrahim Nemati, Korosh Vatanparvar, Viswam Nathan, Tousif Ahmed, Md Mahbubur Rahman, Daniel McCaffrey, Jilong Kuang, and Jun Alex Gao. 2021. Listen2cough: leveraging end-to-end deep learning cough detection model to enhance lung health assessment using passively sensed audio. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 5, 1, Article 42, 22 pages. DOI: 10.1145/3448124.

[43] Yunke Zhang, Fengli Xu, Tong Li, Vassilis Kostakos, Pan Hui, and Yong Li. 2021. Passive health monitoring using large scale mobility data. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 5, 1, Article 49, 23 pages. DOI: 10.1145/3448078.

[44] Xianrui Zheng, Chao Zhang, and Philip C Woodland. 2021. Adapting gpt, gpt-2 and bert language models for speech recognition. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 162–168.