Privacy-Preserving Feature Extraction for Cloud-Based Wake Word Verification

Timm Koppelmann, Alexandru Nelus, Lea Schönherr, Dorothea Kolossa, and Rainer Martin

Institute of Communication Acoustics, Ruhr-Universität Bochum, Bochum, Germany

{timm.koppelmann, alexandru.nelus, lea.schoenherr, dorothea.kolossa, rainer.martin}@rub.de

Abstract

Wake word detection and verification systems often involve a local, on-device wake word detector and a cloud-based verification node. In such systems, the audio representation sent to the cloud-based server may exhibit sensitive information that might be intercepted by an eavesdropper. To improve privacy of cloud-based wake word verification (WWV) systems, we propose to use a privacy-preserving feature representation that minimizes the automatic speech recognition (ASR) capability of a potential attacker. The proposed approach employs an adversarial training schedule that aims to minimize an attacker's word error rate (WER) while maintaining a high WWV performance. To this end, we apply an adaptive weighting factor in the combined loss function to control the balance between minimizing the WWV loss and maximizing the ASR loss. We show that the proposed training method significantly reduces possible privacy risks while maintaining a strong WWV performance.

Index Terms: Wake word verification, wake word detection, privacy, automatic speech recognition, adversarial training

1. Introduction

Recent advances in machine learning (ML) and signal processing along with increasing availability of embedded devices have led to a wide dissemination of voice-controlled human-machine interfaces. Controlling a smartphone, a computer or even a vehicle via voice brings several benefits but also implies obvious privacy concerns and might lead to unsatisfactory performance issues when *automatic speech recognition* (ASR) operates continuously. A common safe-guard is to activate the ASR system only after specific voice commands, entitled *wake words*, are uttered, e.g., ”Ok, Google” or ”Alexa”.

The process of spotting a wake word, denoted as *wake word detection* (WWD), is usually performed at local device level in order to hinder the transmission of sensitive information via a (potentially) fallible communication medium [1]. The field of single-device solutions for WWD is continually progressing, with performance and computational efficiency taking a central role [2, 3]. Nevertheless, it is common practice in current Internet-of-Things (IoT) implementations to transmit raw audio or audio features via the Internet to a more resourceful cloud-based system for an improved WWD decision [4, 5, 6], which we denote as *wake word verification* (WWV). The sharing of WWV-purposed audio or audio features implies inherent privacy risks, as such data can be intercepted by an ASR-based eavesdropper and used to perform speech-to-text transcription [7]. This scenario is further illustrated in Figure 1.

Given the proposed framework, we provide a proof-of-concept for privacy-preserving high-level feature representations that maintain full functionality for cloud-based WWV. At the same time, upon interception, this feature representation is designed to reduce the privacy risks posed by an ASR-based attacker. Such an approach follows the *privacy-by-design* (PbD) principle stipulated by the European Union General Data Protection Regulation (GDPR) [8] and might be used as an extra layer of privacy, in addition to other privacy-preserving techniques like data encryption. In this work we focus on alleviating some of the aforementioned privacy risks rather than optimizing the performance of a cloud-based WWV system in comparison to a local detector. We assume that the cloud-based WWV system with (in principle) unlimited computational resources and, e.g., more frequently updated back-end model will always outperform a locally-embedded WWD system.

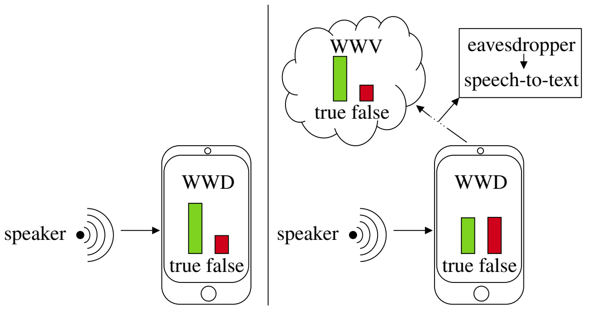

Figure 1: *Privacy risks in wake word detection (WWD) and verification (WWV) systems. Left: local WWD, no feature interception is possible. Right: WWD with cloud-based WWV: low WWD confidence results in transmission of data to a cloud-based WWV system. This data can be intercepted by an attacker and used to perform speech-to-text transcription.*

Several methods for improving privacy at the feature level in distributed audio applications have been proposed, ranging from adversarial [9] and Siamese [10] to variational information training [11]. These had the purpose of disentangling sources of variations such as speaker gender or domestic activity information from speaker identity. The methods have not been applied in WWD or WWV scenarios and no ASR-based attack on intercepted features has been previously studied.

To this end, we define our baseline WWV system and divide it into a *feature extractor*, which we envision residing on the local device, and a *verifier* residing on the server side. It is assumed that the feature extractor is public (*white-box* attack) and provides the feature representation Z. This allows the attacker to train an ASR system which can be further used to perform speech-to-text transcription on intercepted features. It is shown that Z, although designed with the end goal of WWV, still carries a significant amount of ASR-accessible data.

We then propose to use dimensionality reduction and adversarial training in order to transform the baseline WWV system into a *privacy-preserving* WWV system. As a consequence, a

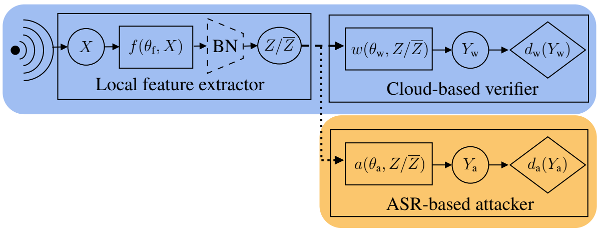
X
$f(\theta_\mathrm{f}, X)$
BN
$Z/\overline{Z}$
Local feature extractor
$w(\theta_\mathrm{w}, Z/\overline{Z})$
Y_w
$d_\mathrm{w}(Y_\mathrm{w})$
Cloud-based verifier
$a(\theta_\mathrm{a}, Z/\overline{Z})$
Y_a
$d_\mathrm{a}(Y_\mathrm{a})$
ASR-based attacker

Figure 2: *WWV system (blue box) and attacker system (orange box). The feature extractor computes either Z (baseline system) or $\overline{Z}$ (privacy-preserving system). While both systems use a similar network topology, the latter also employs adversarial training and optionally also a bottleneck layer (BN).*

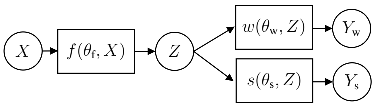
X
$f(\theta_\mathrm{f}, X)$
Z
$w(\theta_\mathrm{w}, Z)$
Y_w
$s(\theta_\mathrm{s}, Z)$
Y_s

Figure 3: *Flow chart of the multi-task learning setup: A high-level feature extractor f feeds two parallel dense output layers. Only the upper path is further used for the baseline WWV system.*

new privacy-preserving feature representation $\overline{Z}$ is computed and is tested against an ASR-based eavesdropper, showing a significant decrease in ASR-related privacy risks with only a minor loss in WWV performance.

The paper is structured as follows: in Section 2 we present the proposed baseline WWV system, followed by a description of the ASR-based attacker and the privacy-preserving WWV system. The proposed privacy-preserving training approach is presented in Section 3, followed by the experimental setup and results in Sections 4 and 5. Conclusions are drawn in Section 6.

2. System description

2.1. Baseline WWV system

Based on the work of Sun et al. [3], we use a time-delay neural network (TDNN) [12] as local feature extractor f, transforming low-level audio features X into a high-level feature representation Z. The extracted features Z are then processed by a dense layer w with output Y_w and a subsequent speech decoder d_w, simulating a cloud-based WWV application. The topology of the WWV architecture is depicted in the blue box in Figure 2.

Closely following the implementation in [3], the feature extractor f is initialized as an acoustic model for ASR, employing a separate dense output layer s with output Y_s and ASR-purposed training targets. After initialization, layers s and w are concomitantly used in conjunction with f for joint training of ASR and WWV, as depicted in Figure 3. This multi-task learning approach minimizes the combined loss function

$$\min_{\theta_\mathrm{f},\theta_\mathrm{w},\theta_\mathrm{s}} [\lambda L_\mathrm{w}(\theta_\mathrm{f}, \theta_\mathrm{w}) + (1-\lambda) L_\mathrm{s}(\theta_\mathrm{f}, \theta_\mathrm{s})], \qquad (1)$$

w.r.t. the involved weights and biases, θ_f, θ_w, and θ_s, where L_w and L_s describe the WWV and the ASR loss, respectively, and the weighting factor λ controls the contributions of the two goals. More information about the initialization and multi-task learning is provided in Section 4.3. After multi-task learning, only the feature extractor f and the subsequent output layer w are further used in our baseline WWV system alongside the lattice-based speech decoder d_w, which is not trained but only used for evaluation purposes, as described in Section 4.4.

2.2. ASR-based attacker

We consider an attacker that intercepts the exported feature representation via eavesdropping and aims to obtain the speech transcription using ASR. As depicted in the orange box in Figure 2, the proposed attacker employs a single dense layer a with output Y_a, followed by a speech decoder d_a. The privacy risk resulting from using the proposed WWV model is then measured by the attacker ASR performance. Considering that the baseline WWV model relies on an ASR-based initialization via transfer learning and employs the presented multi-task training scheme, a potentially high privacy risk of transcribing speech from Z becomes obvious. This has also been confirmed by our experiments presented in Section 5.

2.3. Privacy-preserving WWV system

Our work aims to tackle the aforementioned privacy risks of WWV systems by developing a privacy-preserving high-level feature representation $\overline{Z}$, which is also indicated in Figure 2 and further detailed in Section 3. By using this proposed feature representation, we want to obtain a WWV performance similar to that of the baseline WWV system but at the same time drastically decrease the attacker's ASR performance.

3. Privacy-preserving training

3.1. Dimensionality reduction

A first approach for tackling the ASR-related privacy risk is to reduce the amount and the complexity of information that can be represented in $\overline{Z}$. We first perform the ASR initialization described in Section 2.1 and then add a bottleneck (BN) layer of size D to the feature extractor f in order to compress the feature space. Thus, we decrease the capacity of $\overline{Z}$ to represent detailed speech information. We then continue with the same multi-task learning procedure described in Section 2.1 and systematically modify the size of the BN layer (and thus also of $\overline{Z}$) in order to find a suitable trade-off between WWV and ASR performance. Results are shown in Section 5.

3.2. Adversarial training

The second route to lower ASR-related privacy risks, in conjunction with a reduced dimensionality of $\overline{Z}$, is the usage of an adversarial training schedule. This can be split into an initial pre-training step followed by two alternating adversarial steps.

In the pre-training step, we apply the same multi-task training procedure as in Section 3.1. This already produces a well-performing privacy-preserving WWV system. In order to further deprecate the attacker's ASR performance, additional training is performed by taking into consideration the attacker model. This is done in several adversarial iterations of the following two steps:

1. Further train the WWV system by considering the attacker model: Adapt $\overline{Z}$ so that it minimizes WWV loss L_w while simultaneously maximizing the attacker's ASR loss L_a. The trade-off between WWV and ASR performance is controlled by the scaling factor λ. In this new system, the attacker weights and biases θ_a are frozen while only the parameters θ_f and θ_w of the WWV system are updated according to:

$$\min_{\theta_\mathrm{f},\theta_\mathrm{w}} [\lambda L_\mathrm{w}(\theta_\mathrm{f}, \theta_\mathrm{w}) - (1-\lambda) L_\mathrm{a}(\theta_\mathrm{f}, \theta_\mathrm{a})]\,. \qquad (2)$$

2. Attacker training: Perform attacker ASR training using the intercepted high-level feature representation $\overline{Z}$ from

the previous step. The feature extractor weights and biases θ_f remain fixed while only the attacker's parameters θ_a are now updated according to:

$$\min_{\theta_\mathrm{a}}[L_\mathrm{a}(\theta_\mathrm{f},\theta_\mathrm{a})]. \tag{3}$$

The training process is further optimized by dynamically adapting the weighting factor λ in every training iteration. This prevents a quickly increasing L_a from dominating the combined loss term in (2), which would lead to a highly unbalanced training process. Therefore, we set

$$\lambda = 1 - \frac{L_\mathrm{w}(\theta_\mathrm{f},\theta_\mathrm{w})}{\psi L_\mathrm{a}(\theta_\mathrm{f},\theta_\mathrm{a})}. \tag{4}$$

During training, a strong attacker counteracts the goal of improving privacy-preservation while a poorly performing attacker can have too much of a negative influence on the WWV. Therefore, we set $\psi = 20$, which empirically leads to a good balance between these constraints. Note that we used a state-of-the-art ASR framework (see Section 4) as attacker and therefore imply full access to the attacker's model and all its parameters.

4. Experimental setup

4.1. Wake word selection

We use ”Mister” as the designated wake word, as it consists of two syllables and therefore has a good pronunciation length. Very short wake words, like ”Hi”, ”Stop”, or ”Down”, might cause a high number of false detections, whereas very long wake words might not be sufficiently user-friendly. Additionally, ”Mister” has a reasonably high number of occurrences in the dataset (see Table 1), which is required for training and testing a solid baseline system.

4.2. WWV and ASR database

All experiments in this work utilize the LibriSpeech database [13]. We only use clean data for the proposed concept as to better observe the limitations of both WWV and ASR attacker. In both cases, we employ the *train-clean-100* subset for training, *dev-clean* for validation, and *test-clean* for testing. In order to extend the WWV test set, we also add all occurrences of ”Mister” from the *train-clean-360* subset. Table 1 shows the number of positive examples (occurrences of ”Mister”) and negative examples (every other word occurrence) in the different sets.

Table 1: *Positive and negative examples of the wake word ”Mister” in the used datasets. For wake word verification, 3985 further positive examples are added to the test set.*

Set	Pos. examples	Neg. examples
Training	1199	990101
Validation	52	54402
Evaluation	48+3985	52576

4.3. Neural network configuration and training

Table 2 displays the structure of our TDNN feature extractor f, including the considered context size and layer dimensionality. After each layer, a batch normalization layer is inserted. The subsequent output layers w, s, and a map the high-level feature representations $Z/\overline{Z}$ to context-dependent phonetic states of hidden Markov model (HMM)-based *language models* (LMs). We use the Kaldi toolkit [14] to build a large-vocabulary LM for ASR and a highly reduced LM for WWV, containing phonetic representations of ”Mister” and background/noise fillers.

Table 2: *Context and dimensionality of the layers of the TDNN feature extractor f. The input layer processes the narrow frame context, while higher layers consider wider context ranges.*

Layer	Context	Dimensionality
Input	$\{-2,-1,0,+1,+2\}$	512
Layer 1	$\{-2,0,+2\}$	512
Layer 2	$\{-4,0,+4\}$	512
Output	$\{0\}$	1500

As low-level input features X, we used the first 13 *Mel-Frequency Cepstral Coefficients* (MFCCs) [15] and their first and second derivatives, calculated over frames of 25 ms with a 10 ms frame shift. Training targets have been created by performing forced alignment in Kaldi, using an auxiliary Gaussian mixture model (GMM) acoustic model and the LMs on the training set.

The neural networks in this work have been implemented in Python, utilizing the PyTorch-Kaldi toolkit [16] in addition to the TDNN implementation from [17]. All networks have been trained by minimizing the cross-entropy losses $L_\mathbf{w}$, $L_\mathbf{s}$, and $L_\mathbf{a}$ using the *Root Mean Square Propagation* (RMSprop) optimizer. We used an initial learning rate of $l_\mathrm{r} = 0.008$ for all training processes except for training the WWV system during the adversarial iterations. Here we started with $l_\mathrm{r} = 0.005$ and gradually decreased across training iterations. To prevent overfitting, all layers of the feature extractor f apply a dropout rate of 0.15 during training.

The baseline WWV system was initialized with 30 epochs of ASR training, followed by 25 epochs of multi-task learning, with or without an inserted BN layer. Here, we empirically set $\lambda = 0.99$. Each adversarial iteration was scheduled as follows: in the first step, the WWV system was trained for a minimum of 15 and a maximum of 30 epochs or until $L_\mathrm{a} > 4000$. In the second step, the attacker was trained for 5 epochs. In order to best observe the evolution of $\overline{Z}$ across adversarial iterations, we separately train a stronger attacker using 25 additional epochs of ASR training after each adversarial iteration. This model does not influence the actual adversarial training process.

4.4. Evaluation metrics

We composed the acoustic and language models into *weighted finite-state transducers* (WFSTs) [18] and employed Kaldi's WFST-lattice-decoder to perform speech decoding with the neural network outputs. For ASR evaluation, we calculated the *word error rate* (WER) for the given evaluation set with the decoded outputs of layer s or a. For WWV, we use *detection error trade-off* (DET) curves, plotting the false alarm rate against the false reject rate. We compared the decoded outputs of layer w with each utterance in the test set. We assumed a correct detection if the number of detected wake words matched the number of actual occurrences in the respective utterance. Different points on the DET curves have been calculated by modifying the ratio between acoustic and language model weights in the Kaldi decoder, prioritizing either the language or acoustic transition probabilities in the decoding graph, as indicated in [14].

5. Results and discussion

In Figure 4 we present the performance of the baseline WWV system along with the first version of privacy-preserving WWV system, which uses only dimensionality reduction via a BN

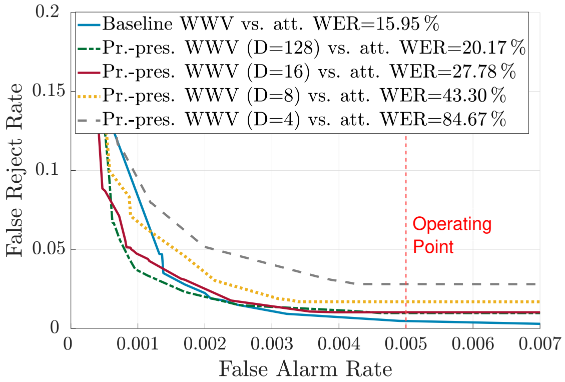

Figure 4: *Wake word verification (WWV) detection error trade-off (DET) curves for the baseline WWV system and the privacy-preserving WWV system. The latter is based solely on dimensionality reduction, where D indicates the size of the bottleneck layer. The performance of an automatic speech recognition (ASR) attacker using the intercepted features is indicated by the word error rate (WER).*

layer. The size D of the BN layer is systematically varied. The baseline WWV system performs similarly to other state-of-the-art implementations [2, 3]. However, an attacker that has intercepted these features can easily use them for ASR, obtaining a WER of 15.95 %. When we include the BN layer and reduce D, we observe increasing WER for the attacker. At the same time, WWV performance is increasingly reduced. Although we can identify suitable operational points (e.g. for $D = 16$ at a false alarm rate of 0.5 % and a false reject rate of 1 %) with a good trade-off between WWV and ASR, we strive for further improvements by employing adversarial training.

Based on previous observations, we now use adversarial training in conjunction with a BN size of $D = 16$ and additionally in conjunction with a WWV system without a BN layer for comparison. Training is performed as indicated in Section 4.3, where after each adversarial iteration (the two alternating steps described in Section 3.2) we separately train a stronger attacker to observe the limitations of $\overline{Z}$. The results are presented in Figure 5. It can be seen that more adversarial iterations have a positive effect on the privacy-preservation of $\overline{Z}$ by strongly increasing the WER of attackers. The increase of the WER over the iterations is much more significant for the system that includes the BN layer: after the 15-th adversarial iteration we achieve 90.81 % WER compared to 55.21 % for the non-BN-layer approach.

Finally, we examine the impact of the adversarial training procedure on WWV performance by comparing the two aforementioned privacy-preserving WWV systems (BN layer of size $D = 16$ and without a BN layer). We specifically consider the state before adversarial training ($A_i = 0$) and after 15 adversarial iterations ($A_i = 15$), cf. Figure 6. At the desired operating point where the false alarm rate is 0.5 % [2], both systems show a similar performance close to 1.4 % false reject rate for $A_i = 15$. Furthermore, comparing $A_i = 0$ and $A_i = 15$, the loss in WWV performance is smaller for the BN-based system than for the non-BN system. Thus, while both systems perform similarly on the WWV task, the BN-based system achieves a much larger attacker WER of 90.81 %.

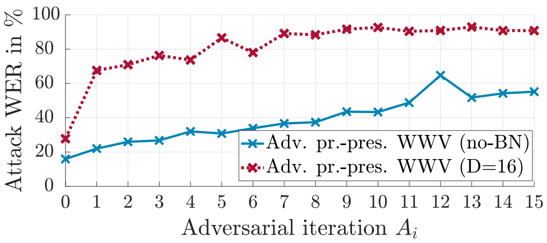

Figure 5: *ASR word error rate (WER) vs. adversarial training iterations of a strong attacker when using privacy-preserving wake word features without a bottleneck (BN) layer (no-BN) and with a BN layer of size $D = 16$. Iteration 0 indicates the state before adversarial training.*

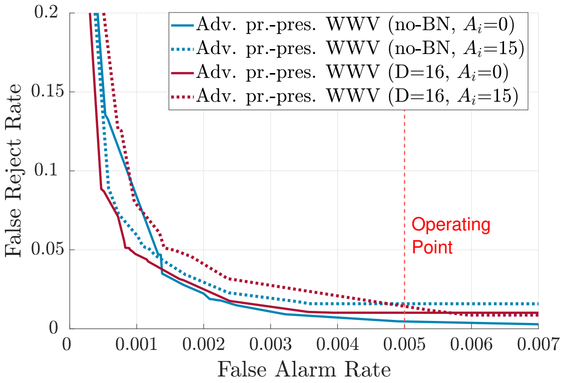

Figure 6: *Wake word verification DET curves for adversarial-training-based privacy-preserving WWV systems with a bottleneck (BN) layer of size $D = 16$ and without a BN layer. Performance before adversarial training ($A_i = 0$) and after $A_i = 15$ adversarial iterations are displayed.*

6. Conclusions & Future Work

We have proposed a privacy-preserving distributed WWV system comprising local on-device feature extraction and cloud-based post-processing. In this scenario, an ASR-based attacker may intercept the features transmitted between local device and cloud server and may use them to transcribe speech. It has been shown that features specifically trained for WWV also carry a significant amount of speech-related information. We have proposed to alleviate this ASR-related privacy risk by employing a feature dimensionality reduction using a BN layer along with adversarial training. This approach has been proven successful as we were able to maintain low false-alarm and false rejection rates while drastically reducing an attacker's ASR performance.

While this work serves as a proof of concept, we expect it to be even more useful when combined with fusion approaches in distributed sensor scenarios. We aim to expand this work into more complex scenarios that will include reverberant signals and multiple distributed smart home devices. Additionally, we plan to examine the impact of different wake word lengths on WWV performance.

7. Acknowledgements

This work has been supported by the German Research Foundation (DFG) - Project Number 282835863 and under Germany's Excellence Strategy - EXC 2092 CASA - 390781972.

8. References

[1] W. Ali, G. Dustgeer, M. Awais, and M. A. Shah, "Iot based smart home: Security challenges, security requirements and solutions," in *23rd International Conference on Automation and Computing, ICAC 2017, Huddersfield, United Kingdom, September 7-8, 2017*. IEEE, 2017, pp. 1–6.

[2] G. Chen, C. Parada, and T. N. Sainath, "Query-by-example keyword spotting using long short-term memory networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*. IEEE, 2015, pp. 5236–5240.

[3] M. Sun, D. Snyder, Y. Gao, V. K. Nagaraja, M. Rodehorst, S. Panchapagesan, N. Strom, S. Matsoukas, and S. Vitaladevuni, "Compressed time delay neural network for small-footprint keyword spotting," in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, F. Lacerda, Ed. ISCA, 2017, pp. 3607–3611.

[4] European Data Protection Board, "Guidelines 02/2021 on virtual voice assistants," 2021 (accessed March 22, 2021), https://edpb.europa.eu/our-work-tools/public-consultations-art-704/2021/guidelines-022021-virtual-voice-assistants_en.

[5] L. Schönherr, M. Golla, T. Eisenhofer, J. Wiele, D. Kolossa, and T. Holz, "Unacceptable, where is my privacy? Exploring accidental triggers of smart speakers," *CoRR*, vol. abs/2008.00508, 2020.

[6] Amazon.com, "Amazon developer," 2018 (accessed October 12, 2020), https://forums.developer.amazon.com/articles/107401/action-required-client-code-changes-and-feature-en.html,.

[7] P. Kumar, A. Gurtov, J. Iinatti, M. Ylianttila, and M. Sain, "Lightweight and secure session-key establishment scheme in smart home environments," *IEEE Sensors Journal*, vol. 16, pp. 1–1, 01 2015.

[8] European Parliament and Council, "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)," 2016.

[9] A. Nelus and R. Martin, "Gender discrimination versus speaker identification through privacy-aware adversarial feature extraction," in *Proceedings of the 13th ITG Symposium on Speech Communication, Oldenburg, Germany, October 10-12, 2018*. VDE / IEEE, 2018, pp. 1–5.

[10] A. Nelus, S. Rech, T. Koppelmann, H. Biermann, and R. Martin, "Privacy-preserving siamese feature extraction for gender recognition versus speaker identification," in *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, G. Kubin and Z. Kacic, Eds. ISCA, 2019, pp. 3705–3709.

[11] A. Nelus, J. Ebbers, R. Haeb-Umbach, and R. Martin, "Privacy-preserving variational information feature extraction for domestic activity monitoring versus speaker identification," in *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, G. Kubin and Z. Kacic, Eds. ISCA, 2019, pp. 3710–3714.

[12] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. Interspeech 2015*. ISCA, 2015, pp. 3214–3218.

[13] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "LibriSpeech: An ASR corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 5206–5210.

[14] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011, iEEE Catalog No.: CFP11SRW-USB.

[15] S. Furui, *Digital Speech Processing: Synthesis, and Recognition, Second Edition,*, ser. Signal Processing and Communications. Taylor & Francis, 2000.

[16] M. Ravanelli, T. Parcollet, and Y. Bengio, "The pytorch-kaldi speech recognition toolkit," in *Proc. of ICASSP*, 2019.

[17] C. Luu, "TDNN," 2019 (accessed October 19, 2020), https://github.com/cvqluu/TDNN.

[18] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," in *Computer Speech & Language*, vol. 16.1, 2002, pp. 69–88.