Available online at www.sciencedirect.com

ScienceDirect

Computer Speech and Language 42 (2017) 20–37

COMPUTER SPEECH AND LANGUAGE

www.elsevier.com/locate/csl

Spoofing voice verification systems with statistical speech synthesis using limited adaptation data

Ali Khodabakhsh *, Amir Mohammadi, Cenk Demiroglu

Electrical and Computer Engineering Department, Ozyegin University, Istanbul, Turkey

Received 9 October 2015; received in revised form 16 August 2016; accepted 24 August 2016
Available online 5 September 2016

Abstract

State-of-the-art speaker verification systems are vulnerable to spoofing attacks using speech synthesis. To solve the issue, high-performance synthetic speech detectors (SSDs) for attack methods have been proposed recently. Here, as opposed to developing new detectors, we investigate new attack strategies. Investigating new techniques that are specifically tailored for spoofing attacks that can spoof the voice verification system and are difficult to detect is expected to increase the security of voice verification systems by enabling the development of better detectors. First, we investigated the vulnerability of an i-vector based verification system to attacks using statistical speech synthesis (SSS), with a particular focus on the case where the attacker has only a very limited amount of data from the target speaker. Even with a single adaptation utterance, the false alarm rate was found to be 23%. Still, SSS-generated speech is easy to detect (Wu et al., 2015a, 2015b), which dramatically reduces its effectiveness. For more effective attacks with limited data, we propose a hybrid statistical/concatenative synthesis approach and show that hybrid synthesis significantly increases the false alarm rate in the verification system compared to the baseline SSS method. Moreover, proposed hybrid synthesis makes detecting synthetic speech more difficult compared to SSS even when very limited amount of original speech recordings are available to the attacker. To further increase the effectiveness of the attacks, we propose a linear regression method that transforms synthetic features into more natural features. Even though the regression approach is more effective at spoofing the detectors, it is not as effective as the hybrid synthesis approach in spoofing the verification system. An interpolation approach is proposed to combine the linear regression and hybrid synthesis methods, which is shown to provide the best spoofing performance in most cases.

Keywords: Statistical speech synthesis; Hybrid speech synthesis; Spoofing verification systems; Speaker adaptation; Synthetic speech detection

1. Introduction

Text-independent voice verification (VV) systems have made tremendous progress in recent years (Martin et al., 2012). Most of the currently popular systems are based on the total variability space (TVS) approach that is based on

This work is supported by TUBITAK 1001 program under Contract Number 112E160.

* Corresponding author at: Electrical and Computer Engineering Department, Ozyegin University, Istanbul, Turkey. Fax: +90 (216) 564 99 99.
E-mail address: ali.khodabakhsh@ozu.edu.tr (A. Khodabakhsh).

http://dx.doi.org/10.1016/j.csl.2016.08.004
0885-2308/© 2016 Elsevier Ltd. All rights reserved.

representing a speech signal with a low-dimensional i-vector, which is then used for verification of claimed speaker identity (Dehak et al., 2011). Performance of those systems is now acceptable for use in many real-life applications such as call centers.

Even though the speaker verification technologies have improved, they are known to be vulnerable to spoofing attacks, which is an important concern in their deployment (De Leon et al., 2012; Kinnunen et al., 2012; Wu et al., 2015a, 2015b). Moreover, improvements in the concatenative and statistical speech synthesis systems (SSS) as well as the voice conversion systems have further spurred the concerns (Wu et al., 2015a). As a result, more effective ways to attack the verification systems and protecting the system from attacks have become increasingly important areas of research (Evans et al., 2014).

One approach that is effective at spoofing attacks is voice conversion (Wu et al., 2012). In Alegre et al. (2012), Gaussian Mixture Model (GMM) based voice transformation using parallel data is found to be effective at spoofing the voice verification systems. To increase the effectiveness of the attacks, segments of speech that get high scores from the voice verification system are repeated, which can be considered as attacking with artificial data. Two countermeasures are also proposed in Alegre et al. (2012). In one approach, distributions of Gaussian components are used to detect repetitions of Gaussians in speech. In a second approach, automatic voice quality assessment tools are used to detect synthetic speech. Spoofing performance of a joint density Gaussian mixture model (JD-GMM) voice conversion system is analyzed as a function of the training data for text-dependent voice verification systems in Wu and Li (2015).

Most parametric speech codecs use minimum-phase filters since the human auditory system is assumed to be insensitive to phase (Quatieri, 2002). If such a speech codec is used during an attack, unnatural phase spectrum can be used to detect the synthetic speech as proposed in De Leon et al. (2012) and Wu et al. (2012).

Detection performance when the synthetic speech detector (SSD) is trained with different kinds of voice conversion techniques is reported in Wu et al. (2012). Besides the magnitude and phase features that rely on a single speech frame, modulation of those features over longer durations is investigated and found to be complimentary to magnitude and phase features in Wu et al. (2013).

A second approach used in spoofing is speech synthesis (Wu et al., 2015a). There are two major approaches to speech synthesis: unit selection and statistical parametric synthesis (Black et al., 2007). Unit selection synthesis requires availability of large amounts of data from the target speaker. In many real-life scenarios, the attacker attempts to spoof into the accounts of many users. For such attacks, it is impractical to collect large amounts of data for each speaker, and the attacker may only have at most few utterances from each speaker. Thus, although effective spoofing attacks can be performed with unit selection synthesis (De Leon et al., 2012), SSS is a more effective way to attack when very limited amount of data are available, since SSS can achieve rapid speaker adaptation with only a couple of utterances (Black et al., 2007; Yamagishi et al., 2009, 2010). Because spoofing attacks with only a few utterances are investigated here, SSS is used as the baseline synthesis technique.

Most of the previous spoofing literature focused on detectors that are designed to detect known spoofing techniques (Wu et al., 2015a, 2015b). However, how effective those detectors are for unknown spoofing attack types remains a serious question. Hence, speech synthesis techniques that are specifically designed for spoofing attacks should be investigated so that detectors that can generalize better and produce more secure voice verification systems can be developed.

We propose three strategies for effective spoofing attacks when limited adaptation data are available to the attacker. In the first approach, a hybrid concatenative/statistical speech synthesis method is proposed. The proposed hybrid system takes advantage of the rapid adaptation capability of the statistical systems while using the available natural speech segments from the speaker as much as possible. Even though spoofing with a unit selection system is hard to detect (Wu et al., 2015), it cannot be deployed in a limited data case as discussed above. Still, here, we show that effectiveness of the attacks can be significantly improved with the proposed hybrid approach that exploits the available units in the database while using SSS when units are not available.

In the second approach, linear regression (LR) is done to transform synthetic speech parameters closer to natural ones. Transformation matrices are learned from a speaker-independent speech database. Even though the resulting features are more natural and more effective than the hybrid approach at spoofing the SSD, they are not as effective in spoofing the verification system. To further boost its effectiveness, in a third approach, we propose an algorithm to combine the hybrid features and transformed features, which is found to be the most effective system for spoofing attacks in most cases. The proposed algorithms were tested using three state-of-the-art synthetic speech detectors (SSD).

Not only they significantly outperformed the baseline SSS system for all three SSDs, but they could also successfully spoof the verification system.

This paper is organized as follows. An overview of the investigated systems is done in Section 2. The text-independent speaker verification algorithm used in this study is described in Section 3. A background on the hybrid speech synthesis approach used here is given in Section 4, and the proposed hybrid algorithm for the limited data case is described in Section 5. The linear regression approach to spoofing and the hybrid + linear regression (HYB + LR) approaches are described in Sections 6 and 7 respectively. Synthetic speech detectors used here are described in Section 8. Experimental results are presented and discussed in Section 9. Finally, conclusion is done and future work is discussed in Section 10.

2. System overview

An overview of the proposed speaker verification system together with the SSD is shown in Fig. 1. Speech signal is first fed to an SSD. If the SSD phase is passed successfully, speaker verification is performed to verify the claimed speaker identity.

An overview of the speech synthesis system used on the attacker's side is shown in Fig. 2. Linear regression between natural and synthetic speech features is trained using a speech database that contains parallel natural and synthetic speech from many speakers. Then, the proposed hybrid unit selection/statistical speech synthesis algorithm is used to

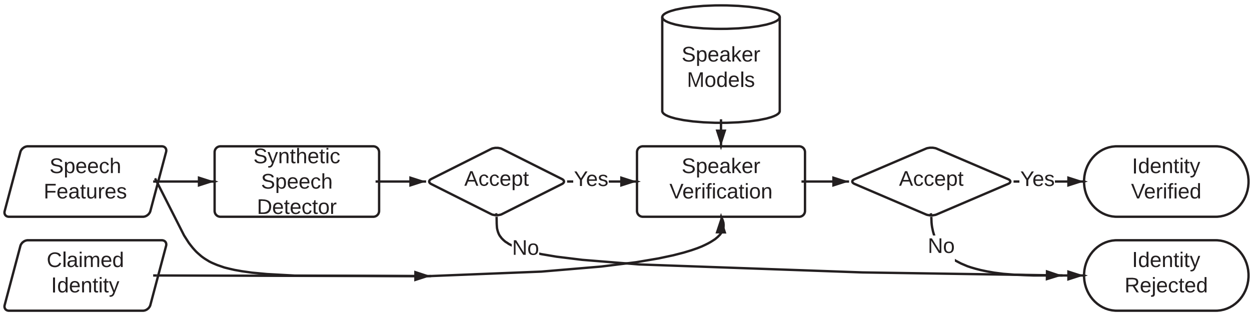

Fig. 1. Overview of the text-independent speaker verification system with the synthetic speech detector.

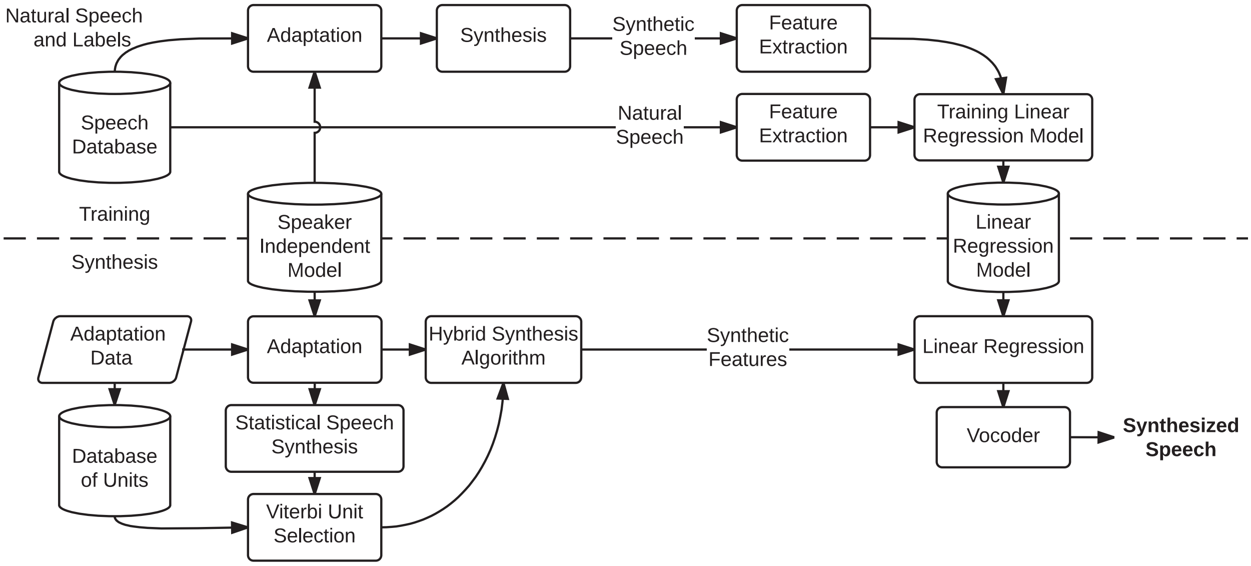

Fig. 2. Illustration of the proposed hybrid speech synthesis and linear regression algorithms. Both model training and synthesis phases are shown.

generate synthetic speech features that are transformed using the trained linear regression model. Final transformed features are then used to vocode synthetic speech.

3. Speaker verification system description

GMM is typically used to represent the acoustic feature space in speaker verification systems. In most of the current systems, a universal background model (UBM) is first trained and then speaker-specific models are obtained by adapting the UBM using a maximum a posteriori adaptation (MAP) approach.

Typically, supervector of mean vectors in UBM is very high dimensional, which increases the number of parameters to adapt. In the factor analysis approach, speaker-dependent mean vectors, $\mathbf{m}_s$, are represented in a lower dimensional eigenspace with

$$\mathbf{m}_s = \mathbf{m}_0 + \mathbf{V}\mathbf{y}_s \tag{1}$$

where $\mathbf{m}_0$ is UBM mean supervector, $\mathbf{V}$ represents the eigenvoice space, and $\mathbf{y}_s$ is a lower dimensional latent vector representing the speaker factors (Kenny et al., 2005).

Eq. (1) models the variability between speakers but it does not model the intersession variability of a given speaker. If we take the session variabilities into account, we can represent

$$\mathbf{m}_{s,h} = \mathbf{m}_0 + \mathbf{V}\mathbf{y}_s + \mathbf{U}\mathbf{x}_{s,h} \tag{2}$$

where U represents the eigenchannel space and $\mathbf{x}_{h,s}$ is the channel factor. Given an utterance from a speaker, $\mathbf{y}_s$ and $\mathbf{x}_{h,s}$ can be estimated jointly using the joint factor analysis (JFA) approach (Kenny et al., 2007).

More recently, a total variability space (TVS) approach is proposed, which combines the speaker and session variabilities in a single total variability matrix T. In the TVS approach,

$$\mathbf{m}_s = \mathbf{m}_0 + \mathbf{T}\mathbf{w}_s \tag{3}$$

where w_s is called an identity vector (i-vector). T matrix is typically trained using a database where multiple sessions are available for each speaker.

In enrollment, an i-vector is extracted from each of the enrollment utterances of a speaker. If there are more than one enrollment utterances, i-vectors extracted from each of them are typically averaged to generate a single i-vector for the speaker. In testing, an i-vector is extracted from the test utterance and compared with the i-vector computed during enrollment. Similarity comparison can be done using cosine distance scoring (CDS), support vector machines (SVM), and probabilistic linear discriminant analysis (PLDA) techniques (Dehak et al., 2011). PLDA technique is used here.

4. Hybrid speech synthesis

Although SSS creates smooth feature trajectories which eliminate the annoying glitches that are observed in the unit selection systems, the quality of speech is higher in the unit selection systems when these glitches do not occur (Black et al., 2007). Hybrid systems attempt to generate high quality speech without the glitches using a combination of unit selection and SSS approaches.

One way to create a hybrid system is using unit selection to get natural speech units while using SSS to concatenate them smoothly. It is also possible to scatter natural speech units throughout utterances while using synthetic speech for the rest of the segments. In that approach, k^{th} segment of synthetic features, $\mathbf{c}_{(k_m,k_n)}$, from frame k_m to frame k_n can be constrained to be equal to natural speech segment $\mathbf{c}_{nat,k}$ during the parameter generation process. If there are a total of K such segments scattered across an utterance, hybrid parameter generation can be formulated as the constrained optimization problem

$$\hat{\mathbf{c}}_h = \arg\max_{\mathbf{c}} p\left(\mathbf{W}\mathbf{c} \middle| \hat{Q}, \boldsymbol{\lambda}\right) \tag{4}$$

such that

$$\mathbf{A}\hat{\mathbf{c}}_h = \mathbf{c}_{nat}. \tag{5}$$

$\hat{Q}$ is the estimated hidden Markov model state sequence for the utterance and λ is the canonical models of feature distributions for the states. W is used to derive the delta and delta–delta features from the static features, $\mathbf{c}_{nat} = \left[\mathbf{c}_{(1_m,1_n)}; \mathbf{c}_{(2_m,2_n)}; \ldots; \mathbf{c}_{(K_m,K_n)}\right]$, and A is a design matrix. To perfectly generate the K natural segments, each row k of A, $\mathbf{a}_k = \left[0_{1\times(k_m-1)}\ 1_{1\times(k_n-k_m+1)}\ 0_{1\times(N_f-k_n)}\right]$ where N_f is the total number of frames in the utterance. Using the Lagrange multiplier γ, the parameter generation problem becomes

$$\hat{\mathbf{c}}_h = \arg\max_{\mathbf{c}} p\left(\mathbf{W}\mathbf{c}\middle|\hat{Q}, \boldsymbol{\lambda}\right) - \gamma\left(\mathbf{A}\mathbf{c} - \mathbf{c}_{nat}\right) \tag{6}$$

The solution to Eq. (6) is (Tiomkin et al., 2011):

$$\hat{\mathbf{c}}_h = \hat{\mathbf{c}} + \left(\mathbf{W}^T\mathbf{U}^{-1}\mathbf{W}\right)^{-1}\mathbf{A}^T\gamma$$

where $\hat{\mathbf{c}}$ is the output of the speech parameter generation without any constraints, and

$$\gamma = \left(\mathbf{A}\left(\mathbf{W}^T\mathbf{U}^{-1}\mathbf{W}\right)^{-1}\mathbf{A}^T\right)^{-1}\mathbf{c}_{nat} - \left(\mathbf{A}\left(\mathbf{W}^T\mathbf{U}^{-1}\mathbf{W}\right)^{-1}\mathbf{A}^T\right)^{-1}\mathbf{A}\left(\mathbf{W}^T\mathbf{U}^{-1}\mathbf{W}\right)^{-1}\mathbf{W}^T\mathbf{U}^{-1}\mathbf{M}.$$

$\mathbf{M} = \left[\boldsymbol{\mu}_{q1}^T, \boldsymbol{\mu}_{q2}^T, \ldots, \boldsymbol{\mu}_{qS}^T\right]$, q_i is the i^{th} observed state in the utterance, $\boldsymbol{\mu}_{q_i}^T$ is the transpose of the mean vector of state q_i repeated d_{q_i} times where d_{q_i} is the duration of state q_i. S is the total number of states in the synthesized utterance. The block diagonal matrix $\mathbf{U}^{-1} = diag\left[\mathbf{U}_{q1}^{-1}, \mathbf{U}_{q2}^{-1}, \ldots, \mathbf{U}_{qS}^{-1}\right]$ where $\mathbf{U}_{q_i}^{-1}$ is the inverse covariance matrix of state q_i repeated diagonally d_{q_i} times.

5. Proposed hybrid approach

The hybrid approach described above enforces the system to use the available state-level natural segments. In the limited adaptation case, the number of natural state-level segments in the database is very limited and there is typically at most one or two possible segments available for each state.

If the natural segments do not fit well in the context, which is highly probable in the limited data case, that can cause distortion in the neighboring frames. Not only the static features are distorted but also the velocity and acceleration features are distorted, which can further reduce the effectiveness of the attacks. To ameliorate the distortions in synthetically generated segments that are neighboring the natural segments, we propose another hybrid approach where natural features replace the statistical mean vectors in the supervector M when a natural segment exists in the database. Thus, if natural segments are available for state q_i in the unit selection database, $\boldsymbol{\mu}_{q_i}$ is modified such that

$$\boldsymbol{\mu}'_{q_i}(f) = \mathbf{c}_{nat,i}(f)$$

where $\mathbf{c}_{nat,i}$ is the selected natural unit and f is the frame index.

The duration of state q_i, d_{q_i}, is set to the duration of the natural segment $\mathbf{c}_{nat,i}$. Inverse covariance matrix of frame f, $\mathbf{U}_{q_i}^{-1}$, is formulated as follows. If the segment is longer than or equal to N_{min} frames, then

$$\mathbf{U}_{q_i}^{-1'}(f) = \frac{dist\left(f, d_{q_i}/2\right)}{d_{q_i}/2}\mathbf{U}_{q_i}^{-1}(f)$$

where $dist\left(f, d_{q_i}/2\right)$ indicates the $L1$ distance of frame f from the middle of the state. This approach allows large covariances at the boundaries, which allows the parameter generation algorithm to modify the natural segments as well as the synthetic segments more flexibly and create smooth trajectories at the boundaries. Moreover, covariances

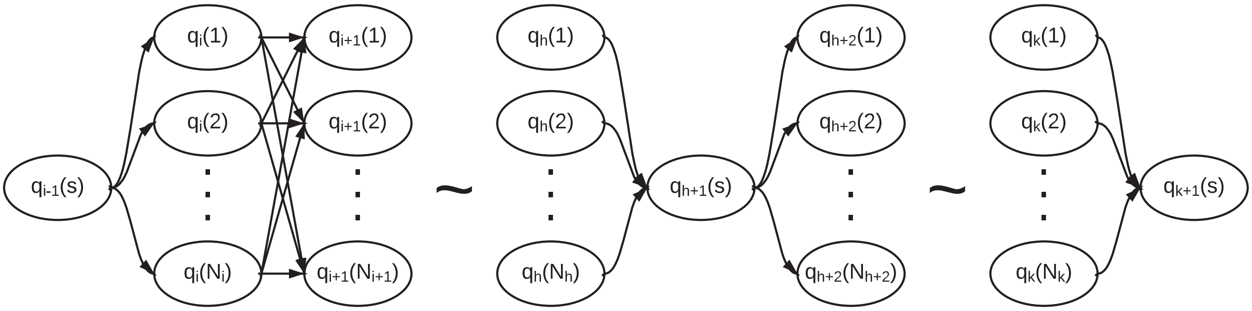

Fig. 3. Illustration of the trellis for finding the best fitting natural segments for hybrid synthesis. $(i-1)^{th}$, $(h+1)^{th}$, and $(k+1)^{th}$ states are generated with SSS. The rest of the states are generated using unit selection synthesis.

get smaller as the frames get further away from the boundary and approach to the middle of the state. Hence, the parameter generation algorithm is enforced to generate features that get closer to natural segments as the frames approach to the middle of the state and exactly pass through the natural features in the middle of the state.

If the segment is too short, then enforcing the parameters to pass through the natural frames in the middle of the state can create abrupt changes at the boundaries. To avoid the problem, if the segment is shorter than N_{min} frames, then

$$\mathbf{U}_{q_i}^{-1'}(f)=\mathbf{U}_{q_i}^{-1}.$$

which allows flexibility in parameter generation throughout all frames.

After the parameters are modified, baseline unconstrained parameter generation algorithm is used to create the parameter trajectories.

5.1. Segment selection

Even when a limited amount of adaptation data are available, more than one candidate is sometimes available for a state. For those cases, the search space is organized as a graph where each node in the graph represents either synthetic features or natural features as shown in Fig. 3. The best path with the lowest cost through the graph is selected with the Viterbi algorithm. When concatenating two segments, concatenation cost is the Euclidean distance

$$d(s_k, s_{k+1})=(\mathbf{c}_k(f_k)-\mathbf{c}_{k+1}(f_k+1))^T(\mathbf{c}_k(f_k)-\mathbf{c}_{k+1}(f_k+1))$$

where $\mathbf{c}_k(f_k)$ represents the final frame f_k corresponding to segment s_k. Similarly, $\mathbf{c}_{k+1}(f_k+1)$ represents the initial frame of the next segment $\mathbf{s}_{k+1}$.

Using the distance metric above and the Viterbi decision rule, the selected segments $\mathcal{S}$ for a given utterance is

$$\mathcal{S}=\arg\min_{\mathcal{S}}\sum_{j=1}^{N_{st}-1}d(s_j, s_{j+1})$$

where N_{st} is the total number of states in the utterance.

To use the Euclidean distance above in the selection algorithm, synthetic speech features are required before the Viterbi search. Baseline SSS parameter generation algorithm is first used to generate synthetic frames that are then used for searching for natural segments that fit best in the context.

6. Linear regression approach

Hybrid synthesis can increase the effectiveness of the attacks by increasing the similarity of synthetic and natural parameters. However, there is only few natural frames used during synthesis and the rest of the frames are generated with the parameter generation algorithm. Thus, it still has problems spoofing the synthetic speech detectors since most of the feature trajectories are generated synthetically. Hence, more effective methods are needed to spoof the synthetic speech detectors, and we propose using linear regression to transform synthetic features so that they are closer to natural feature vectors.

Let $\hat{\mathbf{c}}_s(f)$ be the output of the parameter generation algorithm at frame *f* and state *s*. The transformed features

$$\hat{\mathbf{c}}_{s,t}(f) = \mathbf{A}^{(s)}\hat{\mathbf{c}}_s(f)$$

where $\mathbf{A}^{(s)}$ is state-dependent regression matrix.

$\mathbf{A}^{(s)}$ is estimated from a speaker-independent speech database as follows. A speaker-independent (SI) speech synthesis model is first trained. Then, models for the rest of the speakers in the training set are generated using the constrained structural maximum a posteriori linear regression (CSMAPLR) speaker adaptation algorithm. To learn the relationship between synthetic and original features, all of the natural recordings from all training speakers are synthesized with SSS. Durations of the states are obtained from the natural recordings with time-alignment. Thus, durations of each synthetic and natural states match exactly.

Once parallel synthetic and natural speech utterances are generated, matching frames $(\hat{\mathbf{c}}_s(f_k), \mathbf{c}_s(f_k))$ from original and synthetic utterances are pooled together in set $S_s = \{x : x = (\hat{\mathbf{c}}_s(f_k), \mathbf{c}_s(f_k)), k = 1, 2, \ldots, N_s\}$ for each state *s*, and the transformation matrix $\mathbf{A}^{(s)}$ is estimated using the maximum-likelihood criterion

$$\hat{\mathbf{A}}^{(s)} = \arg\max p\left(S_s | \mathbf{A}^{(s)}\right)$$

7. Hybrid + linear regression approach

In experiments, hybrid approach was found to be more effective at spoofing the voice verification system and linear regression system was more effective at spoofing the detectors. Therefore, both methods can be used together for more effective attacks. In this combined approach, natural frames used in the hybrid system are not transformed. However, the rest of the synthetic frames that are generated by the parameter generation algorithm are transformed using linear regression as follows:

$$\hat{\mathbf{c}}_{hyb,lr}(f) = \alpha_f \hat{\mathbf{c}}_{hyb}(f) + (1 - \alpha_f)\hat{\mathbf{c}}_{lr}(f)$$

where $\hat{\mathbf{c}}_{hyb}(f)$ is the output of the hybrid approach at frame *f*, $\hat{\mathbf{c}}_{lr}(f)$ is its linearly transformed version, and $\hat{\mathbf{c}}_{hyb,lr}(f)$ is the combined feature vector that is found by linear interpolation. α_f is the frame-dependent interpolation factor and defined by $\alpha_f = \begin{cases} 0, & f_d \geq I \\ \dfrac{I - f_d}{I}, & f_d < I. \end{cases}$ where f_d is the distance of frame *f* to the nearest natural segment inserted by the hybrid algorithm. *I* is experimentally set to 5. Performance was not found to be sensitive to *I* as long as it is not too small ($I < 3$) or too large ($I > 10$).

Effectively, the HYB + LR algorithm uses higher weight for the hybrid algorithm as frame *f* gets closer to a natural segment and relies on the LR algorithm as the frame gets away from natural segments.

Note that the hybrid parameter generation algorithm attempts to preserve the natural segments while generating smooth trajectories. Hence, synthetically generated segments are significantly different compared to the output of the baseline SSS algorithm. The effect is higher for frames that are closer to the natural segments. Thus, the interpolation algorithm proposed here takes advantage of that by using higher weight for the hybrid algorithm for frames that are closer to the natural segments.

8. Synthetic speech detectors

The proposed algorithms are tested with three different synthetic speech detectors (SSD).

The first detector computes log-likelihood ratio (LLR) using Gaussian mixture models (GMMs) of MFCC features extracted from natural and synthetic speech. If the GMM for natural speech is denoted with $\mathbf{\Gamma}_{nat}$ and the GMM for synthetic speech is denoted with $\mathbf{\Gamma}_{syn}$, then log-likelihood ratio (LLR) given *N* observation vectors $\mathcal{O}$ is

$$\mathcal{LLR}(\mathcal{O}) = \frac{1}{N}\left(\log(\mathcal{O}|\mathbf{\Gamma}_{nat}) - \log(\mathcal{O}|\mathbf{\Gamma}_{syn})\right).$$

If $\mathcal{LLR}(\mathcal{O})$ is above a threshold ζ, $\mathcal{O}$ is classified as natural. Otherwise, $\mathcal{O}$ is classified as synthetic.

In the second detector, phased-based modified group delay (MGD) features (Wu et al., 2012) are used together with the LLR approach. Speech vocoders typically use minimum-phase filters. Because natural speech signal is not minimum-phase, phase-based features have been one of the more successful features for detecting vocoded speech. Thus, they are used here for evaluating the proposed attack algorithms.

The third detector is based on using local binary patterns (LBP), which is a technique used for texture analysis in image signal processing (Alegre et al., 2013). They have been shown to be successful in detecting unnatural distributions of local patterns in time–MFCC space. Features that are derived from the histograms of local patterns are used with a support vector machine (SVM) classifier.

9. Experiments

9.1. Experiment setup

The attacker needs significant amounts of data for training the SSS and linear regression models. Similarly, the defender needs to train the voice verification and synthetic speech detection models. Wall Street Journal (WSJ1), Resource Management (RM1), and TIMIT databases were used for training, development, and testing of all components. Table 1 shows the databases, number of speakers, and amount of data from each speaker that were used in the experiments. Tests were done with male speakers only. Details of experiment setup are described below.

9.1.1. Attacker SSS system

On the attacker side, an SI model is required for adapting to target speakers. SI model was trained using 4 speakers from the WSJ1 database with 1200 utterances from each of them. Speaker adaptive training (SAT) was used during training.

SI model was trained with 123 dimensional vectors consisting of 39 STRAIGHT features, 1 energy, 1 log Fundamental Frequency (F0) coefficient, and their delta and delta–delta features. A 25 msec analysis window with 5 msec frame rate was used for feature extraction. Phonemes were modeled with 5 state Hidden Semi-Markov Models (HSMMs).

For each enrolled speaker, different statistical models were created using adaptation with one, two, three, and four utterances. Synthesis was done for all of the 69 speakers enrolled into the verification system. Enrollment data were not used for adaptation. Experiments where 150 utterances were used for adaptation were also done for comparison purposes. CSMAPLR algorithm was used for adaptation (Yamagishi et al., 2009). Global variance (GV) algorithm was used for synthesis (Tomoki and Tokuda, 2007).

Table 1
Databases, number of speakers, and number of utterances per speaker that were used in training the text-to-speech speaker independent (TTS SI), linear regression (LR), voice verification (VV), and synthetic speech detector (SSD) systems in the attacker and defender sides.

		Attacker		Defender	
		WSJ1	TIMIT	WSJ1	RM1
TTS SI	Speakers	4	–	84	7
	Utt/spkr	1200	–	60	600
LR	Speakers	–	326	–	–
	Utt/spkr	–	10	–	–
VV	Speakers	–	–	84	101
	Utt/spkr	–	–	60	40
SSDs	Speakers	–	–	84	101
	Utt/spkr	–	–	60	40

9.1.2. Attacker hybrid system

The data available for speaker adaptation in each of the experiments were state-aligned using the HSMM SI synthesis models. Feature vectors corresponding to each observed state were stored in a unit selection database. Those units were then used in the hybrid synthesis algorithm.

9.1.3. Attacker LR system

Linear regression models were trained using 326 speakers from the TIMIT database with 10 utterances per speaker. Speaker-adapted models were generated using 10 utterances and those models were then used to create parallel synthetic and natural utterances. Only the static features were transformed. Delta and delta–delta features were computed after transformation.

There were not enough data to learn linear regression matrices for each state. A minimum of 400 frames were used for learning the LR matrices. Out of 7907 states, 5057 states had more than 400 frames. For states with less data, LR approach was not used. We have also found that the performance of LR does not improve when more 1000 frames are used in training the LR matrices. Thus, to reduce the computational load, a maximum of 1000 frames were used in the LR training stage.

9.1.4. Defender voice verification system

The voice verification system was trained with 101 speakers from the RM1 database with 40 utterances per speaker and 84 speakers from the WSJ1 database with 60 utterances per speaker as shown in Table 1.

Verification system used 19 Mel-Frequency Cepstral Coefficients (MFCC) features together with their delta and delta–delta features. Static energy feature was not used but its delta and delta–delta features were used. A 512 mixture UBM was trained and the rank of the T matrix in Eq. (3) was set to 400. Dimensionality of i-vectors was first reduced to 200 using LDA. PLDA was then used for scoring. The rank of the speaker matrix was set to 100 for PLDA.

9.1.5. Defender SSS system

Similar to the attacker, an SI model is needed for generating synthetic speech to train the synthetic speech detectors (SSDs). SI model was trained using 7 speakers from the RM1 database with 600 utterances per speaker and 84 speakers from the WSJ1 database with 60 utterances per speaker. Speaker adaptive training (SAT) was used during training.

Two different speech synthesis systems were developed for the defender side. The first system was matched to the system of the attacker and used the same set of speech features described in Section 9.1.1. To test the performance of the SSD under mismatched conditions, the second system used 25 Mel-generalized cepstrum (MGC) coefficients as opposed to the STRAIGHT-based features used by attacker. GV was used in both cases during synthesis.

9.1.6. Defender SSD systems

Detectors discussed in Section 8 were used. Radial basis function (RBF) kernel is used for SVM. Five hundred twelve Gaussians were used to model the natural speech and synthetic speech.

Synthesized versions of the test data used for testing the verification system were used to assess the performance of the SSDs. The same MFCC features that were used at the voice verification system were used for SSDs.

9.1.7. Development and test setup

Decision thresholds of the speaker verification system and the SSD system were tuned using the development data. For the speaker verification system, 100 utterances were used for client tests and 68 × 100 utterances were used for impostor tests for each enrolled speaker.

For tuning the SSD, 100 natural utterances per speaker were used for client tests and 680 synthetic utterances per speaker were used for spoofing tests. Synthesis was done using the SSS developed on the defender side. Details of the development data are shown in Table 2.

In tests, 69 speakers from the WSJ1 database were enrolled into the system using 1 utterance from each speaker. Each enrollment utterance was around 4–6 seconds long. For each enrolled speaker, 59 client tests and 408 impostor

Table 2
Number of speakers and utterances in the development and test sets that were used for the evaluation of SSD and voice verification systems.

	Development	Test
Target speakers	69	69
Genuine trials	6900	4071
Impostor trials	46 920	28 152
Spoofed trials	46 920	28 152

tests were done to test the performance of the base system. Impostor tests were created by using 6 utterances from each of the 68 impostor speakers among the enrolled speakers. Details of the test data are shown in Table 2.

In spoofing attack tests, for each enrolled speaker, 59 client tests were done where natural speech from the true speaker was presented to the speaker verification system. Each enrolled speaker was tested with 408 synthetic utterances for each adaptation data size.

9.2. Results and discussion

9.2.1. Performance of the synthetic speech detectors (SSDs)

In the first set of experiments, performance of the SSDs was measured using the proposed synthesis systems with small amounts of adaptation data. For comparison, performance was also measured when 150 utterances were available. Results are reported for both matched and mismatched conditions in Table 3. In the matched case, both SSDs and the attacker use STRAIGHT features (Kawahara et al., 1999). In the mismatched case, synthetic speech that was used to train the SSDs was generated with MGC features (Tokuda et al., 1994) while the attacker used STRAIGHT features for synthesis.

For the matched condition, in general, MFCC-based SSD was found to be effective in detecting synthetic speech generated with SSS as shown in Table 3. The performance is initially lower with the SI model, increases with increasing data size and starts decreasing again after 3 utterances. Hence, the MFCC-based SSD was found to have a generalization problem across adaptation sizes. The SSD cannot work well when the samples are too specific to a target speaker (4utt and above) or when the samples are generated with an SI model.

Hybrid system was more effective than the SSS system at spoofing as expected. Significant improvement was obtained when more than 2 utterances are available to the hybrid system. The LR system, however, significantly boosted the spoofing performance even when only one utterance was available. Because the LR algorithm modifies the SSS-generated samples, its performance with increasing data size followed the same pattern observed with the SSS system.

Performance of the HYB + LR algorithm is substantially higher than all other spoofing algorithms for the MFCC-based SSD. Significant performance improvement was obtained compared to SSS and hybrid algorithms even with one utterance. Fig. 4 shows how log-likelihood ratio distribution of synthetic speech approaches to natural speech distribution with the LR and HYB + LR approaches.

The MGD-based SSD does not work as well as the MFCC-based SSD for the SSS system except for the 150utt case. Even though similar pattern in performance was observed with increasing adaptation data size, MGD-based SSD generalized better than the MFCC-based SSD for the SSS system across different adaptation data sizes. Moreover, it performed well for the hybrid system.

Performance of LR is lower with the MGD-based SSD compared to the MFCC-based SSD. Even though LR-generated feature vectors were closer to natural than synthetic features, those features did not exactly match natural or synthetic features that were used for training the SSD. Thus, even though the synthesized speech after LR is minimum-phase, performance of the detector degraded due to generalization problems. Still, the MGD-based SSD was found to be substantially more robust to proposed spoofing algorithms compared to the MFCC-based SSD.

LBP-based SSD performed comparable to MFCC- and MGD-based SSDs for the SSS system. However, its performance was significantly lower than other SSDs for the proposed attack methods, which indicates that it cannot generalize well to other spoofing methods. The same effect was observed in mismatched experiments where the error rates of the LBP system were found to be unacceptably poor even for the SSS system.

Table 3
Missed detection rates of SSDs with the SSS and proposed attack methods using five different adaptation data sizes.

			SSS	HYB	LR	HYB + LR
Matched	MFCC (0.81%)	SI	0.24	X	X	X
		1utt	0.22	**0.22**	13.96	24.57
		2utt	**0.18**	**0.21**	9.79	36.25
		3utt	**0.12**	1.28	9.85	51.21
		4utt	**0.18**	3.34	10.52	63.95
		150utt	1.81	88.26	43.33	94.63
	MGD (0.80%)	1utt	0.83	0.75	**2.13**	**1.57**
		2utt	0.55	0.79	**1.36**	**1.67**
		3utt	0.44	**0.79**	**1.18**	**1.89**
		4utt	0.49	**0.79**	**0.98**	**1.46**
		150utt	**0.62**	**0.67**	**1.40**	**0.87**
	LBP (1.09%)	1utt	**0.14**	0.99	44.48	73.52
		2utt	0.22	5.11	44.76	85.62
		3utt	0.25	14.54	45.81	91.17
		4utt	0.37	27.10	47.80	94.78
		150utt	1.29	83.39	74.63	91.99
Mismatched	MFCC (0.89%)	1utt	**1.71**	**2.88**	39.62	63.22
		2utt	**1.62**	**7.11**	33.53	79.75
		3utt	**2.58**	21.78	30.11	90.45
		4utt	**3.04**	37.96	30.70	95.29
		150utt	15.78	95.02	73.18	98.74
	MGD (3.03%)	1utt	2.83	7.29	**9.74**	**11.84**
		2utt	2.47	7.36	**9.68**	**10.82**
		3utt	3.03	**7.72**	**10.11**	**9.87**
		4utt	3.91	**7.33**	**10.68**	**9.07**
		150utt	**7.06**	**18.04**	**10.40**	**16.87**
	LBP (0.13%)	1utt	89.98	97.30	100.00	100.00
		2utt	92.72	99.03	100.00	100.00
		3utt	94.91	99.64	100.00	100.00
		4utt	95.80	99.82	100.00	100.00
		150utt	87.98	100.00	100.00	100.00

In the matched case, both the synthetic speech that was used for training the SSDs and the synthetic speech that was used for attacks were synthesized using the STRAIGHT features. In the mismatched case, MGC features were used for synthesizing training data for SSDs while the attacker used STRAIGHT features for synthesis. In both cases, best SSD performance for each adaptation data size is shown in bold for each system. EERs of SSDs obtained with the development data are also shown in parenthesis. Those EER points were used to set the detector thresholds.

To further analyze the reasons behind the poor performance of LBP-based SSD under mismatched conditions, histograms of the SVM scores are shown for natural and synthetic speech for the SSS-based attacks in Fig. 5. Even though part of the poor performance can be explained with a calibration problem, substantial overlap between synthetic and natural scores indicates that the SSD cannot generalize well under mismatched conditions even if it was calibrated for those attacks.

Performance of all three SSDs degraded substantially under mismatched conditions. Even though MFCC-based SSD performed better than others for the SSS case, MGD-based SSD performed significantly better for the proposed attack types.

9.2.2. *Effect of delta features on SSD performance*

SSS systems tend to generate overly smooth trajectories even when the global variance (GV) algorithm is used. Moreover, LR algorithm works without taking the delta features into account. Therefore, the impact of delta features on the performance of the SSDs was investigated. To that end, experiments with the SSDs were performed using static features only.

Results are shown in Table 4. One of the most interesting observations is that the LR system was dramatically easier to detect with the GMM-based SSD when the delta features were not used in the SSD. The reason that the delta features are effective at improving the spoofing performance of the LR approach is as follows. Linear

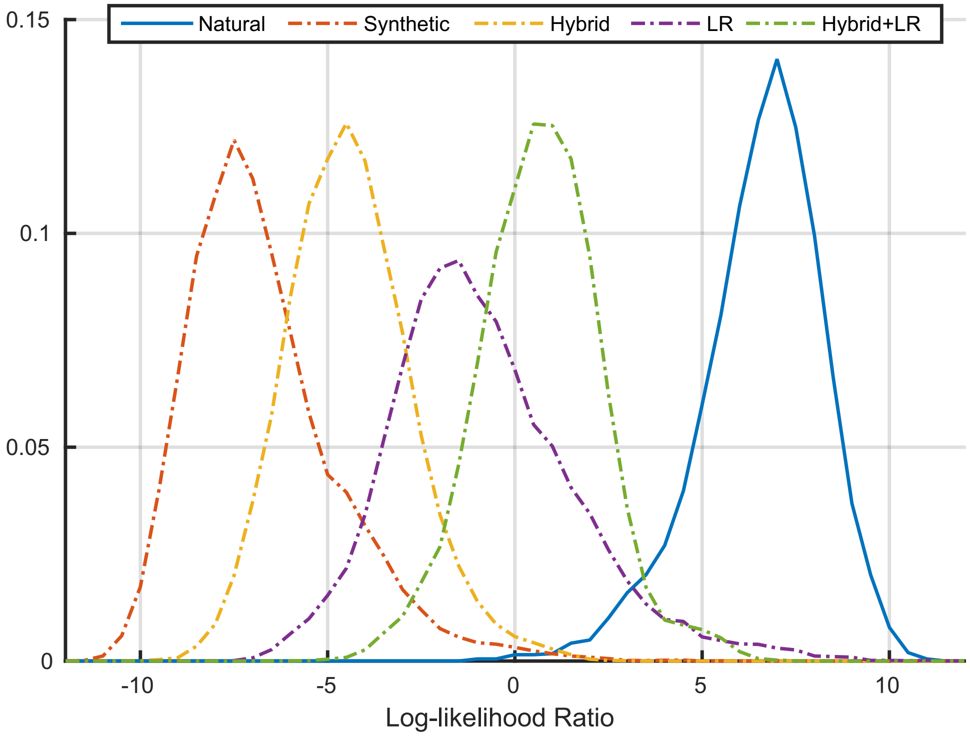

Fig. 4. Normalized histograms of log-likelihood ratio (LLR) scores for synthetic and natural utterances for MFCC-based SSD. Distributions for all synthetic types are shown for matched case. One utterance is available to the attacker.

transformations are done independently on each frame, which increases the frame-to-frame variations in the LR approach. Hence, the smooth trajectories generated by SSS are distorted and the resulting delta features get closer to natural features, which helps significantly in spoofing the SSDs. The effect was observed more with the MFCC- and LBP-based SSDs compared to the MGD-based SSD.

The second observation is that not using the static features hurt the performance of the MFCC-based SSD. This was expected since smooth trajectories make it easier to detect the SSS-based synthetic speech. In contrast to MFCC-based SSD, MGD-based SSD performed better without the delta features even for the SSS case.

9.2.3. Performance of the voice verification system

Performance of the voice verification system with natural speech is shown in Table 5. Threshold of the system during testing was set to the equal-error-rate (EER) point computed with the development data. Delta features help significantly reduce the error rates as shown in Table 5.

False alarm rates of the verification system when spoofed with synthetic speech are shown in Table 6. Since genuine trials are the same in all cases, and threshold is set with the development data, missed detections have the same values shown in Table 5 in spoofing attacks. Therefore, only the false alarm rates are presented in Table 6.

Baseline SSS system was found to have significant spoofing capability even when only one utterance is available to the attacker. Performance rapidly increased when more data became available. When the verification system used only the static features, spoofing rates of the SSS system were higher. Thus, delta features degraded the spoofing performance of SSS at the verification system.

Hybrid approach drastically increased the spoofing performance compared to SSS at all adaptation data sizes except 150 utterances. Similar to SSS, hybrid approach performed better when only static features were used. One factor behind that is the spoofing performance of SSS-generated features, which degrade with delta features. Another factor is the impact of natural segments on the neighboring SSS-generated features during the parameter generation process.

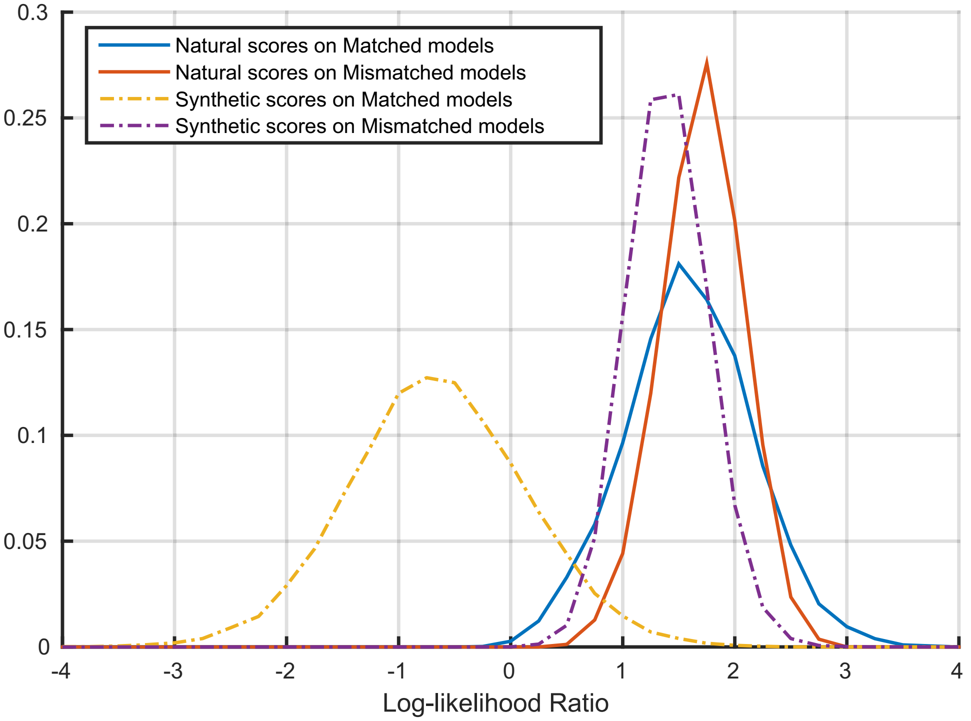

Fig. 5. Normalized histograms of soft decision SVM scores for synthetic and natural utterances for the LBP-based SSD using one utterance with the SSS spoofing method. Distributions for both matched and mismatched conditions are shown.

Table 4
Missed detection rates of SSDs with the SSS and the proposed attack methods using five different adaptation data sizes using only static features.

			SSS	HYB	LR	HYB + LR
Matched	MFCC (1.22%)	1utt	0.55	1.43	6.27	9.43
		2utt	0.27	1.63	3.78	11.91
		3utt	0.21	1.83	3.52	16.56
		4utt	0.31	2.82	3.99	22.89
		150utt	3.63	47.80	23.98	59.55
	MGD (0.65%)	1utt	**0.08**	**0.26**	**1.40**	**1.11**
		2utt	**0.10**	**0.36**	**0.88**	**1.31**
		3utt	**0.08**	**0.78**	**0.81**	**1.52**
		4utt	**0.12**	**0.42**	**0.66**	**1.11**
		150utt	**0.38**	**0.36**	**1.17**	**0.53**
	LBP (4.48%)	1utt	0.32	0.85	3.07	8.28
		2utt	0.20	1.44	2.26	10.28
		3utt	0.13	2.65	1.95	13.38
		4utt	0.31	4.71	2.80	15.37
		150utt	4.14	17.99	13.91	24.83

Hence, delta and delta–delta features were not used. Matched conditions are used where both the synthetic speech that was used for training the SSDs and the synthetic speech that was used for attacks were synthesized using the STRAIGHT features. Best SSD performance for each adaptation data size is shown in bold for each system. EERs of SSDs obtained with the development data are also shown in parenthesis. Those EER points were used to set the detector thresholds.

Smoothing at the boundaries of natural segments can distort the features close to those boundaries, which can reduce the spoofing performance when delta features were used.

LR approach also increased the spoofing performance compared to SSS when static features were used. However, its performance was poorer than SSS when delta features were used in addition to static features. Because LR only

Table 5
Performance of the voice verification system.

Statics	EER (Development)	0.70
	MD (Test)	0.74
	FA (Test)	0.29
Statics + delta	EER (Development)	0.46
	MD (Test)	0.52
	FA (Test)	0.15

Operating threshold is set to equal-error-rate (EER) point with the development data. Missed detection (MD) and false alarm (FA) rates are reported on the test data. Results are presented for two systems. One system uses only static features while the second system uses static and delta features.

Table 6
False alarm rates of the voice verification system under attack.

		SSS	HYB	LR	HYB + LR
Static	1utt	29.95	**61.49**	32.71	54.09
	2utt	35.22	**74.13**	37.37	63.72
	3utt	41.08	**80.52**	42.33	69.25
	4utt	46.49	**83.55**	47.16	72.04
	150utt	**89.90**	87.24	86.24	86.46
Static + Delta	1utt	22.99	**46.81**	22.95	37.53
	2utt	29.36	**57.13**	28.43	45.24
	3utt	35.47	**62.34**	33.86	49.53
	4utt	40.86	**63.85**	39.04	51.51
	150utt	**84.43**	78.52	77.97	77.29

Missed detection rates are shown in Table 5. Results are presented for two systems. One system uses only static features while the second system uses static and delta features. Best performing algorithm for each adaptation data size is shown in bold.

transforms the static features and does that independently for each frame, delta features can get distorted, which hurt the spoofing performance at the verification system.

Performance of the HYB + LR algorithm is significantly better than the LR algorithm but worse than the hybrid algorithm. Hence, even though LR helped significantly boost the spoofing performance at the SSD as discussed in the previous section, it also reduced the performance of the hybrid system at the speaker verification phase when used in tandem.

Spectral distortion during synthetic speech generation is highly correlated with the spoofing performance (Wu and Li, 2015). In our case, the hybrid method decreases the spectral distortion since natural speech features are used instead of the synthetic ones. Hence, the hybrid method can improve the spoofing performance at the voice verification block. Similarly, the objective function of the LR method is to reduce the distortion in the synthetic frames. However, the LR method also randomly distorts the delta features. Because of that, even though the LR method performs better than the baseline with static features, its performance degrades when delta features are used. For the same reason, HYB + LR method performs worse than the hybrid method.

9.2.4. *Performance of the combined system*

In the combined system, utterances were first processed by the SSD. Utterances that could pass the SSD were then fed to the voice verification system as shown in Fig. 1. For testing the combined system, the protocol proposed in Evans et al. (2014) was used. Using that approach, the voice verification system threshold was set to the EER point and the SSD threshold was set to fix the false alarm rates at 0.5%, 1%, and 5% using development data in three different experiments.[1] The voice verification system was set to operate at the EER point in all three cases. The SSDs and the voice verification system were both tuned using the development data.

[1] Note that synthetic utterances that are classified as natural speech at the SSD cause missed detection at the SSD. However, they cause false alarm at the voice verification system if they are verified as genuine clients.

Table 7
Performance of the combined voice verification and MFCC-based SSD systems for matched and mismatched conditions.

		Matched			Mismatched		
SSD-FA		0.50	1.00	5.00	0.50	1.00	5.00
Combined MD		0.71	1.35	5.55	0.64	1.18	4.99
SSS	1utt	0.13	0.02	0.00	0.74	0.48	0.15
	2utt	0.10	0.03	0.00	0.77	0.54	0.13
	3utt	0.02	0.02	0.01	1.46	0.99	0.27
	4utt	0.05	0.02	0.00	1.95	1.32	0.40
	150utt	2.37	1.27	0.43	16.96	12.86	5.63
HYB	1utt	0.03	0.01	0.00	0.68	0.47	0.13
	2utt	0.50	0.10	0.00	6.61	3.86	0.62
	3utt	2.01	0.68	0.03	19.27	13.23	3.94
	4utt	5.14	1.68	0.16	32.14	24.00	8.32
	150utt	**76.50**	70.74	46.42	**78.35**	**77.80**	70.89
LR	1utt	5.45	3.49	1.85	13.87	11.76	7.03
	2utt	4.18	2.33	1.20	13.83	11.37	6.48
	3utt	5.84	3.32	0.88	14.88	12.28	7.42
	4utt	7.33	4.16	0.79	17.38	14.46	8.76
	150utt	44.56	30.93	13.65	63.94	58.38	42.02
HYB + LR	1utt	**18.58**	**9.07**	**1.88**	**30.35**	**27.00**	**16.37**
	2utt	**27.78**	**15.22**	**3.60**	**41.27**	**38.13**	**26.64**
	3utt	**36.84**	**24.12**	**5.86**	**47.68**	**45.88**	**36.05**
	4utt	**43.99**	**31.88**	**9.15**	**50.69**	**49.73**	**42.65**
	150utt	76.04	**71.73**	**51.60**	77.22	76.91	**72.30**

Performance is assessed when the voice verification is set to operate at the EER = 0.46% point on the development data and SSD false alarm rates (SSD-FA) are set to 0.5%, 1%, and 5%. False alarm rates of the synthesis systems are reported for different adaptation data sizes. Missed detection rates of the combined system (combined-MD) are also reported. Best performing algorithm for each adaptation data size and SSD threshold is shown in bold.

Combined system experiments were done using the MFCC- and MGD-based SSDs because those performed significantly better than the LBP-based SSD. MFCC-based SSD was used with delta features whereas the MGD-based SSD was used without the delta features, which are the best configurations for those SSDs for the SSS-based attacks.

Results for the MFCC-based SSD are shown in Table 7. All three proposed algorithms significantly outperformed the SSS system for the MFCC-based SSD. LR was more successful than the hybrid system in the combined system particularly when the amount of adaptation data was less than four utterances. Even though the hybrid system performed better than LR at spoofing the verification system, LR performed substantially better at spoofing the SSD, which explains its higher performance in the combined system.

HYB + LR algorithm significantly outperformed the other algorithms in both matched and mismatched conditions. The false alarm rates of the verification system were unacceptably high even when the attacker has access to only one utterance from the target speaker.

MGD-based SSD performed better than the MFCC-based SSD as shown in Table 8. The proposed spoofing algorithms performed significantly better than the baseline SSS system. Under matched conditions, HYB + LR algorithm performed better than others. Under mismatched conditions, the hybrid algorithm performed the best.

The fact that only the static features were used for the MGD-based SSD significantly degraded the performance of the LR-based attacks. Moreover, LR also has lower performance at the verification stage compared to the hybrid system. Because of those two factors, LR performed significantly worse than the hybrid system in the combined mismatched tests. That also caused degradation in the performance of the HYB + LR algorithm.

10. Conclusion and future work

In this work, effective techniques for spoofing a state-of-the-art speaker verification system are proposed. Investigation of such techniques can enable the development of better SSDs and more secure voice verification systems. Therefore, as opposed to most work in the literature on SSD techniques, we focused on attack techniques.

Table 8
Performance of the combined voice verification and MGD-based SSD systems for matched and mismatched conditions.

		Matched			Mismatched		
SSD-FA		0.50	1.00	5.00	0.50	1.00	5.00
Combined MD		0.44	1.08	5.72	0.71	1.30	5.70
SSS	1utt	0.01	0.00	0.00	5.13	2.54	0.22
	2utt	0.01	0.00	0.00	5.40	2.25	0.17
	3utt	0.04	0.02	0.01	7.10	3.25	0.27
	4utt	0.09	0.05	0.02	8.60	4.46	0.60
	150utt	0.31	0.21	0.09	25.55	15.69	**2.80**
HYB	1utt	0.13	0.05	0.00	**18.43**	**11.41**	**2.52**
	2utt	0.27	0.10	0.00	**20.62**	**11.82**	**2.75**
	3utt	0.51	0.27	0.11	**23.74**	**13.93**	**3.15**
	4utt	0.35	0.12	0.00	**25.01**	**15.10**	**3.12**
	150utt	0.40	0.08	0.00	**34.03**	**20.75**	2.34
LR	1utt	0.30	0.21	**0.08**	5.86	2.85	0.29
	2utt	0.30	0.22	0.06	6.47	2.85	0.29
	3utt	0.30	0.19	0.04	8.82	4.08	0.38
	4utt	0.34	0.19	**0.07**	10.62	5.32	0.66
	150utt	**1.01**	**0.67**	**0.26**	25.27	13.17	1.13
HYB + LR	1utt	**0.42**	**0.24**	0.05	15.50	9.47	1.26
	2utt	**0.60**	**0.33**	**0.08**	17.65	10.12	1.32
	3utt	**0.69**	**0.40**	**0.13**	19.98	11.33	1.28
	4utt	**0.52**	**0.26**	0.05	20.81	12.00	1.37
	150utt	0.49	0.16	0.03	33.28	19.75	2.12

Performance is assessed when the voice verification is set to operate at the EER = 0.46% point on the development data and SSD false alarm rates (SSD-FA) are set to 0.5%, 1%, and 5%. False alarm rates of the synthesis systems are reported for different adaptation data sizes. Missed detection rates of the combined system (combined-MD) are also reported. Best performing algorithm for each adaptation data size and SSD threshold is shown in bold.

Even though the baseline SSS system was successful at spoofing the verification system, its performance dramatically dropped when an SSD was used as a countermeasure. We proposed hybrid synthesis (HYB), linear regression (LR), and their interpolation (HYB + LR) for spoofing attacks. Proposed methods not only improved the effectiveness of spoofing at the verification system compared to SSS but also they were more effective at spoofing three state-of-the-art SSDs both under matched and mismatched conditions.

10.1. Generalization issues

Mismatch of synthesis features during defense and attack significantly degraded the performance of SSDs. Moreover, performance of the SSDs fluctuated significantly with the amount of adaptation data both under matched and mismatched conditions. Those two results indicate the importance of developing detectors that not only generalize to different attack types but also generalize to adaptation data size and synthesis features.

MGD-based SSD generalized best across the three SSDs to feature mismatch, attack type, and adaptation data size. Performance of the LBP-based SSD, however, degraded dramatically under mismatch conditions and proposed attack algorithms.

10.2. Effect of delta features

Delta features were found to be useful for the MFCC-based SSD. However, the MGD-based SSD performed better when only static features were used. Thus, MFCC-based SSD was used with delta features and MGD-based SSD was used without delta features in the combined system.

Rapid frame-to-frame variations that are generated with the LR approach significantly degrade the performance of the SSDs. Those rapid variations are captured by the delta features. Thus, when the delta features are not used by the SSDs, spoofing performance of the LR approach degrades substantially. For the same reason, removing the delta features also caused significant performance degradation for the HYB + LR algorithm.

10.3. Spoofing performance

Hybrid algorithm was substantially more effective than the SSS algorithm in spoofing the verification system. LR algorithm performed comparable to the SSS algorithm, and HYB + LR algorithm performed better than LR and worse than the hybrid algorithm.

In the combined system, HYB + LR algorithm performed significantly better than all other systems for the MFCC-based SSD. This is mostly related to its high performance at spoofing the SSD with the LR and the voice verification system with the hybrid algorithms.

For the MGD-based SSD, even though HYB + LR algorithm performed the best under matched conditions, hybrid algorithm performed the best under mismatched conditions. Performance of the LR algorithm at spoofing the SSD degraded substantially when static features were not used. That resulted in lower performance both for LR and HYB + LR algorithms under mismatched conditions.

10.4. Future work

In the future work, nonlinear regression techniques, such as kernel regression, will be investigated to further boost the spoofing performance. Fused with the hybrid approach, we expect more sophisticated regression techniques to be even harder to detect and more successful at spoofing the SSD and the verification system.

References

Alegre, F., Vipperla, R., Evans, N., Spoofing countermeasures for the protection of automatic speaker recognition systems against attacks with artificial signals. In: INTERSPEECH, 13th Annual Conference of the International Speech Communication Association, 2012.

Alegre, F., Amehraye, A., Evans, N., A one-class classification approach to generalised speaker verification spoofing countermeasures using local binary patterns. In: Biometrics: Theory, Applications and Systems (BTAS), 2013 IEEE Sixth International Conference on, IEEE, 2013, pp. 1–8.

Black, A.W., Zen, H., Tokuda, K., Statistical parametric speech synthesis. In: Acoustics, Speech and Signal Processing. ICASSP, IEEE International Conference on, Vol. 4, 2007.

De Leon, P.L., Pucher, M., Yamagishi, J., Hernaez, I., Saratxaga, I., 2012. Evaluation of speaker verification security and detection of hmm-based synthetic speech. IEEE Trans. Audio Speech Lang. Process. 20 (8), 2280–2290.

Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., Ouellet, P., 2011. Front-end factor analysis for speaker verification. IEEE Trans. Audio Speech Lang. Process. 19 (4), 788–798.

Evans, N., Kinnunen, T., Yamagishi, J., Wu, Z., Alegre, F., De Leon, P., Speaker recognition anti-spoofing. In: Handbook of Biometric Anti-Spoofing, Springer, 2014, pp. 125–146.

Kawahara, H., Masuda-Katsuse, I., de Cheveigné, A., 1999. Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based f0 extraction: possible role of a repetitive structure in sounds. Speech Commun. 27 (3), 187–207.

Kenny, P., Boulianne, G., Dumouchel, P., 2005. Eigenvoice modeling with sparse training data. IEEE Trans. Speech Audio Process. 13, 345–354.

Kenny, P., Boulianne, G., Ouellet, P., Dumouchel, P., 2007. Joint factor analysis versus eigenchannels in speaker recognition. IEEE Trans. Audio Speech Lang. Process. 15 (4), 1435–1447.

Kinnunen, T., Wu, Z.-Z., Lee, K.A., Sedlak, F., Chng, E.S., Li, H., Vulnerability of speaker verification systems against voice conversion spoofing attacks: the case of telephone speech. In: Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference, 2012, pp. 4401–4404.

Martin, A.F., Yadagiri, M., Doddington, G.R., Greenberg, C.S., Stanford, V.M., The 2012 NIST speaker recognition evaluation. In: NIST SRE 2012 Workshop, 2012.

Quatieri, T.F., 2002. Discrete-Time Speech Signal Processing: Principles and Practice. Pearson Education India, Uttar Pradesh, India.

Tiomkin, S., Malah, D., Shechtman, S., Kons, Z., 2011. A hybrid text-to-speech system that combines concatenative and statistical synthesis units. IEEE Trans. Audio Speech Lang. Process. 19 (5), 1278–1288.

Tokuda, K., Kobayashi, T., Masuko, T., Imai, S., Mel-generalized cepstral analysis - a unified approach to speech spectral estimation. In: ICSLP, 1994.

Tomoki, T., Tokuda, K., 2007. A speech parameter generation algorithm considering global variance for HMM-based speech synthesis. IEICE Trans. Inf. Syst. 90 (5), 816–824.

Wu, Z., Li, H., 2015. On the study of replay and voice conversion attacks to text-dependent speaker verification. Multimed. Tools Appl. 1–17.

Wu, Z., Siong, C.E., Li, H., Detecting converted speech and natural speech for anti-spoofing attack in speaker recognition. In: INTERSPEECH, 2012.

Wu, Z., Kinnunen, T., Chng, E.S., Li, H., Ambikairajah, E., A study on spoofing attack in state-of-the-art speaker verification: the telephone speech case. In: Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012, pp. 1–5.

Wu, Z., Xiao, X., Chng, E.S., Li, H., Synthetic speech detection using temporal modulation feature. In: ICASSP, 2013.

Wu, Z., Kinnunen, T., Evans, N., Yamagishi, J., Hanilçi, C., Sahidullah, M., et al., ASVspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge. In: INTERSPEECH, 2015.

Wu, Z., Evans, N., Kinnunen, T., Yamagishi, J., Alegre, F., Li, H., 2015a. Spoofing and countermeasures for speaker verification: a survey. Speech Commun. 66, 130–153.

Wu, Z., Khodabakhsh, A., Demiroglu, C., Yamagishi, J., Saito, D., Toda, T., et al., SAS: a speaker verification spoofing database containing diverse attacks. Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), 2015b.

Wu, Z.-Z., Siong, C.E., Li, H., Detecting converted speech and natural speech for anti-spoofing attack in speaker recognition. In: INTERSPEECH, 2012.

Yamagishi, J., Kobayashi, T., Nakano, Y., Ogata, K., Isogai, J., 2009. Analysis of speaker adaptation algorithms for HMM-based speech synthesis and a constrained SMAPLR adaptation algorithm. IEEE Trans. Audio Speech Lang. Process. 17 (1), 66–83.

Yamagishi, J., Usabaev, B., King, S., Watts, O., Dines, J., Tian, J., et al., 2010. Thousands of voices for HMM-based speech synthesis-analysis and application of TTS systems built on various ASR corpora. IEEE Trans. Audio Speech Lang. Process. 18 (5), 984–1004.