SPOOFING AND ANTI-SPOOFING TECHNIQUES FOR TEXT-INDEPENDENT SPEAKER VERIFICATION SYSTEMS

A Thesis

by

Ali Khodabakhsh

Submitted to the
Graduate School of Sciences and Engineering
In Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in the
Department of Computer Science

Özyeğin University
October 2015

Copyright © 2015 by Ali Khodabakhsh

SPOOFING AND ANTI-SPOOFING TECHNIQUES FOR TEXT-INDEPENDENT SPEAKER VERIFICATION SYSTEMS

Approved by:

Professor Cenk Demiroğlu, Advisor
Department of Electrical and Electronics Engineering
Özyeğin University

Professor Arzucan Özgür
Department of Computer Engineering
Boğaziçi University

Professor Murat Şensoy
Department of Computer Science Engineering
Özyeğin University

Date Approved: 02 October 2015

To my family and friends.

ABSTRACT

There has been substantial progress in the speaker verification field in recent years. I-vector based approach in particular received significant attention due to its high performance. Improvements in the verification technology also led to concerns about spoofing attacks to which the i-vector based methods are vulnerable. Here, we first 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. However, it is well-known that speech that is generated with SSS is easy to detect using features that are extracted from the magnitude or the phase spectrum [1]. Therefore, for more effective attacks, 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 statistical synthesis method. Moreover, proposed hybrid synthesis makes detecting synthetic speech more difficult 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. An interpolation approach is proposed to combine the regression and hybrid synthesis methods which is shown to provide the best spoofing performance. Furthermore, we investigated the effectiveness of spoofing attacks with statistical speech synthesis systems when there is additive noise. Experiment results show that the attacks get substantially more effective when noise is added to synthetic speech. We also propose a synthetic speech detector that uses session differences in i-vectors to detect between synthetic and natural speech. We experimentally show that the detector has less

than 0.5% total error rate in most cases for the matched noise conditions. As a third contribution, we present our participation in generation of the first version of speaker verification spoofing and anti-spoofing database, named SAS corpus. The corpus includes nine spoofing techniques, two of which are speech synthesis, and seven are voice conversion. Two protocols were designed, one for standard speaker verification evaluation, and the other for producing spoofing materials. Hence, they allow the speech synthesis community to produce spoofing materials incrementally without knowledge of speaker verification spoofing and anti-spoofing. To provide a set of preliminary results, we conducted speaker verification experiments using two state-of-the-art systems. Without any anti-spoofing techniques, these two systems are extremely vulnerable to the spoofing attacks implemented in our SAS corpus. This work later gave birth to the first automatic speaker verification spoofing and countermeasures challenge. In our participation in this challenge, we investigated three algorithms that weigh likelihood-ratio scores of individual frames in Gaussian mixture model based detectors, phonemes, and sound-classes depending on how much information they carry. The proposed methods learn to detect both short-time and long-time artifacts which make them more reliable compared to a baseline system that treats all frames and phonemes with equal weight. Significant improvement over the baseline system has been obtained for known attack methods that were used in training the detectors. However, improvement with unknown attack types was not substantial.

ÖZETÇE

Son yıllarda konuşmacı doğrulama alanında önemli ilerleme olmuştur. I-vectöre dayalı yaklaşım yüksek performansı nedeniyle dikkat çekmiştir. Doğrulama teknolojisindeki gelişmeler ayrıca savunmasız i-vektöre dayalı metotlara karşı spoofing (yanıltıcı) saldırılara dair kaygılara yol açtı. Burada, biz ilk olarak saldırganın hedef konuşmacıdan elde ettiği limitli veri ile istatiksel ses sentezi (İSS) yöntemine karşı i-vektörüne dayalı doğrulama sisteminin savunmasızlığını inceledik. Ancak, bilindiği gibi İSS yönteminden elde edilen konuşmanın anlaşılması genlik ve faz spektrumdan elde edilen karakteristikler ile mümkündür [1]. Bu yüzden, biz daha etkili saldırılar için, hibrid istatiksel/birleştirmeli sentezleme tasarladık ve hibrid sentezlemenin istatiksel yönteme göre doğrulama sistemlerinde yanlış alarm oranını önemli ölçüde arttırdığını gösterdik. Ek olarak, tasarlanan hibrid sentezleme, orijinal sesten alınan verilerin az olması durumunda bile sentetik sesin anlaşılmasını daha zor hale getiriyor. Saldırının etkinliğini ilerletmek için sentetik karakteristikleri daha doğal karakteristiklere dönüştüren doğrusal regresyon yöntemi tanımladık. En iyi performansı sağladığı gösterilen ara değer kestirimi yaklaşımı regresyon ve hibrid sentezleme yönteminin birleştirilmesiyle tasarlandı. Ayrıca, fazladan gürültü eklendiğinde istatiksel ses sentezi ile yapılan yanıltıcı(spoofing) atakların etkinliğini inceledik. Deney sonuçları sentetik sese gürültü eklendiğinde atakların önemli ölçüde daha etkili olduğunu göstermiştir. Ayrıca i-vektörler içinde session farkı kullanarak sentetik ve doğal sesi ayırt eden bir sentetik ses detektörü tasarladık. Bir çok durumda gürültülü koşullarda detektörün 0.5%'den daha düşük hata oranı aldığını deneysel yöntemle gösterdik. Üçüncü katkı olarak, SAS kütüphane adı verilen konuşmacı doğrulama ve yanıltmaya karşı koruma veritabanının oluşumunda yer aldık. Kütüphanede ikisi ses sentezi ve yedisi

ses dönüşümü teknikleri olmak üzere dokuz adet yanıltma tekniği içeriyor. standart konuşmacı doğrulama değerlendirme ve yanıltma gereçleri için iki farklı protokol dizayn edildi. Bu yüzden, konuşmacı doğrulama yanıltma ve yanıltmaya karşı koruma bilgisi olmadan ses sentezi topluluğunun aşamalı olarak spoofing (yanıltıcı) gereçlerin üretmelerine izin verildi. Ön sonuçları sağlamak için en gelişmiş sistemleri kullanarak iki farklı konuşmacı doğrulama deneyi yürüttük. Herhangi bir yanıltmaya karşı koruma tekniği uygulanmadığında, bu iki sistem SAS veritabanı kullanılarak yapılan yanıltıcı ataklara karşı son derece savunmasızdır. Bu çalışma daha sonra ilk otomatik konuşmacı doğrulama yanıltma ve karşı önlem challenge doğmasını sağlamıştır. Biz bu challenge katıldığımızda Gaus karışım modeline dayalı detektörlerin içindeki her bir çerçevenin, bölümün olabilirlik oranı skorlarının ağırlığı, birim sesler ve ses dosyalarının ne kadar bilgi taşıdığıyla ilgili 3 algoritma araştırdık. Bu metotlar kısa ve uzun yapay kısımları belirliyor ve bu olay bu metotları bütün bölümlerin ve birim seslerin eşit etkide olduğu bazal sistemden daha güvenilir yapıyor. Detektörlerin öğrenme aşamasında kullanılan atak yöntemleri bilindiğinde bazal sistemde önemli gelişme elde edildi. Fakat, yabancı ataklara çeşitlerine karşı bir gelişme mevcut değil.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude towards my supervisor, Professor Cenk Demiroğlu, for his full support, expert guidance and encouragement throughout my study and research and mentoring me through the learning process of this master thesis. Without his incredible patience and timely wisdom and counsel, my thesis work would have been a frustrating and overwhelming pursuit. I also would like to thank my thesis committee members Professor Arzucan Özgür and Professor Murat Şensoy for spending their precious times. I would also like to thank my fellow graduate students and my past and present friends in the Speech Processing laboratory who helped me throughout this academic exploration and made this journey more fun. I would also like to give my special thanks to my family, and specially my partner in life, Zahra Ghodrati, for their unconditional love and support. I am also grateful to the following former or current staff at Özyeğin University for giving us an opportunity to work with wonderful people in a warm and friendly environment.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Text-independent voice verification (VV) systems have made tremendous progress in recent years [2]. Most of the currently popular systems are based on the total variability space (TVS) approach that is based on representing a speech signal with a low-dimensional i-vector which is then used for verification of claimed speaker identity [3]. Performance of those systems are 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 [1, 4, 5, 6]. Moreover, improvements in the concatenative and statistical speech synthesis systems (SSS) as well as the voice conversion systems have further spurred the concerns [1]. As a result, more effective ways to attack the verification systems and protecting the system from attacks have become increasingly important areas of research [7].

Despite many efforts on development of effective anti-spoofing methods, the absence of a standard database has resulted in a diverse set of individual spoofing databases, none of which is helpful for developing generalized countermeasures. This makes comparisons across different spoofing approaches difficult, and generalized countermeasures cannot readily be developed or evaluated using these databases.

1.1 Spoofing

Effectiveness of SSS approach has been shown in large-scale experiments in [1]. To further improve its effectiveness and make it harder to detect, we propose three strategies. We first propose a hybrid concatenative/statistical speech synthesis method for

spoofing attacks on verification systems when limited adaptation data is available. 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. We show that effectiveness of the attacks can be significantly improved with the proposed hybrid approach.

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 addition, we investigated the possibility of attacking the system by intentionally adding noise to synthetic speech with the hypothesis that noise can reduce the smoothness of synthetic speech and make it more difficult to detect. Noises at and above 10dB are added to synthetic speech because utterances at those signal to noise ratio (SNR) values are expected to be common in real-life. We have found that the attacks get substantially more effective when noise is added to synthetic speech even when the verification system is trained with matched noise conditions.

1.2 Anti-spoofing

One of the biggest obstacles in deployment of speaker verification technology in real-life scenarios, especially in high-security applications such as telephone banking, is the difficulty in countering spoofing attacks. Even though verification of speaker identity through human voice has been shown to be successful [2], state-of-the art verification systems have been shown to be vulnerable to spoofing attacks using speech synthesis

and voice conversion [1].

Besides showing the effectiveness of the method for attack, we propose a novel and simple synthetic speech detector that uses session differences in i-vectors to detect between synthetic speech. We then experimentally show that the proposed detector has error rates less than 0.5% in all test conditions. To make the problem more challenging, we used more advanced techniques such as global variance (GV) [8] and STRAIGHT vocoding [9] on the attacker side but not on the detection side. Even when there is such mismatch between training and test data, the detector is found to perform well in most cases.

Furthermore, we report our contribution in developing a standard database involving multiple varieties of spoofing attacks. We present the current spoofing and anti-spoofing (SAS) database and a preliminary set of benchmark results, for text-independent ASV. The database includes both speech synthesis and voice conversion spoofing attacks, which are two of the most accessible and effective spoofing approaches currently available [1, 7]. To improve the diversity of the data, speech synthesis techniques in two training scenarios were employed and seven voice conversion techniques in one training scenario. State-of-the-art statistical parametric speech synthesis methods were used to implement speech synthesis, while the voice conversion spoofing sets were created using one publicly-available open-source toolkit and six state-of-the-art conversion techniques.

In addition, we investigate several detectors without attack-specific prior assumptions. Our approach is only based on the assumption that long- and/or short-duration artifacts will be observed in the synthetic speech without any constraints on the type of artifacts. Artifacts that occur in stop sounds during synthesis because of their rapidly changing dynamics and sudden glitches that occur frequently with the unit selection systems are examples of short-duration artifacts. The overly-smooth parameters generated with HMM-based synthesis is an example to long-duration artifacts.

The counterspoofing algorithm should be sensitive to both types of artifacts to be effective.

We have investigated detectors that can capture both short and long-duration artifacts. The first detector was developed using an unsupervised approach where a Gaussian mixture model (GMM) is trained for natural speech and a GMM is trained for synthetic speech. After aligning each speech frame with a Gaussian, each Gaussian component is treated as an independent detector and detector scores are fused with logistic regression.

Our second method is based on designing detectors that are focused on detecting artifacts in specific phonemes. This approach can be successful at detecting phoneme-specific artifacts in synthetic speech. However, some of the phonemes are not observed frequently enough in most utterances. To reduce the data sparsity issue, broad-level sound class detectors are used in a third approach. Similar to the Gaussian approach, score fusion is done for the phoneme- and class-based methods.

All three methods performed substantially better than the baseline detector that treats all Gaussians and phonemes equally for the known attack types. However, the proposed systems did not substantially improve the baseline system for unknown attack types. Fusing the three proposed detectors further improved the counterspoofing performance both in known and unknown conditions.

1.3 Outline of This Thesis

The rest of this thesis is organized as follows. An overview of previous works on this topic is presented in chapter 2, along with a brief background on the technologies used. Chapter 3 explains proposed algorithms and presents experiments and results obtained from each. Finally chapter 4 concludes this thesis and discusses future works.

CHAPTER II

PREVIOUS WORK

2.1 Literature Review

Some of the prior methods for spoofing the SV systems and detection of spoofing attacks are described below.

2.1.1 Spoofing

One approach that is effective at spoofing attacks is voice conversion [10]. In [11], 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.

Voice conversion methods typically require significant amount of parallel data to be successful. However, in many practical cases, the attacker is required to attack the verification system with very limited amount of adaptation data to be able to spoof a large number of accounts. Statistical speech synthesis (SSS) systems are particularly suitable for such attacks since adaptation with a couple of utterances are feasible in those systems [12, 13, 14].

There are two major approaches to speech synthesis: unit selection and statistical parametric synthesis [12]. Even though unit selection synthesis is relatively harder to detect, it is also challenging to deploy in the context of spoofing since unlike the HMM-based approach that can adapt to the target with seconds of data, unit selection requires hours of training data. Thus, although effective spoofing attacks can be performed with unit selection synthesis [4], SSS is a more effective way to

attack when very limited amount of data is available, since SSS can achieve rapid speaker adaptation with only a couple of utterances [12, 13, 14].

2.1.2 Anti-spoofing

Most of the literature on the spoofing problem has focused on algorithms that were designed to counter specific types of attacks. For example, one method of synthesizing speech is the HMM-based approach where smooth speech parameters are generated and speech is synthesized with a vocoder. Even though HMM-based synthesis can successfully spoof the modern verification systems, it is also easy to detect by exploiting the unnaturally smooth trajectories of the parameters [8, 15, 16, 17].

Moreover, most parametric speech codecs use minimum-phase filters since the human auditory system is assumed to be insensitive to phase [18]. If such a speech codec is used during an attack, unnatural phase spectrum can be used to detect the synthetic speech as proposed in [4, 19]. However, in many distributed speech applications, only the spectral magnitude features are transmitted to avoid increasing the network traffic and minimize the delay. Moreover, phase from real speech can be used during synthesis which makes the phase-based approach ineffective [1, 20].

Some voice conversion systems exhibit low parameter variability across an utterance compared to natural speech and that was also exploited for detecting voice conversion [21]. Two countermeasures are also proposed in [11]. 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.

Modified speech detection performance when the synthetic speech detector (SSD) is trained with different kinds of voice conversion techniques is reported in [19]. Besides the magnitude and phase features that rely on a single speech frame, modulation of those features over longer duration is investigated in [22]. The modulation features

are found to be complimentary to magnitude and phase features in [22].

In the context of unit selection synthesis, existing counterspoofing methods typically use jumps in fundamental frequency at the concatenation points [23, 24].

Development of detectors that work well independent of the type of attack is a relatively new research area. One promising approach is to use a local binary pattern (LBP) analysis for feature extraction [25]. In that approach, a one-class classifier is trained with features derived only from natural speech. The classifier learns the spectro-temporal model of speech and can detect synthetic signals that do not fit well to that model.

There are a few attempts to design spoofing databases involving multiple varieties of spoofing attacks. In [26, 27], a spoofing database was designed based on RSR2015 [28] including both replay and voice conversion attacks. However, only a simple voice conversion technique was used. In [25], voice conversion, speech synthesis and artificial signal spoofing approaches were implemented on the NIST 2006 subset. However, only one voice conversion and one speech synthesis approach was employed, and only male speakers were included. No standard spoofing database exists that includes a diverse variety of spoofing techniques.

2.2 Speaker Verification Systems

GMM are 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 [29].

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 $\mathbf{U}$ 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 [30].

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

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

where $\mathbf{w}_s$ is called an identity vector (i-vector). $\mathbf{T}$ 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 [3].

2.3 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 [12]. 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

$$\widehat{\mathbf{c}}_h = \arg\max_{\mathbf{c}} p(\mathbf{W}\mathbf{c}|\widehat{Q}, \boldsymbol{\lambda}). \tag{4}$$

such that

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

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

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

Solution to Eq. 6 is [31]

$$\widehat{\mathbf{c}}_h = \widehat{\mathbf{c}} + (\mathbf{W}^T\mathbf{U}^{-1}\mathbf{W})^{-1}\mathbf{A}^T\gamma \tag{7}$$

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

and

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

$\mathbf{M} = [\boldsymbol{\mu}_{q_1}^T, \boldsymbol{\mu}_{q_2}^T, ..., \boldsymbol{\mu}_{q_S}^T]$, 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 $\mathbf{U}^{-1} = diag[\mathbf{U}_{q_1}^{-1}, \mathbf{U}_{q_2}^{-1}, ..., \mathbf{U}_{q_S}^{-1}]$ where $\mathbf{U}_{q_i}^{-1}$ is the inverse covariance matrix of state q_i repeated diagonally d_{q_i} times.

2.4 Synthetic Speech Detectors

A type of detectors typically used in anti-spoofing research is based on the GMM of the natural and synthetic speech features together with log-likelihood ratio (LLR) based detection. If the GMM for natural speech is denoted with $\boldsymbol{\Gamma}_{nat}$ and the GMM for synthetic speech is denoted with $\boldsymbol{\Gamma}_{syn}$, then LLR given N observation vectors $\mathcal{O}$ is

$$\mathcal{LLR}(\mathcal{O}) = \frac{1}{N}(\log(\mathcal{O}|\boldsymbol{\Gamma}_{nat}) - \log(\mathcal{O}|\boldsymbol{\Gamma}_{syn})). \tag{8}$$

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

The second detector used here is based on using the i-vectors for SSD. Given a test utterance, an i-vector is extracted using the voice verification system and the SVM-based SSD is used for verifying that the utterance is natural. This detector is explained in details in section 3.2.

2.5 The First Automatic Speaker Verification Spoofing and Countermeasures Challenge (ASVspoof 2015)

The objective of ASVspoof 2015 was to stimulate the development of novel, generalized spoofing countermeasures which are able to detect variable spoofing attacks

Table 1: Number of non-overlapping target speakers and utterances in the training, development and evaluation datasets of ASVspoof challenge.

Subset	#Speakers		#Utterances	
	Male	Female	Genuine	Spoofed
Training	10	15	3750	12625
Development	15	20	3497	49875
Evaluation	20	26	9404	184000

implemented with multiple, different algorithms. It aimed to facilitate the development of spoofing countermeasures without the inappropriate use of prior knowledge as regards specific spoofing attacks, stimulate the development of generalized countermeasures, and provide a level playing field to facilitate the comparison of different spoofing countermeasures on a standard dataset, with standard protocols and metrics.

The evaluation was based upon Spoofing and Anti-spoofing corpus (SAS) containing both genuine and spoofed speech. Genuine speech was collected from 106 speakers (45 male, 61 female) and with no significant channel or background noise effects. Spoofed speech was generated from the genuine data using a number of different spoofing algorithms. The full dataset was partitioned into three subsets, the first for training, the second for development and the third for evaluation. The number of speakers in each subset is illustrated in Table 1. There is no speaker overlap across the three subsets regarding target speakers used in voice conversion or TTS adaptation.

2.5.1 Datasets

For training and development sets, Each spoofed utterance is generated according to one of three voice conversion and two speech synthesis algorithms. The voice conversion systems include those based on (i) frame-selection, (ii) spectral slope shifting and (iii) a publicly available voice conversion toolkit within the Festvox system. Both speech synthesis systems are implemented with the hidden Markov model-based speech synthesis system (HTS). All data in the training set may be used to train

spoofing detectors or countermeasures.

The evaluation data includes a similar mix of genuine and spoofed speech. Spoofed data are generated according to diverse spoofing algorithms. They include the same 5 algorithms used to generate the development dataset in addition to others, designated as "unknown" spoofing algorithms. Being intentionally different, they try to give some insight into countermeasure performance 'in the wild, i.e. performance in the face of previously unseen attacks.

2.5.2 Performance measures

ASVspoof 2015 focuses on standalone spoofing detection. Participants should assign to each trial a real-valued, finite score which reflects the relative strength of two competing hypotheses, namely that the trial is genuine or spoofed speech. The primary metric for ASVspoof 2015 is the threshold-free equal error rate (EER), defined as follows. Let $P_{fa}(\theta)$ and $P_{miss}(\theta)$ denote the false alarm and miss rates at threshold θ:

$$\begin{aligned} P_{fa}(\theta) &= \frac{\#\{spoof\ trials\ with\ score > \theta\}}{\#\{total\ spoof\ trials\}}, \\ P_{miss}(\theta) &= \frac{\#\{genuine\ trials\ with\ score \leq \theta\}}{\#\{total\ genuine\ trials\}}, \end{aligned} \tag{9}$$

$P_{fa}(\theta)$ and $P_{miss}(\theta)$ are, respectively, monotonically decreasing and increasing functions of θ. The EER corresponds to the threshold θ_{EER} at which the two detection error rates are equal, i.e. $EER = P_{fa}(\theta_{EER}) = P_{miss}(\theta_{EER})$.

CHAPTER III

PROPOSED METHODS

3.1 Hybrid Speech Synthesis and Linear Regression

Overview of the speech synthesis system used on the attacker's side is shown in Fig. 1. Linear regression between natural and synthetic speech features are 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 generate synthetic speech features that are transformed using the trained linear regression model. Final transformed features are then used to vocode synthetic speech.

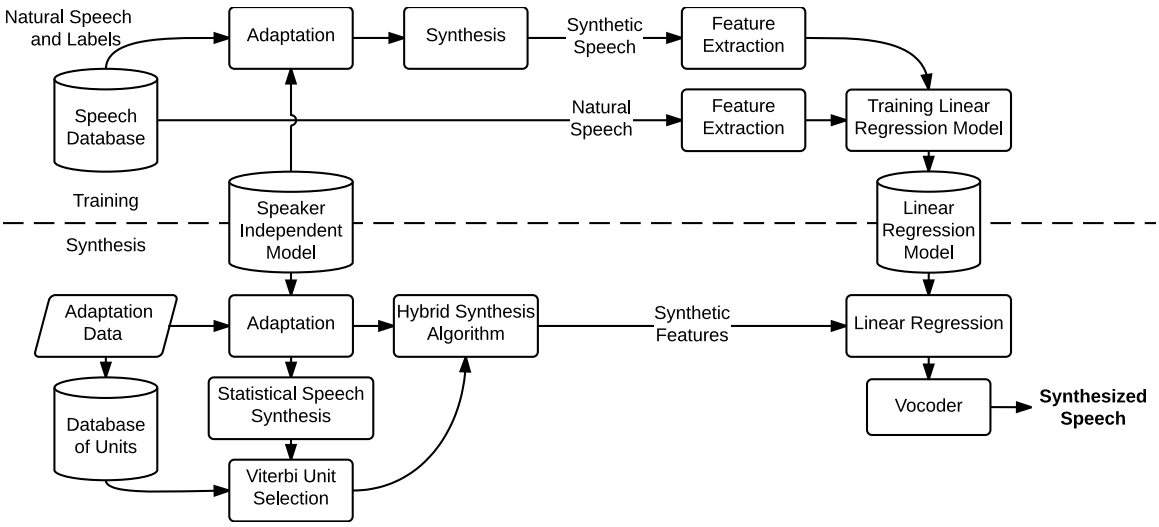


Figure 1: Illustration of the proposed hybrid speech synthesis and linear regression algorithms. Both model training and synthesis phases are shown.

3.1.1 Proposed Hybrid Approach

The hybrid approach described earlier 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 are 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 $\mathbf{M}$ 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

$$\boldsymbol{\mu}_{q_i}^{'}(f) = \mathbf{c}_{nat,i}(f) \tag{10}$$

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

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

$$\mathbf{U}_{q_i}^{-1'}(f) = \frac{dist(f, d_{q_i}/2)}{d_{q_i}/2}\mathbf{U}_{q_i}^{-1}(f) \tag{11}$$

where $dist(f, d_{q_i}/2)$ 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 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

$$\mathbf{U}_{q_i}^{-1'}(f) = \mathbf{U}_{q_i}^{-1}. \tag{12}$$

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.

3.1.1.1 Segment Selection

Even when a limited amount of adaptation data is 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. 2. 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

$$d(s_k, s_{k+1}) = (\mathbf{c}_k(f_k) - \mathbf{c}_{k+1}(f_k+1))^T(\mathbf{c}_k(f_k) - \mathbf{c}_{k+1}(f_k+1)) \tag{13}$$

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

$$\mathcal{S} = \arg\min_{\mathcal{S}} \sum_{j=1}^{N_{st}-1} d(s_j, s_{j+1}) \tag{14}$$

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.

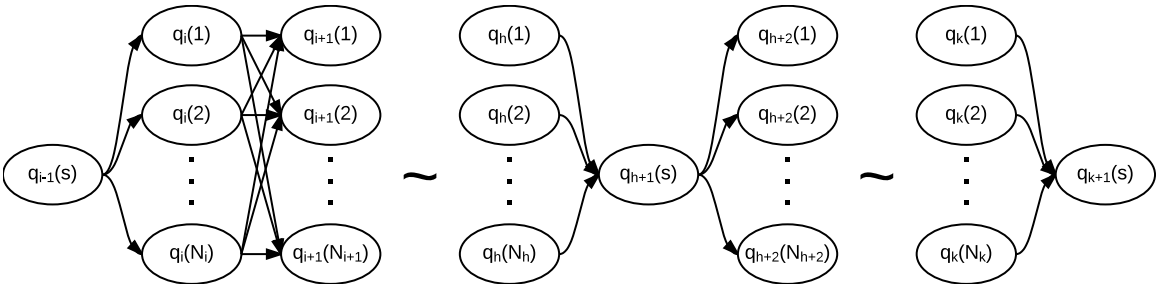


Figure 2: Illustration of the trellis for finding the best fitting natural segments for hybrid synthesis. $(i-1)^{th}$, $(h+1)^{th}$, and $(k+1)^{th}$ states are generated with SSS. Rest of the states are generated using unit selection synthesis.

3.1.2 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 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

$$\hat{\mathbf{c}}_{s,t}(f) = \mathbf{A}^{(s)}\hat{\mathbf{c}}_s(f) \tag{15}$$

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 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 $S_s = \{x : x = (\hat{\mathbf{c}}_s(f_k), \mathbf{c}_s(f_k)), k = 1, 2, ..., N_s\}$ for each state s, and the transformation matrix $\mathbf{A}^{(s)}$ is estimated using the maximum-likelihood criterion

$$\hat{\mathbf{A}}^{(s)} = \text{argmax}\, p(S_s|\mathbf{A}^{(s)}) \tag{16}$$

3.1.3 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, rest of the synthetic frames that are generated by the parameter generation algorithm are transformed using linear regression as follows

$$\hat{\mathbf{c}}_{hyb,lr}(f) = \alpha_f \hat{\mathbf{c}}_{hyb}(f) + (1 - \alpha_f)\hat{\mathbf{c}}_{lr}(f) \tag{17}$$

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 which is found by linear interpolation. α_f is the frame-dependent interpolation factor and defined by

$$\alpha_f = \begin{cases} 0 & , f_d \geq I \\ \frac{I - f_d}{I} & , f_d < I. \end{cases} \tag{18}$$

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.

3.1.4 Experiments

An overview of the proposed speaker verification system together with the SSD is shown in Fig. 3. Speech features that are derived from the short-time magnitude spectra are first fed to an SSD. If the SSD phase is passed successfully, speaker verification is performed to verify the claimed speaker identity.

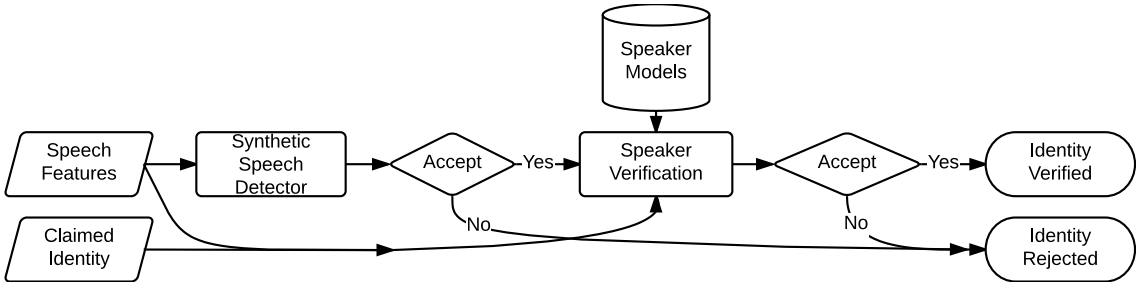


Figure 3: Overview of the text-independent speaker verification system with the synthetic speech detector.

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 2 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.

Table 2: 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

3.1.4.1 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. 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 was not used for adaptation. Experiments when 150 utterances were used for adaptation are also done for comparison purposes. CSMAPLR algorithm was used for adaptation [13]. Global variance (GV) algorithm was used for synthesis [8].

3.1.4.2 Attacker Hybrid System

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

3.1.4.3 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 was not enough data to learn linear regression matrices for each state. A minimum of 400 frames was 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.

3.1.4.4 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 2.

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. 512 mixture UBM was trained using 84 male speakers, and 60 utterances from each speaker. T matrix in Eq. 3 was trained using those same speakers and utterances. Rank of the T matrix was set to 400. Dimensionality of i-vectors were first reduced to 200 using LDA and then further reduced to 100 using PLDA.

3.1.4.5 Defender SSS System

Similar to the attacker, an SI model is needed for generating synthetic speech to train the synthetic speech detectors (SSDs). Two SI models were 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 above. 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.

3.1.4.6 Defender SSD Systems

GMM and SVM detectors were used as discussed in Section 2.4. Linear kernel is used for SVM. 512 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.

3.1.4.7 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 $\times$ 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 their synthesized versions were used for impostor tests. Synthesis was done using the SSS developed on the defender side. Details of the development data are shown in Table 3.

Table 3: 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	46920	28152
Spoofed trials	6900	4071

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 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 3.

The attacker has only 1, 2, 3, or 4 utterances available for adapting to the target speaker. For comparison, we have also considered the case where the attacker has 150 utterances for adaptation.

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 59 synthetic utterances for each adaptation data size.

3.1.5 Results and Discussion

3.1.5.1 Performance of the Synthetic Speech Detectors (SSDs)

In the first set of experiments, performance of the SSDs were 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 4. In the matched case, both SSDs and the attacker use STRAIGHT features [9]. In the mismatched case, synthetic speech that was used to train the SSDs was generated with MGC features [32] while the attacker used STRAIGHT features for synthesis.

Table 4: Equal-error-rates (EERs) of the GMM and SVM based SSDs for the proposed systems with five different adaptation data sizes. 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.

			SSS	HYB	LR	HYB+LR
Matched	GMM	1utt	0.10	**0.47**	**3.49**	**4.62**
		2utt	0.10	**0.56**	**2.33**	**6.31**
		3utt	**0.02**	**1.25**	**2.31**	**7.66**
		4utt	**0.03**	**2.14**	**1.94**	**9.90**
		150utt	**0.39**	22.11	**7.59**	22.25
	SVM	1utt	**0.05**	**0.47**	3.93	7.44
		2utt	**0.05**	1.23	2.97	8.97
		3utt	0.07	2.31	3.05	10.78
		4utt	0.05	3.19	2.63	12.77
		150utt	0.61	**21.67**	9.26	**21.62**
Mismatched	GMM	1utt	**0.29**	**2.09**	**12.04**	**19.90**
		2utt	**0.42**	**3.19**	**10.39**	**25.40**
		3utt	**1.20**	**6.78**	**11.05**	**32.15**
		4utt	**1.08**	**10.12**	**11.86**	**36.23**
		150utt	**1.11**	**29.62**	**24.32**	**29.87**
	SVM	1utt	3.37	6.68	21.54	31.17
		2utt	4.62	12.11	19.58	35.22
		3utt	4.86	16.34	20.09	38.47
		4utt	5.06	20.85	21.32	41.07
		150utt	7.91	49.18	35.37	49.03

Table 5: Equal-error-rates (EERs) of the GMM and SVM SSDs for the proposed systems with five different adaptation data sizes. Only static features are used and delta features are ignored. 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, for each adaptation data size, performance is shown in bold if the EER is lower compared to the corresponding EER in Table 4.

			SSS	HYB	LR	HYB+LR
Matched	GMM	1utt	0.32	1.45	**2.63**	**4.27**
		2utt	0.15	1.87	**1.55**	**6.21**
		3utt	0.12	2.04	**1.47**	**6.58**
		4utt	0.25	3.02	**1.50**	**8.23**
		150utt	1.67	35.69	**7.44**	35.72
	SVM	1utt	0.05	1.30	**3.68**	10.17
		2utt	0.05	2.19	**2.48**	10.66
		3utt	**0.02**	3.17	**2.16**	12.13
		4utt	**0.02**	4.27	**2.43**	13.78
		150utt	1.45	33.97	11.45	33.82
Mismatched	GMM	1utt	2.24	6.48	**6.14**	**11.20**
		2utt	2.24	5.53	**4.15**	**12.75**
		3utt	2.48	9.01	**4.86**	**15.62**
		4utt	3.19	10.86	**4.32**	**17.22**
		150utt	6.24	37.19	**15.50**	37.68
	SVM	1utt	3.59	7.20	**10.19**	**15.87**
		2utt	**4.32**	**9.73**	**9.16**	**18.03**
		3utt	**4.69**	**11.99**	**9.41**	**20.71**
		4utt	**4.74**	**14.76**	**9.95**	**23.46**
		150utt	**7.44**	**42.59**	**19.68**	**42.45**

For the matched conditions in Table 4, even though the SSS system can sometimes spoof the SSDs, its performance is substantially lower than the other systems. Moreover, spoofing performance tends to decrease with increasing data size and then increases again for the 150utt case. Similar trend is observed in the LR case. This behavior occurs because the synthetic speech that is used for training the SSDs were synthesized with speaker-adapted models and the models were trained with 40 utterances per speaker. Thus, the SSDs worked better when the amount of adaptation data used for generating the SSS models at the attacker and defender sides approached

each other.

The three proposed algorithms all performed better than the SSS system for the matched conditions. When less than 4 utterances are available, LR system performed better than the hybrid system. When enough data is available, hybrid system outperformed the LR system. HYB+LR algorithm uses both techniques, and its performance is significantly better than both algorithms.

For the matched conditions, GMM detector performed better than the SVM detector in most cases. For the SSS case where the performance of the SVM-based SSD is remarkably stable through 1utt to 4utt cases. Hence, for the limited data with SSS, SVM-based detector is found to be more robust to the adaptation data size.

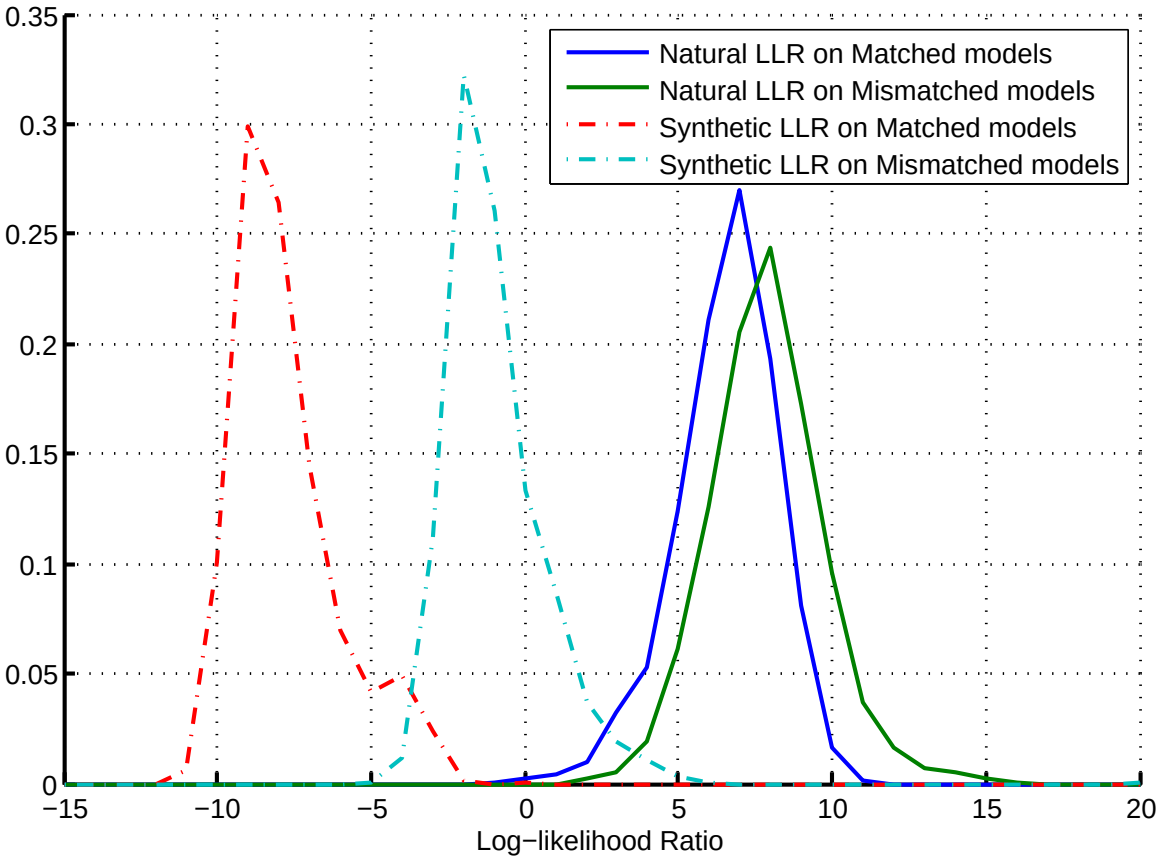
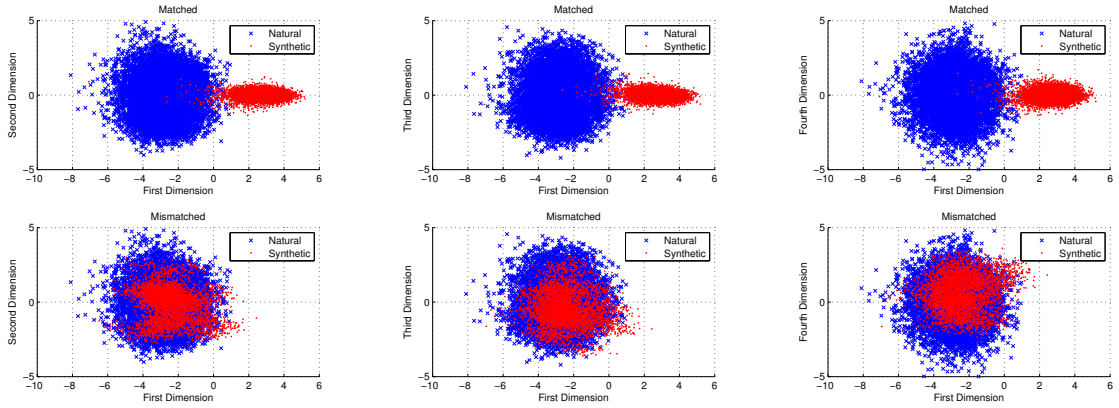


Figure 4: Normalized histograms of log-likelihood ratio (LLR) scores for synthetic and natural utterances. Distributions for both matched and mismatched conditions are shown.



(a) Scatter diagrams of first and second dimensions for matched and mismatched conditions.

(b) Scatter diagrams of first and third dimensions for matched and mismatched conditions.

(c) Scatter diagrams of first and fourth dimensions for matched and mismatched conditions.

Figure 5: Scatter diagrams of different dimensions of i-vectors, after reducing their dimensionality with LDA, for matched and mismatched conditions.

For the mismatched conditions, GMM-based SSD significantly performed better than the SVM-based SSD for all conditions. To gain insight into the reasons, histograms of the LLR scores computed by the GMM-based SSDs are shown in Fig. 4 and scatter diagrams of some of the dimensions of i-vectors after reducing their dimensionality with LDA are shown in Fig. 5. Even though distributions of natural and synthetic LLR scores approach each other for mismatched conditions in the GMM-based SSD, the overlap between them is not substantial. However, natural and synthetic speech i-vectors significantly overlap for mismatched conditions as shown in Fig. 5. Effect of channel mismatch is known to significantly degrade the i-vector performance [3]. The same effect seems to significantly degrade the SSD performance in mismatched conditions.

3.1.5.2 Effect of Delta Features on SSD Performance

SSS systems tend to generate overly smooth trajectories even when GV algorithm is used. Therefore, it is interesting to investigate the impact of delta features on the performance of the SSDs. To that end, experiments with the GMM and SVM SSDs were performed using static features only.

Results are shown in Table 5. In the matched SSS and hybrid cases, performance of the GMM-based SSD degraded which indicates the importance of the delta features. Interestingly, LR system was easier to detect with the GMM-based SSD when the delta features were missing. Because linear transformations are done independently on each frame, frame-to-frame variation increases with the LR approach. Hence, the smooth trajectories are distorted and the resulting delta features get closer to natural features which helps significantly in spoofing the SSDs.

Even though, a pattern similar to GMM-based SSD was observed with the SVM-based SSD, delta features had a lower effect in the performance of SVM-based SSD for matched conditions. Thus, SVM-based SSD was found to rely more on the static features than delta features compared to GMM-based SSD.

Behavior of the GMM-based SSD does not significantly change under mismatched conditions. However, behavior of the SVM-based SSD changes under mismatched conditions. For the SSS and hybrid cases, using delta features degrade the SSD performance when more than 1utt is available. Thus, distortion in the delta features degrade the performance under mismatched conditions with SVM-based SSD. The effect is more severe with the hybrid system compared to the SSS system.

3.1.5.3 Performance of the voice verification system

Performance of the voice verification system with natural speech is shown in Table 6. Threshold of the system during testing was set to the Equal-error-rate (EER) point computed with the development data. Delta features significantly reduce the error rates as shown in Table 6.

False alarm rates of the verification system when spoofed with synthesized speech are shown in Table 7. 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 6 in spoofing attacks. Therefore, only the false alarm rates are presented in

Table 6: Performance of the voice verification system. 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.

Statics	EER (Development)	0.38
	MD (Test)	0.59
	FA (Test)	0.39
Statics+Delta	EER (Development)	0.31
	MD (Test)	0.29
	FA (Test)	0.29

Table 7: False alarm rates of the voice verification system under attack. Missed detection rates are shown in Table 6. 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.

		SSS	HYB	LR	HYB+LR
Static	1utt	21.13	**72.49**	28.89	71.53
	2utt	26.31	88.11	32.74	**88.87**
	3utt	32.06	93.83	37.83	**93.88**
	4utt	37.53	**96.81**	43.55	96.56
	150utt	93.00	97.64	92.41	**97.69**
Static+Delta	1utt	21.98	64.48	28.91	**64.60**
	2utt	30.04	**84.65**	36.28	83.54
	3utt	35.52	**92.43**	42.08	90.62
	4utt	41.17	**96.29**	48.88	94.69
	150utt	91.50	**97.15**	93.00	97.13

Table 7.

Baseline SSS system was found to have significant spoofing capability even when only one utterance is available to the attacker. Performance rapidly increases when more data becomes available. When the verification system uses only the static features, spoofing rates of the SSS system decreases. Thus, delta features improve the similarity of feature vectors to the target speaker which is expected since the static features are generated to maximize the joint likelihood of both static and delta features. Thus, using the delta features together with the static features creates features closer to the target speaker.

Hybrid approach drastically increased the spoofing performance compared to SSS at all adaptation data sizes. Hybrid approach performed better when only static features were used. Thus, the effect of natural segments on the neighboring synthetic speech frames during the parameter generation process seem to distort the delta features around the natural segments which reduced the spoofing performance when delta features were used. However, the difference between static and static+delta cases decreased with increasing data sizes. This is expected since with more data available, hybrid approach can utilize more natural segments.

LR approach also increased the spoofing performance compared to SSS. In the case of LR, static+delta features performed better than the static features. This result is aligned with the SSD results discussed in Section 3.1.5.2 where LR approach was found to be significantly more effective at spoofing the SSD when the delta features were used.

It is interesting to note that LR system can improve the spoofing performance even though the LR transformations are not target-specific. Experimental results showed that getting feature vectors closer to natural features improves the spoofing performance even when the transformations are not target-specific.

Performance of the HYB+LR algorithm is close to the performance of the hybrid algorithm. Thus, LR did not substantially degrade the performance of the hybrid approach while it helped significantly boost the spoofing performance at the SSD as discussed in the previous section.

3.1.5.4 Performance of the combined system

In the combined system, the utterance was first processed by the SSD. Utterances that could pass the SSD were then fed to the voice verification system as shown in Fig. 3. For testing the combined system, the protocol proposed in [7] was used. 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 is 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.

Combined tests were performed using the GMM-based SSD because it outperformed the SVM-based SSD in most of the matched and all of the mismatched conditions as shown in Table 4. Both static and delta features were used in SSD because the SSD was trained for detecting SSS and using delta features improved the performance of GMM-based SSD as shown in Table 4 and Table 5. Speaker verification system also used both static and delta features.

The missed detection and false alarm rates of the combined system are shown in Table 8. For the matched test setup where the SSD was trained with STRAIGHT and the attacker use the STRAIGHT for synthesis, SSS system has %0 false alarm rate even when the SSD false alarm rate is %0.5. Even though the SSS system could spoof the combined system for the mismatched case, the false alarm rates were still substantially lower compared to the proposed systems. SSS was not found to be effective at spoofing the combined system because it cannot spoof the SSD as shown in Table 4.

Hybrid system has higher false alarm rates compared to SSS for the three operating points under the matched and mismatched conditions. Its performance is substantially higher than SSS especially in the mismatched conditions when more than 1utt is available to the attacker. Moreover, performance difference with SSS increases rapidly with increasing adaptation data size.

The LR system performed significantly better than the hybrid system when the adaptation data size is small. Even though the hybrid system is more effective in spoofing the voice verification system compared to LR system in those cases as shown

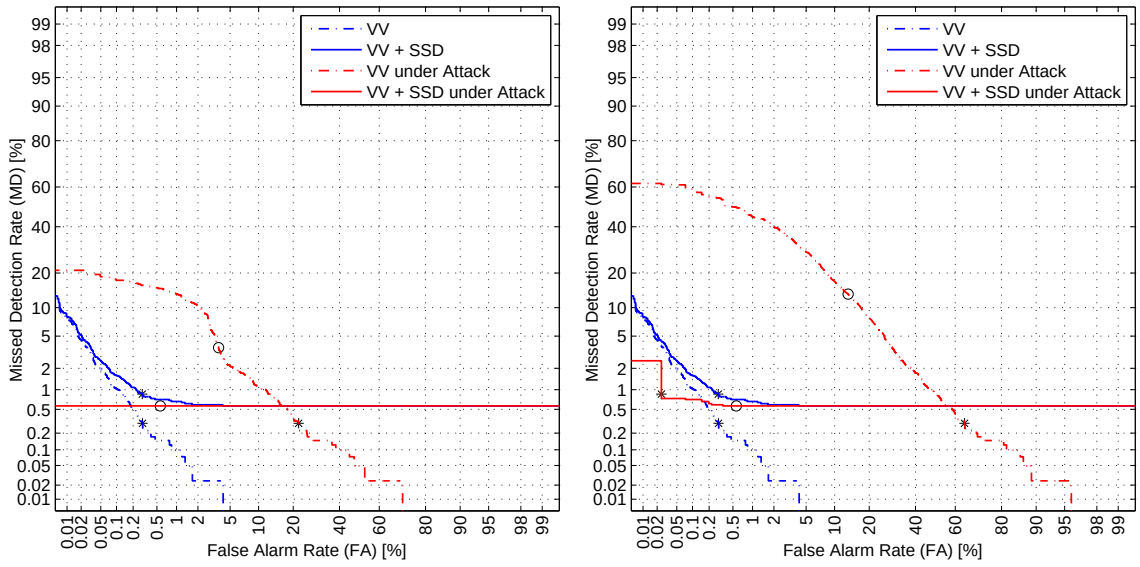
[1]Note that synthetic utterances that are verified as natural speech at the SSD cause missed detection at the SSD. However, they cause false alarm at the voice verification system if they are verified as genuine clients.

Table 8: Performance of the combined voice verification and SSD systems for matched and mismatched conditions. Performance is assessed when the voice verification is set to operate at the EER=%0.38 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.

		Matched			Mismatched		
SSD-FA		0.5	1	5	0.5	1	5
Combined MD		0.86	1.50	5.70	0.79	1.38	5.45
SSS	1utt	0.00	0.00	0.00	0.02	0.00	0.00
	2utt	0.00	0.00	0.00	0.29	0.05	0.00
	3utt	0.00	0.00	0.00	1.52	1.15	0.07
	4utt	0.00	0.00	0.00	1.30	0.52	0.00
	150utt	0.15	0.00	0.00	3.05	0.96	0.07
HYB	1utt	0.02	0.02	0.00	1.45	0.69	0.61
	2utt	0.74	0.15	0.00	16.14	8.77	1.20
	3utt	3.51	1.35	0.05	36.60	22.77	7.81
	4utt	13.19	4.62	0.54	61.24	42.10	16.34
	150utt	**94.94**	**90.69**	72.59	96.14	**93.34**	**79.07**
LR	1utt	3.34	2.33	1.28	19.38	14.71	7.05
	2utt	3.05	2.09	1.30	21.42	15.72	8.74
	3utt	4.72	2.85	0.52	20.46	15.40	10.00
	4utt	5.94	2.90	0.12	25.25	18.84	11.35
	150utt	38.91	23.73	9.38	81.60	72.91	49.47
HYB+LR	1utt	**30.09**	**15.60**	**2.63**	**57.33**	**51.27**	**35.99**
	2utt	**49.57**	**27.73**	**7.07**	**80.74**	**75.58**	**55.51**
	3utt	**66.10**	**45.62**	**12.85**	**89.76**	**86.54**	**70.74**
	4utt	**80.84**	**62.61**	**22.97**	**94.45**	**93.00**	**82.54**
	150utt	94.67	90.52	**73.18**	**96.27**	93.24	79.05

in Table 7, the success of the LR system substantially outweighs the performance of hybrid system in passing the SSD as shown Table 4. That causes the performance of the LR system in the combined results to be higher when there is minimal adaptation data. However, when more data became available, hybrid system significantly outperformed the LR system.

HYB+LR system substantially outperformed both hybrid and LR systems at all limited data sizes. The reasons for this can be inferred from the spoofing performance of the proposed methods at the SSD and the voice verification systems. To gain



(a) SSS system

(b) Hybrid synthesis system

(c) Linear Regression (LR) system

(d) HYB+LR system

Figure 6: Detection error trade-off (DET) curves of the SSS, hybrid, LR, HYB+LR systems are shown for the 1utt matched condition case. False alarm rate of the SSD is %0.5. Following the protocol in [7], DET curves for voice verification (VV) using natural speech, VV+SSD using natural speech, VV under attack using speech synthesis, VV+SSD under attack using speech synthesis are shown. On the DET curves, 'o' indicates the EER points and '*' indicates the operating points after tuning the SSD with the development data.

more insight, detection error trade-off (DET) curves for hybrid, LR, and HYB+LR systems for the 1utt case are shown in Fig. 6. HYB+LR is as good as the hybrid

system in spoofing the voice verification system. Thus, having an additional target-independent LR step did not significantly degrade the spoofing performance at the voice verification block. The substantial performance gain obtained with HYB+LR is found to be related to its spoofing performance at SSD. Not only HYB+LR has significantly higher spoofing performance at the SSD compared to other systems, but also calibration of the operating point based on an SSS system further boosts its performance. Similar calibration problems were also observed with hybrid and LR systems. However, the effect of those were not as severe as the HYB+LR case.

3.2 I-vector Based Synthetic Speech Detection

Even though removing the session effects from the i-vectors is important for successful verification, session differences contain valuable information for detecting synthetic speech. For session-i, channel vector can be defined as

$$m_{c,i} = m_{s,i} - m_s \tag{19}$$

where $m_{s,i}$ is the i-vector extracted in session-i and m_s is the mean i-vector for speaker s.

Channel vectors contain information about the distortions that are session-specific. In the case of synthetic speech, there is additional variability. For example, it is well-known that synthetic features are smoother than natural features which reduce the variance of all features [17]. Moreover, because feature vectors in close proximity are similar to each other, they are assigned to the same Gaussian. Therefore, as opposed to the variety of Gaussians in natural speech, fewer Gaussians are observed with higher frequency in synthetic speech.

We investigated the differences between i-vectors of synthetic and natural speech through visualization. To that end, Fisher linear discriminant analysis (LDA) is used to reduce dimensionality of the channel vectors to 2. Channel vectors of synthetic and natural speech is compared in Fig. 7. In the clean case, there is a clear separation

between synthetic and natural vectors. In the noisy case, the two clusters are still clearly separable. However, the margin is not as large as the clean case. Thus, noise distorts the smooth structure of the synthetic features and make clean and noisy channel less separable.

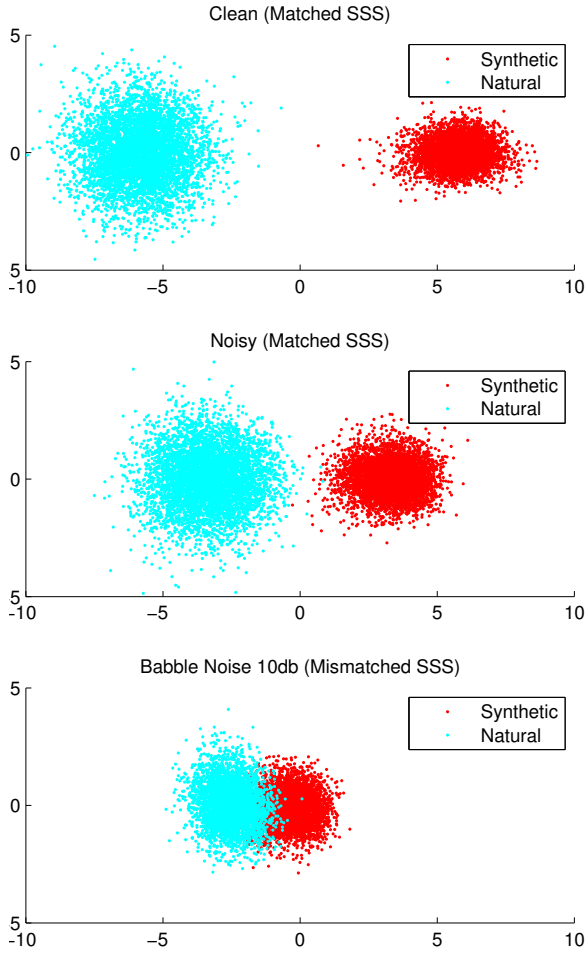


Figure 7: Illustration of channel vectors after they are mapped to 2 dimensions using LDA. In the top figure, clean synthetic and natural data is used where both test and train synthetic data are generated with STRAIGHT and GV. In the middle figure, noisy natural and synthetic data are used where both test and train synthetic data are generated with STRAIGHT and GV. Mixed type of noises are used in training LDA and channel vectors of noisy natural and synthetic speech (mixed noise) are shown. In the bottom figure, LDA is trained on noisy synthetic speech without GV and STRAIGHT but the test data are generated with STRAIGHT and GV. Effect of mismatch in synthesis technologies are shown. Mixed type of noises are used in training LDA and channel vectors of noisy natural and synthetic speech (babble noise) is shown.

Even though the clusters are separable in noisy conditions, an important question arises: what if the attacker and the defender use different SSS technologies? In particular, we are interested in the worst case where the attacker has more advanced technology compared to the defender. To test that condition, STRAIGHT vocoding and GV adjustment is used at the attacker side but not at the defender side. Clusters for synthetic and natural channel vectors at 10dB babble noise are shown in Fig 7. Using different synthesis technologies by the attacker and defender caused significant overlap between the clusters which makes the detection problem harder.

Exploiting the structure in the distribution of channel vectors, a detector is designed to detect synthetic speech. Dimensionality of session vectors are first reduced using LDA. Then, a support vector machine (SVM) with soft-decision output is trained with the noisy synthetic and noisy natural session vectors. Linear kernel is used with the SVM.

3.2.1 Experiments

WSJ1 database [33] is used for the verification experiments similar to [4]. 69 male test speakers are enrolled into the system. Each enrollment utterance is around 4-6 seconds long. For each enrolled speaker, 59 client tests and 340 impostor tests are done. Impostor tests are created by using 5 utterances from each of the 68 impostor speakers among the enrolled speakers. Each test is done using one utterance. Verification system uses 19 dimension MFCC plus 1 energy static features and their delta and delta-delta features. However, static energy is not used which makes the total dimension of features 59. 256 mixture UBM is trained using 84 male speakers, and 60 utterances from each speaker. T matrix is trained using those same speakers and utterances. Rank of the T matrix is set to 400.

Experiments are done for clean training and test data as well as noisy training and test data. Noise is added to clean speech samples at 10, 15, and 20dB SNRs because

when the SNR is below 10 dB, performance of the verification system is found to be unacceptably poor. The detector and the verification systems are trained using a mixture of white, babble, car, and station noisy samples under 10, 15, and 20dB SNRs in noisy conditions. Bus, cafe, metro, and office noises are used only during testing.

For each enrolled speaker, different statistical models are created for attacks using adaptation with one, two, three, and four utterances. Synthesis is done for all of the 69 speakers enrolled into the verification system. Enrollment and test data are not used for adaptation. Experiments when 150 utterances are used for adaptation are also done for comparison purposes. Speaker-independent (SI) model is generated using four male speakers and 1250 utterances from each speaker. Constrained structural maximum a posteriori linear regression (CSMAPLR) algorithm is used for adaptation [13].

SSS systems were trained with 198 dimensional vectors consisting of 40 Mel-Generalized Cepstral (MGC), 1 Log-Fundamental frequency (LF0), and 25 Band APeriodicity (BAP) coefficients and their delta and delta-delta parameters. 25 msec analysis window with 5 msec frame rate is used for feature extraction. Phonemes are modeled with 5 state hidden semi-Markov models (HSMM) [34]. STRAIGHT vocoding and global variance adjustments are done to improve the synthesis quality.[9]

Training data for UBM and T are used for training the detectors. The same features used in the verification system are used for the detector. Similar to the attacker, a speaker-independent (SI) model is needed for creating the synthetic speech database for training the detector. Here, SI model is trained using the training data of the verification system. Synthesized versions of the test data used for testing the verification system are used to assess the performance of the detectors under different conditions. Detector performance is reported in terms of equal-error-rate (EER) for each test condition. Dimension of the channel vectors are reduced to 50 with LDA

before using SVM for synthetic speech detection.

3.2.2 Results and Discussion

Baseline performance of the voice verification system in clean training and test conditions in terms of equal-error-rate (EER) is 0.23%. Performance of the system for individual noise types and SNRs are shown in Table 9. EER calculated under all SNRs and noise types combined is 1.81% which is almost 8-folds increase compared to clean conditions. White noise had particularly higher error rate compared to others since it distorts all of the speech spectrum.

Table 9: EER of the voice verification system for different noise types and SNRs. Verification system is trained with mixed noise conditions and SNRs. White, babble, car, and station noises were used in training of the verification system.

Seen noises	10db	15db	20db
white	4.53	1.98	1.16
babble	1.27	1.23	1.11
car	1.21	1.19	1.26
station	0.96	0.97	1.03
Unseen noises	10db	15db	20db
bus	1.27	1.24	1.22
metro	1.26	1.10	1.13
office	1.25	1.28	1.25
cafe	1.13	1.13	1.15

For spoofing attacks, threshold of the voice verification system is set to 1.81% average EER point. Results with clean train/test and noisy train/test are shown in Fig. 8. Noise substantially increases the effectiveness of the attacks. Effectiveness of car and bus noises are below others since those noise types have lower bandwidth. Interestingly, effectiveness of the attacks are close to each other at different SNRs. This is thought to be a result of the fact the system is trained with a mix of all SNRs and all noises. Moreover, the calibration is also done with a mix of all conditions. Thus, the system does not seem to substantially favor any particular SNR.

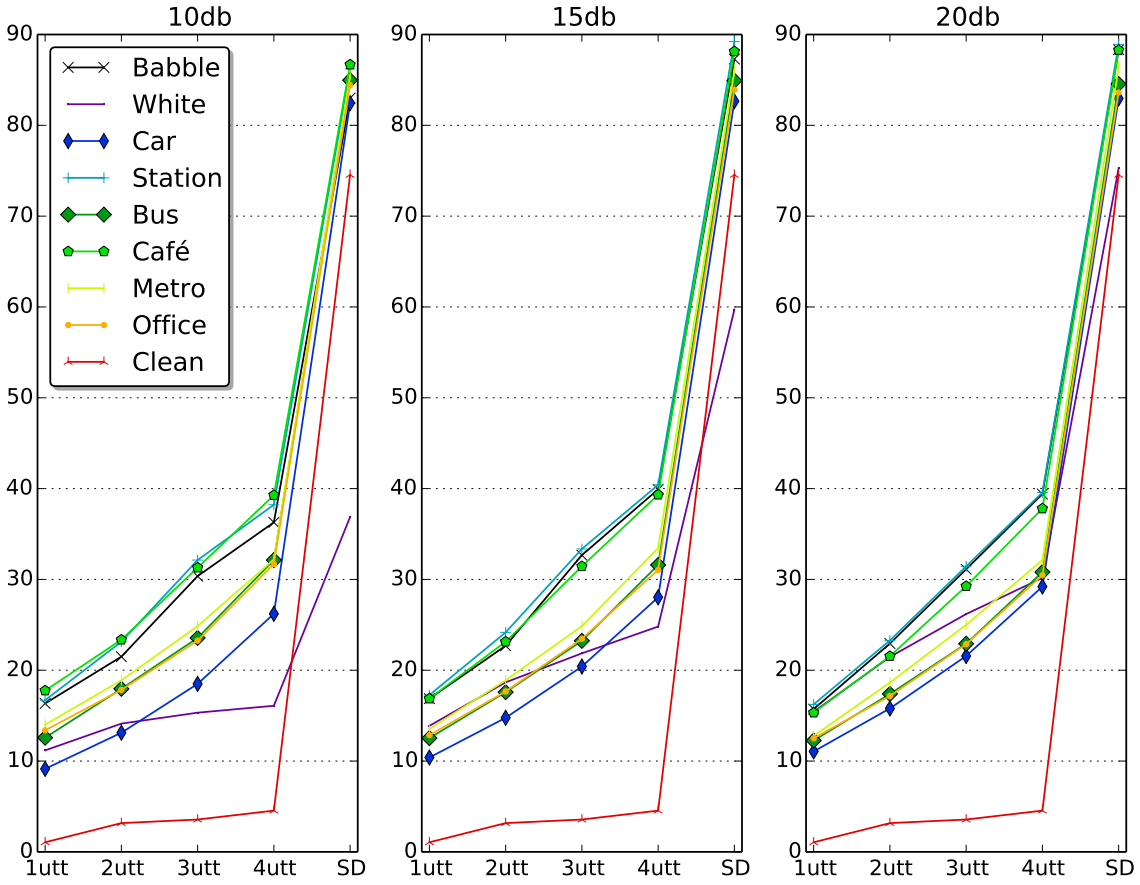


Figure 8: Verification false alarm rates under attack with synthetic speech. Results are reported for both clean and noisy conditions. In the "Clean" case, both test and train samples are clean and it is shown in the figures for comparison purposes. Babble, cafe, and station noise results have almost overlapped here. Metro, bus, and office noise results have almost overlapped here.

Spoofing attacks become more effective when more adaptation data becomes available. However, performance seems to saturate more rapidly in the clean conditions compared to noisy conditions.

White noise has especially lower false alarm rates compared to other noise types. The reason for that can be understood from Fig. 9. In that figure, at 10db, white noise detection error trade-off (DET) curve is significantly separated from the other noise types which holds for other SNR types and adaptation data sizes also. The 1.81% EER, however, is computed by using all noise conditions at all SNRs which causes an outlier effect where the white noise has a big effect on the operating point.

Thus, at the 1.81% operating point, all noises other than white noise have significantly higher false alarm rates compared to missed detection rates as shown in Fig. 9. White noise, however, does not significantly deviate from the EER point. As a result, its false alarm rate is lower than others in spoofing attacks.

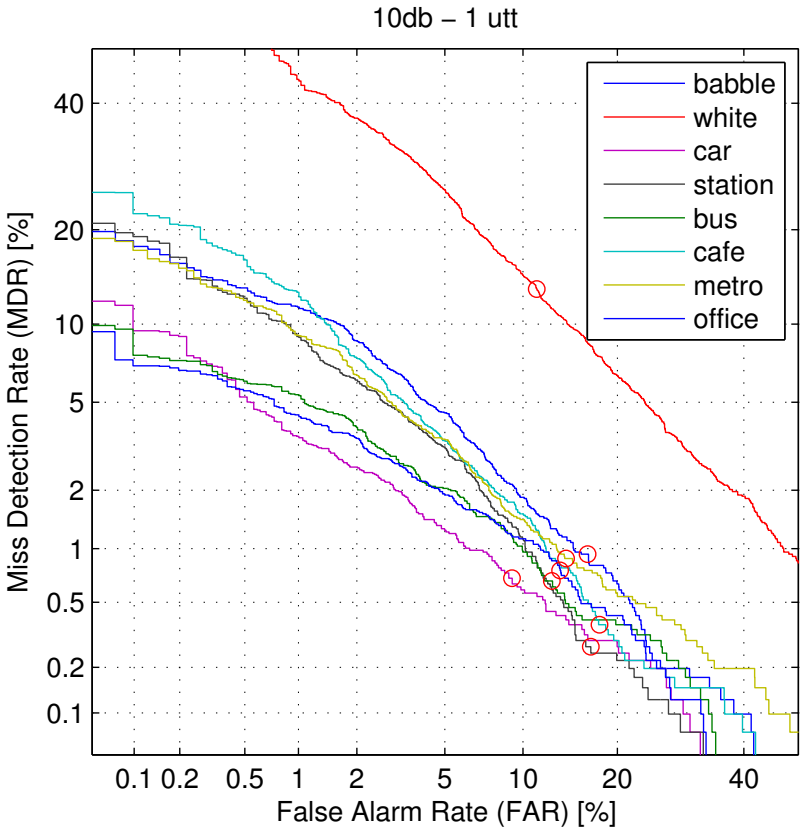


Figure 9: DET curves of the verification system under attack at different noise conditions at 10dB. Natural speech is used for clients and synthetic speech is used for impostors. Performance of the verification system for different noise types are indicated with circles when the verification system is tuned to 1.81% EER point with mixed noise conditions.

The proposed detector has 0% detection error for clean case. For noisy case, EER is less than 0.5% for all noise and SNR conditions as shown in Fig. 10. Thus, synthetic speech can be effectively detected in the i-vector space with very high accuracy as observed visually in Section 3. To check if these results still hold for mismatched SSS technologies in attacker and defense sides, the detector is trained with SSS without

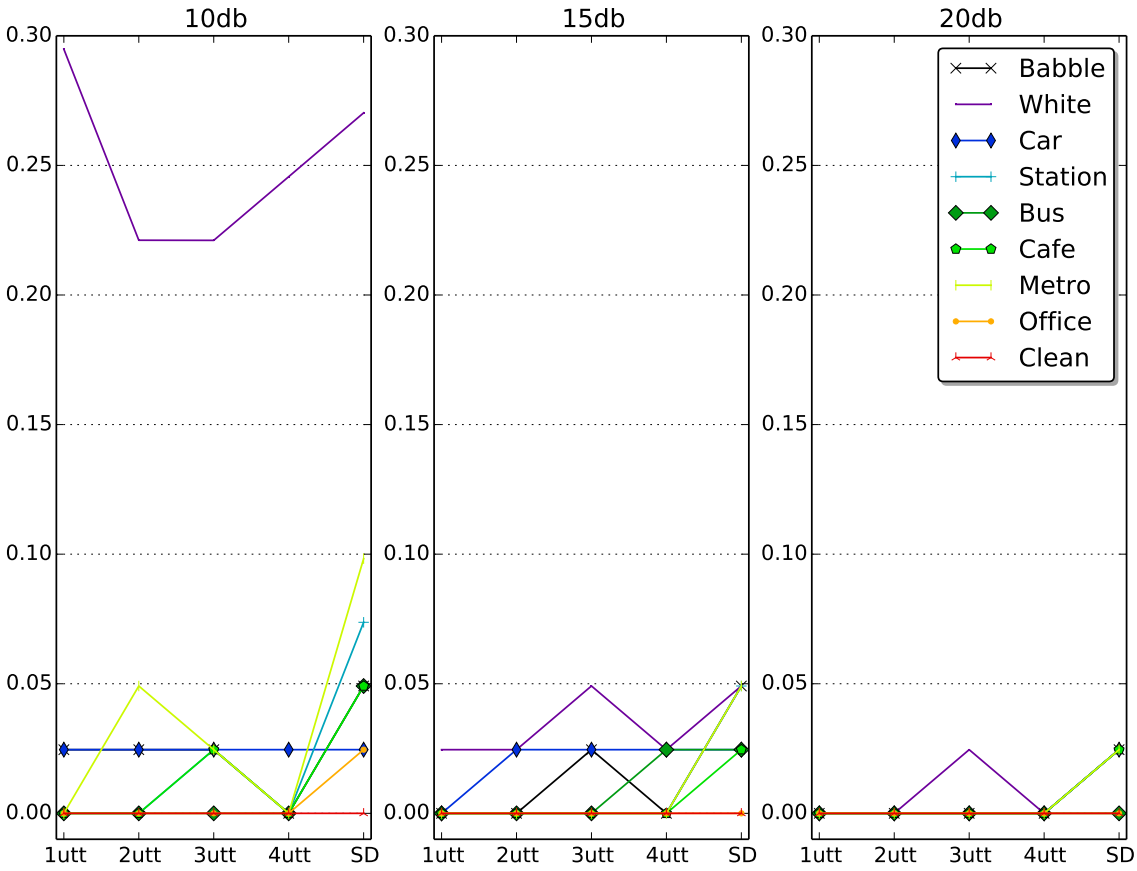


Figure 10: Detector performance (EER) when detector is trained with STRAIGHT vocoder and GV. And, the attacker uses STRAIGHT vocoder and GV as well (Matched condition in SSS). Except for white noise at 10dB, EER of all cases is under 0.1%

GV or STRAIGHT. The attacker, however, used STRAIGHT and GV which are known to increase the quality of speech. Effectiveness of the spoofing attacks in such mismatch conditions are reported in Fig. 11. Under the mismatched SSS synthesis conditions, detection performance decreases substantially especially for babble and white noises. This result calls for training detectors with different synthesis conditions and not fit the detector on one particular type of SSS.

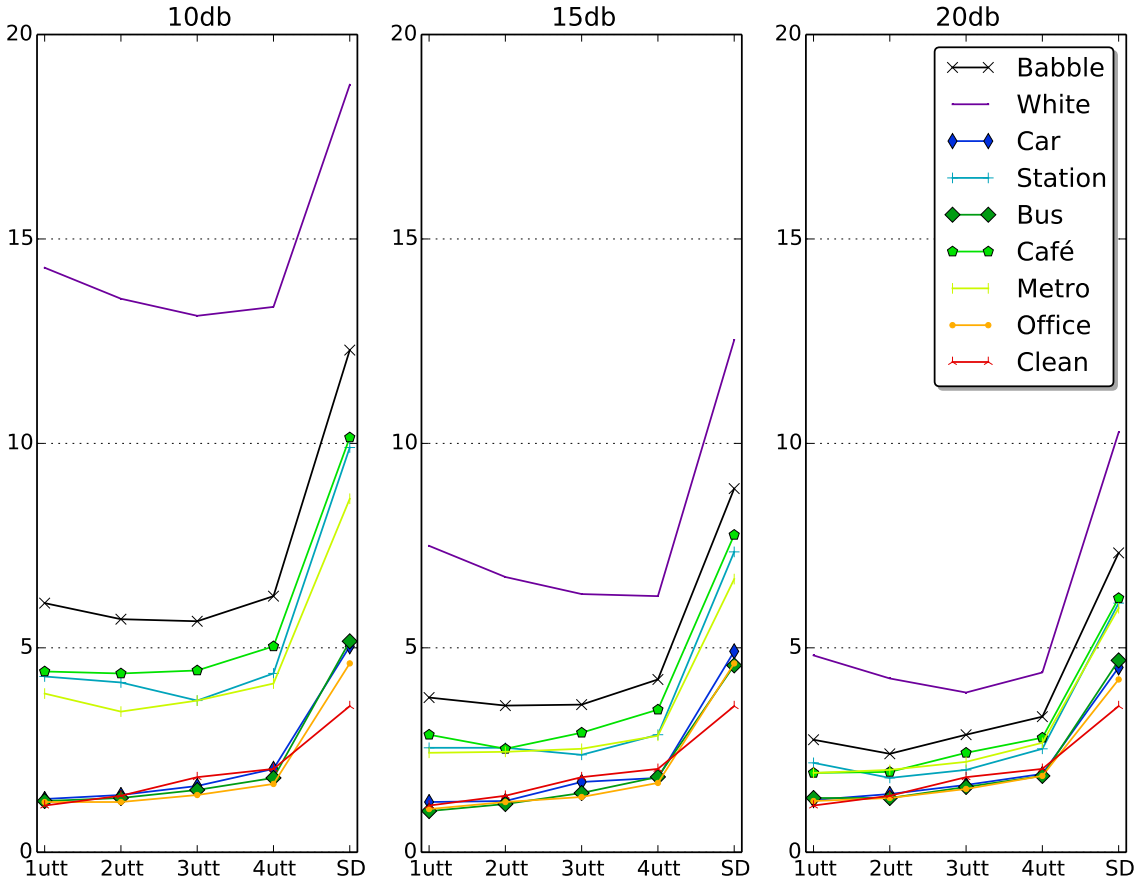


Figure 11: Detector performance when detector is trained without STRAIGHT vocoder or GV but attacker uses those two techniques for generating more natural speech. Metro, cafe, and station noise results have almost overlapped here. Car, bus, and office noise results have almost overlapped here.

3.3 Spoofing and Anti-Spoofing Corpus

3.3.1 Protocol

The SAS spoofing database starts with the Voice Cloning Toolkit (VCTK) database from the Center for Speech Technology Research (CSTR), which is English and freely available. The VCTK database was recorded in a hemi-anechoic chamber using an omni-directional head-mounted microphone (DPA 4035) at a sampling rate of 96 kHz. The motivation for starting with clean studio-recorded speech is that it allows for spoofing attacks that rely on such data. Channel and noise factors can always be simulated at a later date, but in this work we focused only on spoofing under clean

Table 10: Number of trials in the development and evaluation sets of SAS corpus.

	Development			Evaluation		
	Male	Female	Total	Male	Female	Total
Target speakers	45	61	106	45	61	106
Genuine trials	4500	6100	10600	9446	13385	22831
Impostor trials	45000	61000	106000	85592	118000	203592
Spoofed trials	45000	61000	106000	85592	118000	203592

conditions. To design the spoofing database, speech data was taken from VCTK which comprises 45 male and 61 female speakers, and downsampled the signals to 16 kHz at 16 bits-per-sample. The data from each speaker was divided into five parts:

- **Part-A**: 24 parallel utterances (i.e., same text across all speakers) per speaker: training data for spoofing.
- **Part-B**: 20 non-parallel utterances per speaker: additional training for spoofing.
- **Part-C**: 50 non-parallel utterances per speaker: enrollment data for client model training in speaker verification.
- **Part-D**: 100 non-parallel utterances per speaker: development set for speaker verification.
- **Part-E**: Around 200 non-parallel utterances per speaker: evaluation set for speaker verification.

We note that in Part-C, Part-D, and Part-E, all the sentences are randomly selected from newspapers without any repeating sentence across all speakers.

3.3.1.1 Speaker Verification Enrollment and Evaluation

We first introduce the protocol for standard speaker verification evaluation. The enrollment data of each client was selected from Part-C under two scenarios: 5-utterance or 50-utterance enrollments. For 5 utterances this means around 5 to 6 seconds, and

for 50 utterances around 1 minute of speech material. The development set was created from Part-D. It involves genuine trials and impostor trials. All utterances from a client speaker in Part-D were used as genuine trials, and this results in 4500 male and 6100 female genuine trials.

For the impostor trials, 10 randomly selected non-target speakers were used as impostors. All Part-D utterances from a specific impostor were used as impostor trials against the clients model, leading to 45000 male and 61000 female impostor trials. This set is aimed at tuning the system and deciding thresholds. The evaluation is drawn from Part-E. In a similarly fashion to the development set, 9446 male and 13385 female genuine trials, and 85592 male and 118000 female impostor trials were generated. This set is for assessing the performance of speaker verification systems. A summary of the development and evaluation sets is shown in Table 10.

3.3.1.2 Spoofing preparation and execution

We now introduce the protocol for producing the spoofing materials. Two training sets were designed: small and large. The small set consists of data only from Part-A, while the large set includes data from both Part-A and Part-B. Would-be attackers should select one of these to train their spoofing system. The small set comprises parallel training data, and so enables attackers to use voice conversion methods reliant on parallel training data, such as the method implemented in Festvox.

During the execution of speech synthesis spoofing, the transcript of an impostor trial was used as the textual input to the speech synthesis systems, while for voice conversion (VC) spoofing, the speech signal of the impostor trial was the input to the VC system. As a result, the zero-effort impostor trial, the speech synthesis spoofed trial and the voice conversion spoofed trial all have the same language content (i.e., word sequence). The spoofing systems were used to generate spoofing materials for both development and evaluation, and so the number of spoofed trials is exactly the

same as the number of impostor trials (Table 10). This allows fair comparisons to be made between non-spoofed and spoofed speaker verification results.

3.3.1.3 Evaluation metric

As discussed above, the protocol for speaker verification follows the NIST SRE style, so the evaluation metric designed for NIST evaluation can be easily adopted. For example, the performance measures Equal Error Rate (EER), False Acceptance Rate (FAR), False Rejection Rate (FRR) and Detection Cost Function (DCF) can be applied. In the benchmarking results we present here, EERs and FARs will be reported.

3.3.2 Spoofing Approaches

In the current version of SAS, spoofing materials comprise the output from two speech synthesis systems and seven voice conversion systems. These systems are built using both open-source software and collaborators internal systems. Next, we briefly describe the systems that were used to generate the spoofing materials in SAS.

NONE: This is a baseline zero-effort impostor trial in which the impostors own speech is used directly with no attempt to match the target speaker.

SS-SMALL: This HMM-based TTS system is based on the statistical parametric speech synthesis framework described in [12]. The speaker adaptation techniques in this framework allow the generation of a synthetic voice using as little as a few minutes of recorded speech from the target speaker, making it an effective and easily-accessible tool for SV spoofing. The latest version (2.2) of the open-source code HTS [34] was used. In the speech analysis and the average voice training phase, the STRAIGHT vocoder with mixed excitation is used, which results in 60-dimension Bark-Cepstral coefficients, log F0 and 25-dimension band-limited aperiodicity measures [9, 35]. Hidden semi-Markov models (HSMMs) [36] are trained on a large multi-speaker database called voice bank corpus [37] that include hundreds of English speakers to simultaneously model acoustic features and duration. In the speaker adaptation phase, the

speaker-independent HSMMs are transformed using structural variational Bayesian linear regression [38] followed by MAP, using the target speakers data from Part-A. Both the output probability density functions for the acoustic features and the duration model parameters are transformed. To synthesize speech, acoustic feature parameters are generated from the adapted HSMMs using a parameter generation algorithm that considers global variance [8]. An excitation signal is generated using mixed excitation and pitch-synchronous overlap and add [39] and used to excite a Mel-logarithmic spectrum approximation (MLSA) filter [40] corresponding to the STRAIGHT Bark cepstrum, to create the final synthetic speech waveform.

SS-LARGE: This system is the same as SS-SMALL, except that a larger set of adaptation data comprising both Part-A and Part-B was used when adapting the speaker-independent HSMMs to each target speaker.

VC-FESTVOX: This is the voice conversion toolkit within the publicly-available open-source Festvox system. It is based on the algorithm proposed in [41], which is a joint density Gaussian mixture model with maximum likelihood parameter generation considering global variance. The Part-A (i.e., small) set of parallel training data was used, and the default settings of the toolkit were kept, except that the number of Gaussian components in the mixture distributions was set to 32.

VC-GMM: This is another standard GMM-based voice conversion method also using the parallel training data from Part-A. It is very similar to VC-FESTVOX but with some enhancements. STRAIGHT was used as the speech analysis-synthesis method to extract high-quality speech parameters, such as F0, spectral envelope, and aperiodicity measures. The search range for F0 extraction was automatically optimized speaker by speaker to reduce errors. A power threshold for extracting active frames used to estimate the joint density GMM was also optimized automatically per speaker. Two GMMs were trained for separately converting the 1st through 24th Mel-Cepstral coefficients (MCCs) and 5 band aperiodicity measures. The number of

mixture components was set to 32 for the spectral features and 8 for the aperiodicity measures, respectively. For some speaker pairs, the number of components was reduced when defunct mixture components were automatically removed. To enhance the variance of the converted spectral parameter trajectories, GV-based post-filtering [42] was used instead of GV-based parameter conversion.

VC-KPLS: This voice conversion system uses kernel partial least square (KPLS) regression [43], trained on the Part-A (small) parallel data. 300 reference vectors and a Gaussian kernel were used to derive kernel features, and 50 latent components were used in the PLS model. Dynamic kernel features were not included, for simplicity. STRAIGHT was used to extract 24-dimensional Mel-Cepstral coefficients, 25 band aperiodicities (BAPs), and F0.

VC-EVC: This is a many-to-many eigenvoice conversion (EVC) system [44]. The eigenvoice GMM (EV-GMM) was constructed from the training data from one pivot speaker in the ATR Japanese speech database [45], and 273 speakers (137 male, 136 female) from the JNAS database. Settings were the same as in [46]. The 272-dimensional weight vectors were estimated by using the Part-A (small) training data. Covariance matrices in EV-GMM were not updated, i.e. the mean vectors of source and target speakers were independently updated. STRAIGHT was used to extract 24-dimensional Mel-Cepstral coefficients, 5 BAPs, and F0. The number of mixture components was fixed at 128. The conversion method was applied only to the Mel-Cepstral coefficients.

VC-TVC: This is a tensor-based arbitrary voice conversion (TVC) system [46]. To construct the speaker space, the same Japanese dataset as in VC-EVC was used. The size of weight matrices which represent each speaker was set to 48 80. The same part of the SAS database and the same features as in VC-EVC were used, and again only the Mel-Cepstral coefficients were converted, without altering other features.

VC-FS: This is a frame selection voice conversion system, which is a simplified

version of exemplar-based unit selection [47], using a single frame as an exemplar and without a concatenation (join) cost. The Part-A (small) data was used for training. The same features as in VC-KPLS were used, and once again only the Mel-Cepstral coefficients were converted.

VC-C1: As in VC-KPLS and VC-FS, STRAIGHT was used to extract Mel-Cepstral coefficients, BAPs and F0. The first coefficient of the source speakers Mel-Cepstral coefficients was converted by a linear transformation. This is the simplest voice conversion method, since it only changes the overall slope of the spectral envelope, and not any other speaker-specific features. In all the voice conversion approaches, F0 was converted by a global linear transformation: simple mean-variance normalization.

In VC-KPLS, VC-EVC, VC-TVC, VC-FS and VC-C1, source speaker BAPs were simply copied, without undergoing any conversion.

3.3.3 Initial Benchmarking Experiments

To accompany the SAS database, we provide some benchmark speaker verification experimental results.

3.3.3.1 Speaker Verification systems

We used two speaker verification systems representing the current state-of-the-art: Joint Factor Analysis (JFA) [30] and Probabilistic Linear Discriminant Analysis (PLDA) [48], under two enrollment scenarios, 5-utterance and 50-utterance. Both systems used the same front-end to extract acoustic features, comprising 19 dimension MFCC and energy features with delta and delta-delta coefficients. By excluding the static energy feature, 59-dimensional features were used in both systems. The AudioSeg toolkit was used to perform voice activity detection (VAD) [49]. In both systems, we used three Wall Street Journal (WSJ) databases (WSJ0, WSJ1, and

WSJCAM) and the Resource Management database (RM1) for training the Universal Background Model (UBM) and the eigenspaces. From WSJ0 and WSJ1, only the SI training speakers were used. All speakers from the WSJCAM training, development and test sets were used. During scoring, T-norm was applied for both systems.

JFA: A Joint Factor Analysis system with a UBM of 512 components, and eigenvoice and eigenchannel spaces with 300 and 100 dimensions respectively. Cosine scoring was performed on the speaker variability vectors.

PLDA: Using the same UBM as in JFA, the PLDA approach operates in i-vector space, the dimension of which was set to 400. Because i-vectors have a heavy-tailed distribution, radial Gaussianization [50] was performed, then the i-vector dimension was reduced to 200 using linear discriminant analysis (LDA) and the within-class covariance matrices of the resulting vectors were whitened using within-class covariance normalization (WCCN) [3]. The dimensionality of the resulting vectors was further reduced down to 100 by PLDA. Scoring was done with a likelihood ratio test. In the two enrollment scenarios, the short enrollment utterances were merged into sessions of 5 before enrollment. Therefore, after merging, either 1 or 10 sessions were used in enrollment. For PLDA, in the 10 sessions case, i-vectors that were extracted from all 10 sessions were averaged, while for JFA, all features from all sessions were merged. We use JFA-5 and PLDA-5 to denote systems with 5 enrollment utterances (1 session), and JFA-50 and PLDA-50 for the 50-utterance (10 session) case.

3.3.3.2 Initial Benchmarking Results

We only report EERs and FARs for our initial speaker verification results, as the two measures are more related to spoofing. The results are presented in Table 11 and Table 12. Without surprise, the EERs and FARs for the baselines are very low, that is close or below 1% by JFA-50 and PLDA-50 systems, as the SAS database is clean

Table 11: Initial spoofing results on the development set of SAS corpus using the metrics of Equal Error Rate (EER) and False Alarm Rate (FAR) for the two variants (5 and 50) of two speaker verification systems based on Joint Factor Analysis (JFA) or Probabilistic Linear Discriminant Analysis (PLDA).

		EER				FAR			
		JFA		PLDA		JFA		PLDA	
	Spoofing	5	50	5	50	5	50	5	50
Male	(NONE) Baseline	3.29	1.29	1.44	0.66	3.29	1.29	1.44	0.66
	SS-SMALL	25.27	23.83	21.97	19.69	90.80	94.44	90.85	90.98
	SS-LARGE	27.47	25.95	23.96	22.15	93.59	97.23	94.11	94.46
	VC-FESTVOX	**30.09**	**30.36**	**28.94**	**27.97**	**95.55**	**98.32**	**98.60**	**99.20**
	VC-GMM	27.30	27.38	26.76	26.25	92.93	96.51	95.69	96.41
	VC-KPLS	19.60	18.24	20.96	20.11	76.76	84.56	89.45	89.51
	VC-TVC	19.32	17.69	20.03	18.94	73.40	80.32	84.73	84.45
	VC-EVC	15.64	13.12	16.20	14.73	62.34	67.67	80.12	78.83
	VC-FS	23.48	22.49	25.29	23.62	85.84	91.99	94.47	95.41
	VC-C1	3.60	1.44	1.69	0.86	4.48	2.23	2.28	1.25
Female	(NONE) Baseline	6.54	2.08	2.48	1.08	6.54	2.08	2.48	1.08
	SS-SMALL	23.76	17.90	19.49	17.78	79.03	77.01	83.53	89.48
	SS-LARGE	25.71	19.88	22.17	20.73	**83.39**	83.39	89.54	**94.23**
	VC-FESTVOX	**26.36**	**25.04**	**25.42**	**24.74**	82.06	**89.59**	**90.83**	93.20
	VC-GMM	26.32	24.84	23.95	23.65	81.32	88.38	88.70	91.88
	VC-KPLS	19.68	14.40	19.31	17.61	66.85	64.01	79.08	80.56
	VC-TVC	19.63	14.30	17.10	15.09	64.60	63.29	72.99	75.35
	VC-EVC	17.98	11.95	14.99	12.78	61.96	56.64	69.07	70.43
	VC-FS	20.89	15.94	21.08	19.70	68.87	71.19	81.82	87.51
	VC-C1	7.74	2.70	3.07	1.53	11.95	5.06	5.26	3.20

without any channel or noise effects. However, the short duration of the trials prevents the EERs or FARs to go even lower. Even through the ASV systems achieve very good speaker verification performance, they are extremely vulnerable to spoofing attacks. Even the most simple VC-C1 spoofing attack, which only changes the spectral slope of the source speaker, considerably increases the False Alarm Rate (FAR). The more sophisticated attacks using speech synthesis or voice conversion lead to FARs as high as 99.11%. In general, speech synthesis leads to FARs of over 90% for male and over 80% for female, even for the SS-SMALL system which has access to only 24 utterances (Part-A) from the target speaker. Voice conversion spoofing is sometimes an even more effective attack that speech synthesis. It is worth highlighting that

Table 12: Initial spoofing results on the evaluation set of SAS corpus using the metrics of Equal Error Rate (EER) and False Alarm Rate (FAR) for the two variants (5 and 50) of two speaker verification systems based on Joint Factor Analysis (JFA) or Probabilistic Linear Discriminant Analysis (PLDA).

		JFA		PLDA	
	Spoofing	5	50	5	50
Male	(NONE) Baseline	3.43	1.40	1.44	0.66
	SS-SMALL	90.80	94.38	90.71	90.60
	SS-LARGE	93.64	97.32	93.68	94.05
	VC-FESTVOX	**95.46**	**98.44**	**98.41**	**99.11**
	VC-GMM	92.80	96.45	95.59	96.21
	VC-KPLS	77.10	84.70	89.19	89.46
	VC-TVC	73.68	80.67	84.46	84.37
	VC-EVC	62.68	67.94	80.09	78.92
	VC-FS	85.51	91.82	94.17	95.13
	VC-C1	4.66	2.16	2.24	1.15
Female	(NONE) Baseline	6.40	2.02	2.38	1.00
	SS-SMALL	79.43	77.53	83.96	89.88
	SS-LARGE	**83.58**	83.71	**89.90**	**94.55**
	VC-FESTVOX	82.45	**90.07**	88.69	91.27
	VC-GMM	81.88	89.02	89.37	92.41
	VC-KPLS	67.22	64.55	79.64	81.10
	VC-TVC	64.73	63.68	73.30	75.55
	VC-EVC	62.12	57.14	69.95	71.35
	VC-FS	69.12	71.52	82.27	87.78
	VC-C1	11.78	4.92	5.14	3.19

the publicly-available voice conversion toolkit VC-FESTVOX is generally at least as effective as the other voice conversion and speech synthesis techniques. The second interesting observation is that although VC-EVC uses Japanese database to train eigenvoice for adaptation, it still increase FARs as high as other methods. An other observation is that even though more enrollment data is helpful to have lower EERs and FARs on non-spoofed data, it does not achieve lower error rates in the face of spoofing. These spoofing results are consistent with our previous findings on both telephone quality [5, 51] and clean speech [52, 53].

3.4 Importance Weighting in GMM Based Synthetic Detectors

An overview of the proposed system is shown in Fig. 12. Mel-frequency cepstral coefficients (MFCC) are first extracted from the speech utterance. Then, the feature vectors are grouped together into J groups. In one approach, vectors that are aligned with the same Gaussian component of a GMM are grouped together. In another approach, feature vectors that belong to the same phoneme or sound class constitute a group. Details of grouping are described in the next section.

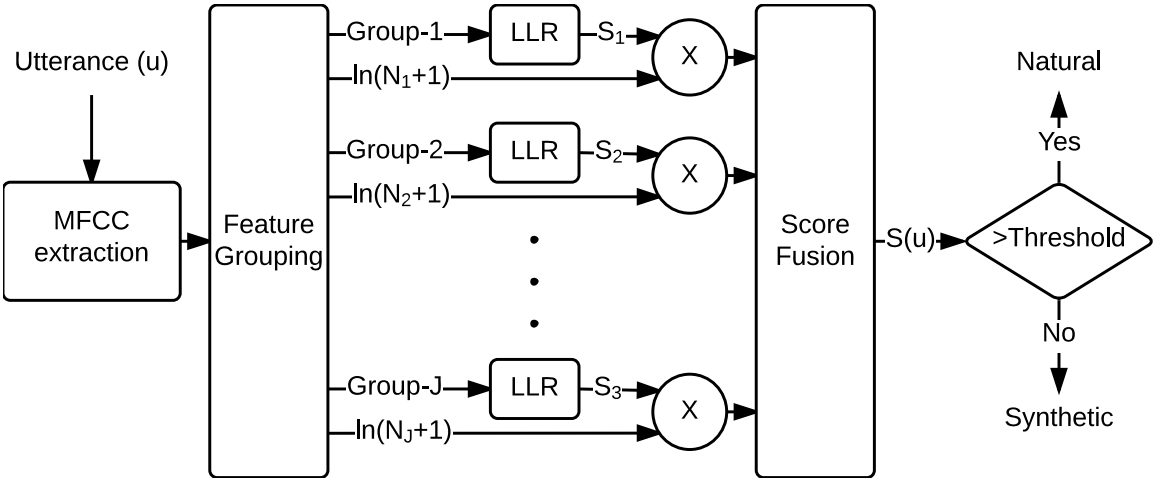


Figure 12: Overview of the proposed synthetic speech detectors.

After grouping, log-likelihood ratio (LLR) detection is done for each group of feature vectors. To compute LLR, a GMM is trained for natural speech and a GMM is trained for synthetic speech. Same GMMs are used for all J groups. Once the score of each group is computed, score fusion is done using a logistic regression function to compute the final score $S(u)$. A hard threshold is used to compute the final decision.

In the baseline detector, which does not use any grouping, given an utterance u, assuming independent speech frames

$$LLR(u) = \frac{1}{N}\sum_{i=1}^{N} log(\mathbf{x}_i|\mathbf{\Lambda}_{nat}) - log(\mathbf{x}_i|\mathbf{\Lambda}_{syn}), \tag{20}$$

where N is the total number of frames, x_i is the feature vector for the i^{th} frame, $\mathbf{\Lambda_{nat}}$

is the canonical model of GMM for the natural speech, and $\mathbf{\Lambda_{syn}}$ is the canonical model of GMM for the synthetic speech. The final decision is done using a hard threshold for $LLR(u)$.

In the proposed approach, the decision is based on the utterance score

$$S(u) = \Phi(S_1, S_2, ..., S_J) \tag{21}$$

where Φ is a nonlinear function and score S_j for each group j is

$$\frac{1}{N_j} \sum_{i=1}^{N_j} log(\mathbf{x}_i^{(j)}|\mathbf{\Lambda}_{nat}) - log(\mathbf{x}_i^{(j)}|\mathbf{\Lambda}_{syn}). \tag{22}$$

The rationale of this approach is to develop detectors that are focused on different segments of speech and weigh each segment depending on its information content. For example, nasals are typically not modeled well by vocoders because of the spectral dip in nasals that are not modeled with an all-pole model. A detector that is focused only on nasals can detect those artifacts. Similarly, synthetic speech may contain some short-duration glitches that are not observed in natural speech. Even though those artifacts may be detectable by some of the Gaussian components in synthetic GMMs. when the frame likelihoods are averaged as in Eq. 20, those short-duration events may not be detected because of the low weight they get and noise introduced in other frames. Focusing on those highly informative Gaussians regardless of their durations and assigning them high weight can improve the detection performance in those cases.

3.4.1 Duration-based Weighting

Distribution of the frame-level LLR values approximately follow a Gaussian distribution in most utterances. By averaging the LLR scores, as done in Eq. 22, assuming Gaussianity, a maximum-likelihood (ML) estimate of the mean is found. Considering the fact that the ML estimate of the mean of a Gaussian has an estimation variance that is inversely proportional with the number of observations, reliability of the

detector j increases when N_j increases. To take the estimation variance, hence the uncertainty of the detector scores, into account, we propose the duration-weighted score

$$S_j^{'} = \ln(N_j + 1)S_j \tag{23}$$

where ln(.) is the natural logarithm.

3.4.2 Feature Grouping Methods

Three feature grouping strategies are investigated. In the phoneme-based approach, each phoneme constitutes a group. Thus, feature vectors that occur within a particular phoneme type in the utterance are grouped together.

One of the problems with the phoneme-based approach is that some of the utterances provided in the challenge were short (2-3seconds) which means that many of the phonemes were not observed in those cases. Because broad acoustic-phonetic sound classes share similar acoustic properties, we hypothesized that if a system performs poorly in synthesizing a phoneme, it will most likely perform poorly for the other phonemes that are acoustically similar. Thus, to make more data available for each group, a class-based approach is used for grouping in the second approach. In the class-based approach, five sound classes are used: vowels, nasals, glides, stops, and rest. The rest class contains all phonemes that do not belong to the other four classes.

The phoneme- and class-based methods are good at detecting artifacts that occur in relatively long segments. However, they are not designed for detecting sudden glitches that can easily occur with unit selection systems or some of the voice conversion systems. Location of those glitches are random for the most part and they may not be detected with detectors that are focused on long-duration segments.

To address the issue of short-duration artifact detection, we propose Gaussian-based grouping where each frame in the utterance is first aligned with the GMM of

natural speech. Then, frames that are aligned with the same Gaussian are grouped together. This approach allows detection of frame-level artifacts and assign them high weight even though they may occur infrequently in the utterance.

3.4.3 Experiments

The synthetic speech detectors were trained with 19 Mel-frequency cepstral coefficients (MFCCs) together with the delta and delta-delta features. In short-time analysis, frame length was 25msec and frame rate was 10msec. Bigaussian voice activity detection (VAD) was used where energy of the speech and noise frames are modeled with single Gaussians and likelihood ratio detector is used to detect speech frames.

The baseline synthetic speech detector had a 512-component GMM to model natural speech. Similarly, synthetic speech was modeled with 512-component GMM. For natural speech, GMM training was initialized using k-means clustering. The GMM for synthetic speech was adapted from the GMM of the natural speech using a maximum a posteriori (MAP) approach. Experiments with synthetic speech GMM that was trained independent of the natural speech GMM were also performed for comparison.

The phoneme-based approach requires a phoneme recognizer since the transcriptions of the challenge data were not available. The Hungarian phoneme recognizer [54] was trained with WSJ-CAM database and used here for phoneme recognition. A total of 37 phonemes were used. Outputs of the phoneme recognizer were mapped to sound classes and used in sound-class based detector also.

The spoofing challenge database was used for training, development and evaluation of all systems[2]. The BOSARIS toolkit [55] was used to train the logistic regression algorithm that was used for fusing the scores of detectors.

[2]We did not participate officially in the challenge because we took part in generating some of the spoofing material.

3.4.4 Results and Discussion

Table 13: Performance of the baseline and proposed detectors in terms of equal-error-rates (EERs) for the development and evaluation data. Results are presented without duration-weighting. S1, S2, and S5 systems use voice conversion (VC). S3 and S4 systems use HMM-based synthesis. Other systems are unknown. Best performing algorithm for each attack type is shown in bold.

		Normal					
		LLR		Logistic Regression			
		Direct	Adapt	Class	Phone	Gauss	Fusion
Development	S1 (VC)	0.47	0.76	0.68	0.69	0.47	**0.41**
	S2 (VC)	10.24	5.12	3.37	3.41	1.89	**1.83**
	S3 (HMM)	0.07	0.07	**0.03**	0.09	0.20	0.17
	S4 (HMM)	0.04	0.09	0.05	**0.03**	0.25	0.20
	S5 (VC)	4.63	3.04	2.78	2.86	1.72	1.57
	Total	4.21	2.42	1.92	1.77	1.17	**1.11**
Evaluation	S1 (VC)	0.54	0.57	0.55	0.56	0.48	**0.41**
	S2 (VC)	9.24	4.47	2.78	2.71	1.89	**1.75**
	S3 (HMM)	0.07	**0.02**	0.04	0.04	0.18	0.12
	S4 (HMM)	0.07	**0.03**	0.05	0.05	0.17	0.11
	S5 (VC)	3.95	1.72	1.99	2.14	1.48	**1.36**
	S6	3.49	1.35	1.39	1.40	1.09	**0.98**
	S7	1.91	1.65	0.84	0.87	0.75	**0.63**
	S8	**0.46**	1.03	0.76	0.85	0.83	0.70
	S9	**0.43**	1.26	0.93	1.02	0.76	0.65
	S10	**27.24**	29.62	32.14	33.59	30.05	29.81
	Known	2.77	1.36	1.08	1.10	0.84	**0.75**
	Unknown	6.70	6.98	7.21	7.54	6.70	**6.55**
	All	4.74	4.17	4.15	4.32	3.77	**3.65**

Experimental results for the development and evaluation data are shown in Table 13 and Table 14. The baseline LLR detector is trained with two different methods. In one approach (LLR-noAdapt), two independent GMMs are trained for the natural and synthetic speech. In the second approach (LLR-Adapt), a GMM is trained for natural speech and then adapted to the synthetic speech using MAP adaptation.

The LLR-Adapt system performed better for known conditions while LLR-noAdapt performed better for unknown conditions. Thus, even though LLR-Adapt performed

Table 14: Performance of the duration-based weighted detectors in terms of equal-error-rates (EERs) for the development and evaluation data. Results are presented with duration-weighting. S1, S2, and S5 systems use voice conversion (VC). S3 and S4 systems use HMM-based synthesis. Other systems are unknown. Best performing algorithm for each attack type is shown in bold.

		Duration-based weighted			
		Logistic Regression			
		Class	Phone	Gauss	Fusion
Development	S1 (VC)	0.54	0.54	0.51	0.46
	S2 (VC)	2.99	3.13	2.26	2.20
	S3 (HMM)	**0.03**	0.09	0.18	0.11
	S4 (HMM)	**0.03**	0.07	0.20	0.13
	S5 (VC)	2.65	2.72	1.59	**1.47**
	Total	1.67	1.66	1.19	1.14
Evaluation	S1 (VC)	0.51	0.50	0.46	0.42
	S2 (VC)	2.63	2.44	2.15	2.03
	S3 (HMM)	**0.02**	0.03	0.15	0.09
	S4 (HMM)	**0.03**	0.04	0.13	0.08
	S5 (VC)	1.89	1.97	1.50	1.40
	S6	1.31	1.24	1.13	1.01
	S7	0.85	0.94	0.70	0.65
	S8	0.71	0.87	0.79	0.70
	S9	0.94	1.02	0.68	0.64
	S10	31.39	32.25	29.88	29.78
	Known	1.02	0.99	0.88	0.81
	Unknown	7.04	7.27	6.64	**6.55**
	All	4.03	4.13	3.76	3.68

better than LLR-noAdapt on average, it could not generalize as good as the LLR-noAdapt. This result indicates that, during GMM training, some of the novel clusters in the synthetic data that were useful for ambiguity detection, could not be modeled well with adaptation of GMM for natural speech.

Gaussian-based system performed better than class- and phoneme-based methods both for known and unknown conditions. In particular, Gaussian-based approach performed better for the S1, S2, and S5 methods, all of which are voice conversion algorithms. Unlike the phoneme- and class-based systems, Gaussian-based detector can learn to detect short-duration artifacts. Thus, the presence of short-duration acoustic distortions seems to be more informative for detecting voice conversion attacks.

Class-based system performed better for S3 and phoneme-based system performed better for S4 attack methods. Both S3 and S4 are generated with HMM-based TTS. Unlike the voice conversion systems, HMM-based TTS systems generate smooth trajectories. Thus, sudden acoustic distortions are rarely generated with those systems. In this case, overly-smooth longer segments seem to be more informative for detection. Small distortions in a long segment can be detected well with class- and phoneme-specific detectors that are focused on particular segments. However, Gaussian-based approach is not expected to be as successful with this type of attack because speech frames are generated with a maximum-likelihood approach in HMM-based synthesis. Thus, the parameter generation algorithm is designed to generate high likelihoods for each frame and individual Gaussians are not expected to detect the artifacts in features.

Duration-based weighting consistently improved class- and phoneme-based performance. However, for the Gaussian-based approach, performance improved slightly for the unknown systems and degraded slightly for the known systems. We believe there are at least two major factors behind this result. Firstly, because an important strength of the Gaussian-approach is its ability to detect short-time artifacts, weighting with duration can hurt its performance. Secondly, duration of observed Gaussians can change significantly depending on the spoofing system used which can increase the variability of features and make the detection task harder. Because ASR systems take phoneme durations into account during recognition, that effect is not as important in the phoneme- and class-based methods.

The core hypothesis in the proposed system was that different Gaussians, phonemes, sound-classes contribute different amounts of information for synthetic speech detection. To test that hypothesis, experiments were performed with each Gaussian, phoneme, and sound-class separately. For the Gaussian case, results are shown in Fig 13, for the phoneme case, results are shown in Fig 14. In both cases, large

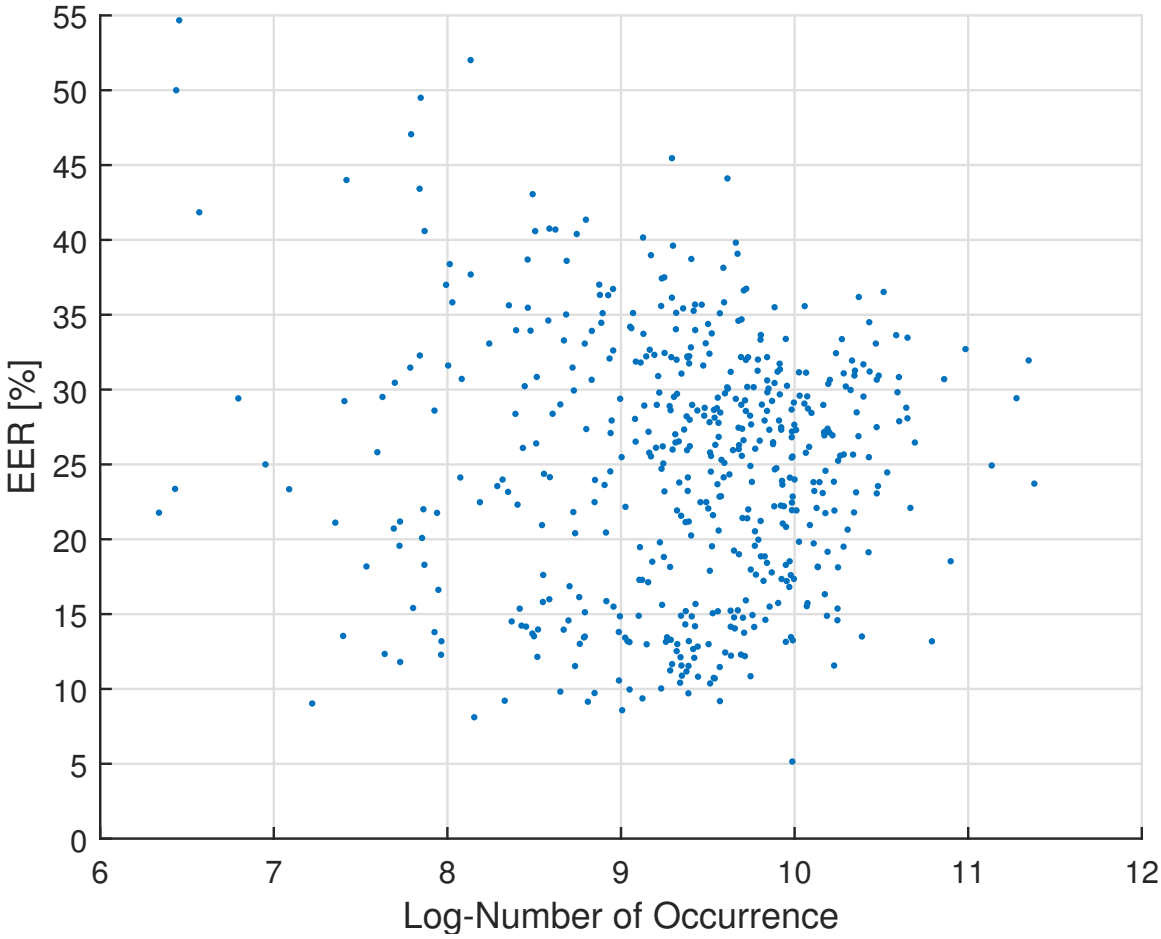


Figure 13: Detection performance of each Gaussian component versus its logarithm of number of occurrence in the development utterances is shown.

variation in detection performance can be observed which verifies our hypothesis.

Detector results for the class-based system is shown in Table 15. Performance of each class is significantly different from each other and they change substantially depending on the attack method. Also note that, even though vowel class is observed more than other classes, their performance is better than other systems only for HMM-based TTS attacks. For the voice-conversion attacks, short-duration stop sounds become more informative even though they occur far less frequently than the vowels.

Fig. 13 shows the correlation of number of occurrences vs EER computed with each of the 512 Gaussians. Even though EER and durations have a negative correlation, the pattern is weak and does not impact the overall detector performance significantly.

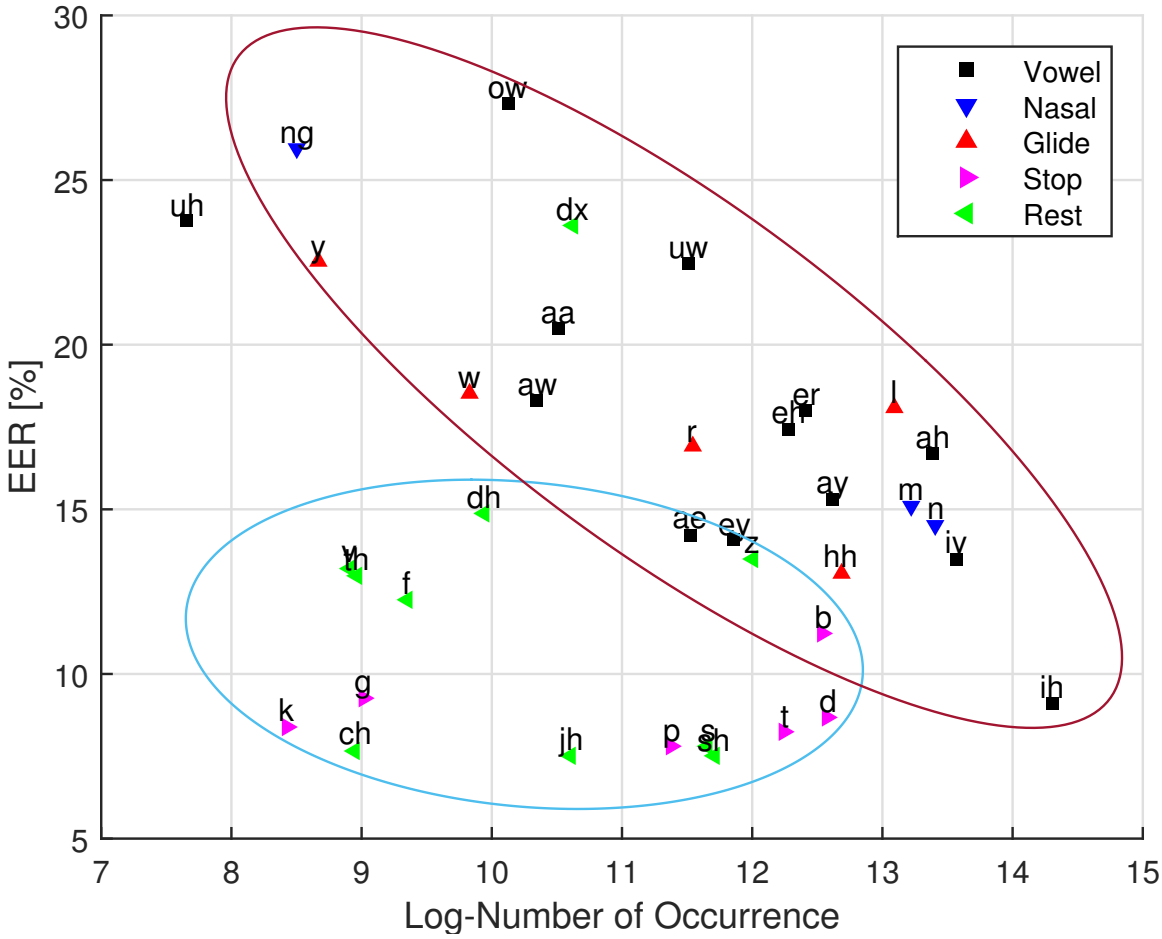


Figure 14: Detection performance of each phoneme versus its logarithm of number of occurrence in the development utterances is shown. Phonemes that are in the same sound-class are shown with the same color and shape.

This result is inline with the finding that duration-based weighting does not improve the performance of the Gaussian-based system.

The effect of duration is more significant with phoneme-based detector compared to the Gaussian-based detector. Duration versus EER is shown in Fig. 14 where a stronger negative correlation is observed compared to the Gaussian case especially for the vocalic sounds. The correlation disappears for some of the highly informative stop and fricative sounds.

The proposed detectors performed substantially better than the baseline detectors for known attack types. However, the difference is not substantial for the unknown

Table 15: Performance of each of the sound-class detectors measured in terms of equal-error-rates (EERs) for the development data. Frequency of observation in development utterances is also shown for each class type.

Class	S1	S2	S3	S4	S5	All	Freq.
Vowel	3.52	13.30	**0.65**	**0.74**	7.26	6.35	0.542
Nasal	8.90	20.82	5.09	5.86	13.79	11.62	0.156
Glide	9.33	21.69	4.10	4.44	15.92	12.15	0.118
Stop	**2.24**	**4.78**	0.70	0.78	**6.77**	**3.68**	0.112
Rest	8.97	10.58	3.32	3.78	16.08	9.43	0.072

attack types. To further boost the performance, the detectors were fused with a second stage of logistic regression algorithm. The fusion improved performance both for known and unknown attack types which indicate that the detectors generate complementary information.

CHAPTER IV

CONCLUSION

In this work, effective techniques for spoofing a state-of-the-art speaker verification system are proposed. 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 linear regression (LR), hybrid synthesis (HYB), and their interpolation (HYB+LR) to spoof the SSD while further improving the effectiveness of spoofing at the verification system.

The proposed systems substantially outperformed the baseline SSS system in spoofing the SSDs both in matched and mismatched conditions. LR approach outperformed the hybrid approach when 1 or 2 utterances were available. However, with increasing data sizes, HYB approach outperformed the LR approach. HYB+LR approach worked better than both HYB and LR systems.

SSS systems are known to generate smooth trajectory and it was interesting to investigate how much that helps SSDs detect synthetic speech. Indeed, delta features were found to be useful for detection under matched conditions for the SSS and hybrid systems. However, the LR system was found to be easier to detect when the delta features were missing which is related to rapid frame-to-frame variations that is generated with the LR approach which confused the SSDs. Under mismatched conditions, SVM-based SSD was found to be more reliable without the delta features, but it is still worse than GMM-based SSD.

SSS system was found to be effective at spoofing the voice verification system. However, hybrid and LR algorithms were substantially more effective than the SSS approach. Hybrid algorithm substantially outperformed both SSS and LR algorithms.

HYB+LR algorithm had comparable performance to the hybrid algorithm. Even though LR algorithm creates more natural but less target-specific features, it still performed better than the SSS algorithm and did not substantially degrade the performance of the hybrid algorithm while spoofing the verification system.

In the combined system, HYB+LR algorithm performed significantly better than all other systems. This is partly related to its high performance at spoofing the SSD and the voice verification system. Moreover, calibration of the SSD with SSS-generated data, which significantly hurt its performance when tested with HYB+LR, also caused significant increase in the false alarm rate with the HYB+LR algorithm.

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.

Moreover, substantial performance gains are obtained when the verification system is trained with mixed noise conditions at and above 10 dB and noise is intentionally added to synthetic speech. We also proposed a synthetic speech detector that is found to have excellent performance in noisy conditions.

The proposed detector did not perform as well when different SSS vocoders are used for training and testing the detector. In the future work, we will focus increasing the robustness of the detector to mismatch in SSS techniques.

In this work, we also have presented the first version of spoofing and anti-spoofing corpus, which is becoming a standard dataset for spoofing and anti-spoofing research. To set an initial benchmark, we have provided spoofing results when attacking two speaker verification systems. Without any countermeasures in place, these verification systems are extremely vulnerable to spoofing attacks from many of the nine spoofing methods included in SAS.

Furthermore, We have investigated a multi-detector approach for counterspoofing

where each detector is focused on a particular acoustic segment. The Gaussian-based detector performed better in voice conversion attacks. Phoneme- and class-based detectors performed better for HMM-based synthesis attacks. Duration-based feature normalization improved the phoneme- and class-based systems but not the Gaussian-based system. The proposed systems performed substantially better than the baseline system in known attack types. In unknown attacks, the improvement was not substantial. Fusing the scores of proposed detectors further improved the performance in both known and unknown conditions.

Our goal was to take a commonly used likelihood ratio detector and use it in a segment-specific manner. The hypothesis here was that different segments contribute different amounts of information and their scores should be weighted accordingly. Results confirmed our hypothesis. Because we did not assume any prior information, we have used the commonly used MFCC features. In the future work, we will investigate a richer set of features and other classifiers such as SVM to further improve the detection performance.

Bibliography

[1] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li, "Spoofing and countermeasures for speaker verification: A survey," *Speech Communication*, vol. 66, pp. 130 – 153, 2015.

[2] A. F. Martin, M. Yadagiri, G. R. Doddington, C. S. Greenberg, and V. M. Stanford, "The 2012 nist speaker recognition evaluation," in *NIST SRE 2012 Workshop*, Dec 2012.

[3] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, pp. 788–798, May 2011.

[4] P. L. De Leon, M. Pucher, J. Yamagishi, I. Hernaez, and I. Saratxaga, "Evaluation of speaker verification security and detection of hmm-based synthetic speech," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, pp. 2280–2290, Oct 2012.

[5] T. Kinnunen, Z.-Z. Wu, K. A. Lee, F. Sedlak, E. S. Chng, and H. Li, "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 on*, pp. 4401–4404, 2012.

[6] N. W. D. Evans, T. Kinnunen, and J. Yamagishi, "Spoofing and countermeasures for automatic speaker verification," in *INTERSPEECH*, (Lyon, FRANCE), Aug 2013.

[7] N. Evans, T. Kinnunen, J. Yamagishi, Z. Wu, F. Alegre, and P. De Leon, "Speaker recognition anti-spoofing," in *Handbook of Biometric Anti-Spoofing*, pp. 125–146, Springer, 2014.

[8] T. Tomoki and K. Tokuda, "A speech parameter generation algorithm considering global variance for hmm-based speech synthesis," *IEICE TRANSACTIONS on Information and Systems*, vol. 90, no. 5, pp. 816–824, 2007.

[9] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné, "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 communication*, vol. 27, no. 3, pp. 187–207, 1999.

[10] Z. Wu, T. Kinnunen, E. S. Chng, H. Li, and E. Ambikairajah, "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)*, pp. 1–5, 2012.

[11] F. Alegre, R. Vipperla, and N. Evans, "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.

[12] A. W. Black, H. Zen, and K. Tokuda, "Statistical parametric speech synthesis," in *Acoustics, Speech and Signal Processing. ICASSP, IEEE International Conference on*, vol. 4, 2007.

[13] J. Yamagishi, T. Kobayashi, Y. Nakano, K. Ogata, and J. Isogai, "Analysis of speaker adaptation algorithms for HMM-based speech synthesis and a constrained SMAPLR adaptation algorithm," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 17, no. 1, pp. 66–83, 2009.

[14] J. Yamagishi, B. Usabaev, S. King, O. Watts, J. Dines, J. Tian, Y. Guan, R. Hu, K. Oura, Y.-J. Wu, *et al.*, "Thousands of voices for HMM-based speech synthesis-analysis and application of TTS systems built on various ASR corpora," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 5, pp. 984–1004, 2010.

[15] T. Satoh, T. Masuko, T. Kobayashi, and K. Tokuda, "A robust speaker verification system against imposture using an hmm-based speech synthesis system.," in *INTERSPEECH*, pp. 759–762, 2001.

[16] L.-W. Chen, W. Guo, and L.-R. Dai, "Speaker verification against synthetic speech," in *Chinese Spoken Language Processing (ISCSLP), 2010 7th International Symposium on*, pp. 309–312, IEEE, 2010.

[17] F. Alegre, R. Vipperla, A. Amehraye, and N. W. D. Evans, "A new speaker verification spoofing countermeasure based on local binary patterns," in *INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association*, (Lyon, FRANCE), 08 2013.

[18] T. F. Quatieri, *Discrete-time speech signal processing: principles and practice*. Pearson Education India, 2002.

[19] Z.-Z. Wu, C. E. Siong, and H. Li, "Detecting converted speech and natural speech for anti-spoofing attack in speaker recognition.," in *INTERSPEECH*, 2012.

[20] D. Matrouf, J.-F. Bonastre, and C. Fredouille, "Effect of speech transformation on impostor acceptance," in *Acoustics, Speech and Signal Processing. ICASSP Proceedings, IEEE International Conference on*, vol. 1, 2006.

[21] F. Alegre, A. Amehraye, and N. Evans, "Spoofing countermeasures to protect automatic speaker verification from voice conversion," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 3068–3072, IEEE, 2013.

[22] Z. Wu, X. Xiao, E. S. Chng, and H. Li, "Synthetic speech detection using temporal modulation feature," in *ICASSP*, 2013.

[23] A. Ogihara, U. Hitoshi, and A. Shiozaki, "Discrimination method of synthetic speech using pitch frequency against synthetic speech falsification," *IEICE transactions on fundamentals of electronics, communications and computer sciences*, vol. 88, no. 1, pp. 280–286, 2005.

[24] P. L. De Leon, B. Stewart, and J. Yamagishi, "Synthetic speech discrimination using pitch pattern statistics derived from image analysis.," in *INTERSPEECH*, 2012.

[25] F. Alegre, A. Amehraye, and N. Evans, "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*, pp. 1–8, IEEE, 2013.

[26] Z. Wu, S. Gao, E. S. Cling, and H. Li, "A study on replay attack and anti-spoofing for text-dependent speaker verification," in *Asia-Pacific Signal and Information Processing Association, 2014 Annual Summit and Conference (APSIPA)*, pp. 1–5, IEEE, 2014.

[27] Z. Wu, A. Larcher, K.-A. Lee, E. Chng, T. Kinnunen, and H. Li, "Vulnerability evaluation of speaker verification under voice conversion spoofing: the effect of text constraints.," in *INTERSPEECH*, pp. 950–954, 2013.

[28] A. Larcher, K. A. Lee, B. Ma, and H. Li, "Text-dependent speaker verification: Classifiers, databases and rsr2015," *Speech Communication*, vol. 60, pp. 56–77, 2014.

[29] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *Speech and Audio Processing, IEEE Transactions on*, vol. 13, pp. 345 – 354, 2005.

[30] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, pp. 1435–1447, May 2007.

[31] S. Tiomkin, D. Malah, S. Shechtman, and Z. Kons, "A hybrid text-to-speech system that combines concatenative and statistical synthesis units," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 19, no. 5, pp. 1278–1288, 2011.

[32] K. Tokuda, T. Kobayashi, T. Masuko, and S. Imai, "Mel-generalized cepstral analysis-a unified approach to speech spectral estimation.," in *ICSLP*, 1994.

[33] D. B. Paul and J. M. Baker, "The design for the wall street journal-based csr corpus," in *Proceedings of the workshop on Speech and Natural Language*, pp. 357–362, Association for Computational Linguistics, 1992.

[34] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. Black, and K. Tokuda, "The HMM-based speech synthesis system (HTS) version 2.0," in *Proc. of Sixth ISCA Workshop on Speech Synthesis*, pp. 294–299, Citeseer, 2007.

[35] J. Yamagishi and O. Watts, "The cstr/emime hts system for blizzard challenge 2010," 2010.

[36] Z. Heiga, K. Tokuda, T. Masuko, T. Kobayasih, and T. Kitamura, "A hidden semi-markov model-based speech synthesis system," *IEICE transactions on information and systems*, vol. 90, no. 5, pp. 825–834, 2007.

[37] C. Veaux, J. Yamagishi, and S. King, "The voice bank corpus: Design, collection and data analysis of a large regional accent speech database," in *Oriental COCOSDA held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE), 2013 International Conference*, pp. 1–4, IEEE, 2013.

[38] S. Watanabe, A. Nakamura, and B.-H. F. Juang, "Structural bayesian linear regression for hidden markov models," *Journal of Signal Processing Systems*, vol. 74, no. 3, pp. 341–358, 2014.

[39] E. Moulines and F. Charpentier, "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones," *Speech communication*, vol. 9, no. 5, pp. 453–467, 1990.

[40] T. Fukada, K. Tokuda, T. Kobayashi, and S. Imai, "An adaptive algorithm for mel-cepstral analysis of speech," in *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, vol. 1, pp. 137–140, IEEE, 1992.

[41] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 8, pp. 2222–2235, 2007.

[42] T. Toda, T. Muramatsu, and H. Banno, "Implementation of computationally efficient real-time voice conversion.," in *INTERSPEECH*, Citeseer, 2012.

[43] E. Helander, H. Silén, T. Virtanen, and M. Gabbouj, "Voice conversion using dynamic kernel partial least squares regression," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 3, pp. 806–817, 2012.

[44] Y. Ohtani, T. Toda, H. Saruwatari, and K. Shikano, "Non-parallel training for many-to-many eigenvoice conversion," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pp. 4822–4825, IEEE, 2010.

[45] A. Kurematsu, K. Takeda, Y. Sagisaka, S. Katagiri, H. Kuwabara, and K. Shikano, "Atr japanese speech database as a tool of speech recognition and synthesis," *Speech Communication*, vol. 9, no. 4, pp. 357–363, 1990.

[46] D. Saito, N. Minematsu, and K. Hirose, "Effects of speaker adaptive training on tensor-based arbitrary speaker conversion.," in *INTERSPEECH*, 2012.

[47] Z. Wu, T. Virtanen, T. Kinnunen, E. Chng, and H. Li, "Exemplar-based unit selection for voice conversion utilizing temporal information.," in *INTERSPEECH*, pp. 3057–3061, 2013.

[48] P. Li, Y. Fu, U. Mohammed, J. H. Elder, and S. J. Prince, "Probabilistic models for inference about identity," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 1, pp. 144–157, 2012.

[49] G. Gravier, M. Betser, and M. Ben, "audioseg: Audio segmentation toolkit, release 1.2," *IRISA, january*, 2010.

[50] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems.," in *Interspeech*, pp. 249–252, 2011.

[51] J.-F. Bonastre, D. Matrouf, and C. Fredouille, "Artificial impostor voice transformation effects on false acceptance rates.," in *INTERSPEECH*, pp. 2053–2056, 2007.

[52] P. L. De Leon, M. Pucher, and J. Yamagishi, "Evaluation of the vulnerability of speaker verification to synthetic speech," 2010.

[53] P. L. De Leon, M. Pucher, J. Yamagishi, I. Hernaez, and I. Saratxaga, "Evaluation of speaker verification security and detection of hmm-based synthetic speech," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 8, pp. 2280–2290, 2012.

[54] P. Schwarz, P. Matejka, L. Burget, and O. Glembek, "Phoneme recognizer based on long temporal context," *Speech Processing Group, Faculty of Information Technology, Brno University of Technology.[Online]. Available: http://speech. fit. vutbr. cz/en/software*, 2006.

[55] N. Brümmer and E. de Villiers, "The bosaris toolkit: Theory, algorithms and code for surviving the new dcf," *arXiv preprint arXiv:1304.2865*, 2013.

VITA

Ali Khodabakhsh was born in Zanjan, Iran. After completing high school in National Organization for Development of Exceptional Talents (NODET), he started to study Electrical/Biomedical Engineering (bio-electric) program in University of Tehran. In 2012, he started M.Sc. in Computer Science at Özyeğin university on Speech Processing at the Speech Processing laboratory.