Classifiers for Synthetic Speech Detection: A Comparison

Cemal Hanilçi[1,2], *Tomi Kinnunen*[1], *Md Sahidullah*[1], *Aleksandr Sizov*[1]

[1]School of Computing, University of Eastern Finland, Finland
[2]Department of Electrical and Electronic Engineering, Bursa Technical University, Turkey
chanil@cs.uef.fi, tkinnu@cs.joensuu.fi, sahid@cs.uef.fi, sizov@cs.uef.fi

Abstract

Automatic speaker verification (ASV) systems are highly vulnerable against *spoofing attacks*, also known as imposture. With recent developments in speech synthesis and voice conversion technology, it has become important to detect synthesized or voice-converted speech for the security of ASV systems. In this paper, we compare five different classifiers used in speaker recognition to detect synthetic speech. Experimental results conducted on the ASVspoof 2015 dataset show that support vector machines with generalized linear discriminant kernel (GLDS-SVM) yield the best performance on the development set with the EER of 0.12 % whereas Gaussian mixture model (GMM) trained using maximum likelihood (ML) criterion with the EER of 3.01 % is superior for the evaluation set.
Index Terms: spoof detection, speaker recognition

1. Introduction

Automatic speaker verification (ASV) aims at recognizing speakers using their voices and is gradually gaining popularity as a biometric person authentication technique alongside with the more traditional face and fingerprint biometrics. However, similar to these biometrics, *spoofing*, the situation of an impostor speaker masquerading as another to gain unauthorized access, is a security problem [1].

Speaker recognition systems can be deliberately spoofed by replay [2], impersonation [3, 4], speech synthesis [5] and voice conversion [6, 7]. Replay attack, repetition of a prerecorded speech signal of the target speaker is one of the easiest ways to spoof recognizers [2, 8]. Impersonation, in turn, is a difficult attack since it requires special skills for mimicking a target speaker [3]. Speech synthesis involves artificial production of a target speaker's voice given a text input whereas voice conversion refers to modification of the speech signal of a source speaker as if it was spoken by the target speaker. Earlier, speech synthesis and voice conversion attacks have received only limited attention, possibly due to low synthesis quality or lack of standard evaluation datasets. However, recent developments in voice conversion and speech synthesis technology and mass-market adoption of speaker verification technology, have drawn increased attention to spoofing attacks [9, 10]. In [6, 7, 11, 12, 13], it has been independently reported that current systems are highly vulnerable to spoofing attacks based on speech synthesis and voice conversion.

Speaker recognition systems should be integrated with appropriate spoofing *countermeasures* to determine whether a speech signal is natural or synthetic/converted, in order to safeguard recognizers against attacks. There are a few studies which concentrate on the detection of natural and synthetic/converted speech signals. For example, in [14], the authors compared three different feature sets and reported EERs of 6.60% and 3.93% for GMM-based and unit selection based converted speech detection, respectively. In [15], four different sets of features including standard mel-frequency cepstral coefficients (MFCCs) were compared in synthetic speech detection task using Gaussian mixture model (GMM) classifier, yielding EER of 10.98% with MFCCs whereas tailored group delay features reduced EER further down to 1.25%. In [16], EER of 2.7% to discriminate converted speech and natural speech was reported. In a more recent study [17], an i-vector system performing speaker verification and spoof detection jointly against voice conversion attacks was proposed with promising results.

Previous studies on spoof detection mostly utilize standard GMM trained using maximum likelihood (ML) criterion [18] classifier and focus on the feature extraction based on the prior knowledge about the synthesis system to improve detection performance. However, robust *generalized* countermeasures are desired to detect various types of attacks with limited prior knowledge about the vocoder and synthesis techniques. Thus, a thorough analysis on classifiers is necessary for the anti-spoofing research. In this paper, we make first attempts towards this goal by comparing five different classifiers for synthetic/converted speech detection used in speaker and language recognition. Besides comparison of different classifiers, we study their parameters as well for generalization of countermeasures for various attacks.

2. Synthetic Speech Detection

Given a speech signal, S, spoofing detection – here, determining whether S is a natural or synthetic/converted speech – can be cast as a hypothesis test,

- H_0: S is natural speech
- H_1: S is synthetic/transformed speech

Therefore, likelihood ratio test can be applied to decide between H_0 and H_1. Suppose that $\mathbf{X} = \{\mathbf{x}_1, \ldots \mathbf{x}_\mathrm{T}\}$ are the feature vectors extracted from S, then the logarithmic likelihood ratio score is given by

$$\Lambda(\mathbf{X}) = \log p(\mathbf{X}|\lambda_{H_0}) - \log p(\mathbf{X}|\lambda_{H_1}). \quad (1)$$

In (1), λ_{H0} and λ_{H1} are the acoustic models to characterize the hypotheses. The parameters of these models are estimated using training data for natural and synthetic/converted speech. In this section, the classifiers used for synthetic/converted speech detection are briefly described.

2.1. Gaussian Mixture Models

Gaussian mixture model (GMM) is a widely used generative model in speech processing [18]. It represents each class as a weighted sum of M multivariate Gaussians, $p(\mathbf{x}|\lambda) =$

$\sum_{i=1}^{M} w_i p_i(\mathbf{x})$, where w_i is the ith mixture weight and $p_i(\mathbf{x})$ is a D-variate Gaussian density function with mean vector $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{\Sigma}_i$. The model parameters are denoted by $\lambda = \{w_i,\ \boldsymbol{\mu}_i,\ \boldsymbol{\Sigma}_i\}_{i=1}^{M}$.

Expectation-maximization (EM) algorithm [18, 19] is used to estimate the parameters of each class independently via maximum likelihood (ML) criterion. In the test phase, given the models, λ_{nat} and λ_{synth}, and feature vectors of the test utterance, $\mathbf{Y} = \{\mathbf{y}_1,\ \ldots \mathbf{y}_T\}$, the detection score is computed as,

$$\Lambda(\mathrm{Y}) = \mathcal{L}(\mathbf{Y}|\lambda_{\text{nat}}) - \mathcal{L}(\mathbf{Y}|\lambda_{\text{synth}}), \tag{2}$$

where $\mathcal{L}(\mathbf{Y}|\lambda) = (1/T)\sum_{t=1}^{T} \log p(\mathbf{y}_t|\lambda)$ is the average log-likelihood of $\mathbf{Y}$ given GMM model λ. λ_{nat} and λ_{synth} are the GMMs for natural and synthetic classes, respectively.

Another common parameter estimation for GMMs is maximum a *posteriori* (MAP) adaptation of a *universal background model* (UBM) trained on a large amount of speech data from many speakers, popularly known as GMM-UBM [20]. The UBM represents a general distribution of the acoustic feature space while the target models, λ_{nat} and λ_{synth}, are obtained via MAP adaptation of the UBM. The mean vectors of the target models are obtained as $\hat{\boldsymbol{\mu}}_i = \alpha_i E_i(\mathbf{x}) + (1-\alpha_i)\boldsymbol{\mu}_i^{\text{ubm}}$. Here, $\alpha_i = n_i/(n_i + r)$ is the adaptation coefficient, n_i is the probabilistic count and $E_i(\mathbf{x})$ is the first order sufficient statistics for the ith Gaussian and r is a *relevance factor*. $r = 0$ corresponds to standard ML parameter estimation with one EM iteration using the UBM as initial model. As r increases, the Gaussians that are closer to the training data are adapted and the remaining components remain unchanged. In the recognition phase, detection score is computed using (2) as above.

2.2. GMM supervectors

Support vector machine (SVM) [21] is a well-known discriminative classifier used extensively in speaker and language recognition [22]. It models the decision boundary between two classes as a separating hyperplane optimized to maximize the margin of separation. In speaker recognition, SVM is generally combined with the GMM (*GMM supervector*) [23]. First, the set of feature vectors extracted from a speech signal is represented with a single high-dimensional vector obtained by concatenation of mean vectors of MAP-adapted GMM. Those supervectors are normalized using the covariance and the weights of UBM and then used as input features to SVM back-end.

In synthetic speech detection with GMM supervectors, one class consists of the training supervectors of natural speech (labeled $+1$) and the other class consists of those of synthetic/converted speech (labeled -1). SVM training yields a set of support vectors, $\mathbf{b}_i$, their weights α_i and a bias term d. All these outputs are collapsed into a single model vector $\mathbf{w} = \sum_{i=1}^{L} \alpha_i t_i \mathbf{b}_i + \mathbf{d}$ where $t_i \in \{+1, -1\}$ are the ideal outputs (class labels of each support vector), $\mathbf{d} = [d\ 0\ \ldots\ 0]^\top$ and L is the total number of support vectors.

In the test phase of GMM-supervector approach, the detection score between the test supervector $\mathbf{b}$ and SVM model vector $\mathbf{w}$ is computed as the inner product $\mathbf{w}^\top\mathbf{b}$.

2.3. GLDS-SVM

In *generalized linear discriminant sequence kernel* SVM (GLDS-SVM) system [22], feature vectors are mapped to higher dimensional space by a polynomial expansion up to a certain maximum degree m. For a $D-$dimensional feature vector, the dimensionality of expanded vectors is $\binom{D+m}{m} = (D+m)!/(D!m!)$. Given a set of feature vectors, $\mathbf{X} = \{\mathbf{x}_1,\ \ldots \mathbf{x}_T\}$, it is represented by average expanded vectors $\mathbf{b} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{b}(\mathbf{x}_t)$ where $\mathbf{b}(\mathbf{x}_t)$ denotes the expansion of the feature vector $\mathbf{x}_t$.

Training the linear SVM model with GLDS kernel using expanded feature vectors and scoring are performed as in GMM-SVM. The advantage of GLDS-SVM over GMM-SVM in synthetic speech detection is that it doesn't require additional data or model (i.e. UBM in GMM-SVM) to compute high-dimensional supervectors.

2.4. I-vector System

The so-called I-vector technique has become a modern *de-facto* standard in speaker recognition [24]. Recently, it has been used for speaker verification and spoof detection jointly against voice conversion attacks in [17]. It extracts a low-dimensional vector, $\mathbf{w}$, called an i-vector, from a speech signal S. A GMM mean supervector is factorized as $\boldsymbol{\mu} = \mathbf{m} + \mathbf{T}\mathbf{w}$, where $\boldsymbol{\mu}$ is the GMM mean supervector, $\mathbf{T}$ is a low-rank rectangular matrix and $\mathbf{w}$ is a low-dimensional i-vector with a prior distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The $\mathbf{T}$ matrix is trained using the EM algorithm and serves as i-vector extractor as detailed in [24].

The extracted i-vectors are pre-processed by applying *within-class covariance normalization* (WCCN) [25] followed by *length normalization* (LN) [26]. In speaker recognition, WCCN normalizes within-speaker variation [24]. In synthetic speech detection, in contrast, we use WCCN to normalize within-class (natural or synthetic) variation caused by changes in speaker or synthesis methods, for instance. To this end, the WCCN transformation matrix, $\mathbf{B}$ in [24], is computed from the training data of each class (natural or synthetic) and used for normalizing the i-vectors. Length normalization [26] is applied to project i-vectors to the unit sphere.

When multiple training utterances are available in i-vector system, each class can be represented by its average training i-vector as $\hat{\mathbf{w}}_{\text{nat}} = (1/J)\sum_{j=1}^{J} \mathbf{w}_{\text{nat}}^j$, where J is the total number of training utterances for natural class and $\mathbf{w}_{\text{nat}}^j$ is the i-vector extracted from the jth training utterance. Average target i-vector, $\hat{\mathbf{w}}_{\text{synth}}$ is similarly computed for synthetic speech.

In the recognition step, *cosine similarity* measure between the i-vector extracted from a test utterance, $\mathbf{w}_{\text{tst}}$ and the target i-vector $\mathbf{w}_{\text{tgt}}$ is computed as [24]:

$$\text{score}(\mathbf{w}_{\text{tgt}}, \mathbf{w}_{\text{tst}}) = \frac{\mathbf{w}_{\text{tgt}}^{\mathrm{T}}\mathbf{w}_{\text{tst}}}{\|\mathbf{w}_{\text{tgt}}\|\|\mathbf{w}_{\text{tst}}\|} = \mathbf{w}_{\text{tgt}}^{\top}\mathbf{w}_{\text{tst}}. \tag{3}$$

where $\|\mathbf{w}_{\text{tgt}}\| = \|\mathbf{w}_{\text{tst}}\| = 1$ due to LN. Given a test i-vector, $\mathbf{w}_{\text{tst}}$, the detection score is computed as:

$$\text{score}_{\text{final}} = \text{score}(\hat{\mathbf{w}}_{\text{nat}}, \mathbf{w}_{\text{tst}}) - \text{score}(\hat{\mathbf{w}}_{\text{synth}}, \mathbf{w}_{\text{tst}}). \tag{4}$$

where, $\hat{\mathbf{w}}_{\text{nat}}$ and $\hat{\mathbf{w}}_{\text{synth}}$ represent the average training i-vectors for natural and synthetic speech classes, respectively. Another method, when multiple training i-vectors are available, is score averaging over all training i-vectors of each class [27], i.e. $\text{score}_{\text{avg}}^{\text{nat}} = (1/J)\sum_{j=1}^{J} \text{score}(\mathbf{w}_{\text{nat}}^j, \mathbf{w}_{\text{tst}})$ where $\text{score}(\mathbf{w}_{\text{nat}}^j, \mathbf{w}_{\text{tst}})$ is the cosine similarity defined in (3) between the jth training i-vector of natural class, $\mathbf{w}_{\text{nat}}^j$, and the test i-vector, $\mathbf{w}_{\text{tst}}$. The final detection score is the difference between average score of natural class and that of synthetic class as defined in (4).

Different from the aforementioned scoring methods in i-vector system, another possible technique is to train an SVM model using the training i-vectors of natural and synthetic classes and then computing the detection score as dot product of SVM model vector and test i-vector.

3. Experimental Setup

3.1. Database

The experiments are conducted on ASVspoof 2015 database which consists of three subsets without target speaker overlap: *Training*, *Development* and *Evaluation*. The training subset consists of natural and synthetic utterances to be used for training the models for natural and synthetic classes. Synthetic utterances are generated using one of three voice conversion (S1, S2 and S5) and two speech synthesis methods (S3 and S4). The development set contains synthetic utterances generated using the same five methods (S1-S5). The evaluation subset, in turn, consists of synthetic utterances from the same five methods used in training and development subsets but also five new *unknown* methods. More details about the database, voice conversion/speech synthesis methods, recording conditions and number of trials and speakers can be found in [28].

3.2. Performance Measure

Equal error rate (EER) is used as the objective performance criterion. It corresponds to the error rate for the threshold at which the false alarm (P_{fa}) and the miss rate (P_{miss}) are equal. The reported EERs are computed using the Bosaris toolkit [29]. In the experiments on development set, we provide EERs of each speech synthesis/voice conversion methods (S1-S5) and the average value of these five error rates. In the evaluation set, in turn, we provide the average EERs for five known methods (S1-S5) and unknown methods (S6-S10).

3.3. Feature Extraction

Standard MFCC features are used in the experiments. While our companion paper [30] demonstrates that these may not be the optimal features for synthetic speech detection task, they are the standard features in speaker verification and provide still low error rates on ASVspoof 2015. In the experiments, 26 dimensional MFCCs and energy features with delta and double delta coefficients are used as the acoustic features. 80 dimensional features by excluding the static energy coefficient (c_0) are used. Simple energy based voice activity detection (VAD) is used to detect and drop non-speech frames [31, p. 24].

3.4. Classifiers

In the experiments, we use five different methods: GMM-ML, GMM-UBM, GMM-SVM, GLDS-SVM and i-vector approach. GMMs with diagonal covariance are trained using 10 EM iterations. Gender-independent UBM is trained using total of 9000 utterances from 150 male and 150 female speakers from WSJ0 and WSJ1 databases [32]. The **T**-matrix, for the i-vector system, is trained using 35704 utterances from 178 male and 177 female speakers selected from WSJ0 and WSJ1 corpora. LIBSVM package [33] is used to train SVM models for GMM-SVM, GLDS-SVM and SVM back-end using i-vector systems.

4. Results

We first optimize the number of Gaussian components used to train natural and synthetic speech models with GMM-ML classifier. Average EERs (%) for different number of Gaussian components are summarized in Table 1. The smallest average EER (0.65%) is obtained with 1024 Gaussians per class. EER rapidly decreases for fewer Gaussians up to 128 components, but slight changes occur afterwards. We fix it to 1024 in the remaining experiments.

Table 1: *Average EERs (%) for different number of Gaussians on development set using GMM-ML classifier.*

# Gauss.	EER (%)	# Gauss.	EER (%)
4	11.05	128	1.23
8	8.27	256	0.91
16	3.25	512	0.73
32	2.51	1024	**0.65**
64	1.97	2048	0.68

4.1. GMM-UBM Results

In the GMM-UBM system, besides the number of Gaussians, the other control parameter requiring optimization is the relevance factor, r, for adapting the component means. In speaker recognition, it is usually selected between $8 \leq r \leq 16$. As we are not aware of previous studies on the effect of r in synthetic speech detection, we study it in Table 2. Interestingly, $r = 0$ yields the smallest EERs. This could possibly be because of the retained Gaussian components without adaptation ($r > 0$ case) which are shared by the UBM and the target models. In speaker recognition, since the likelihood ratio between the target speaker model and the UBM is used as the detection score, effects of retained Gaussians are compensated in the score level. However, in synthetic speech detection, the detection score is computed using natural and synthetic GMMs and the retained components are different for each model. Therefore unadapted components show negative impact on the score level. Thus, adapting all the components ($r = 0$) according to training data gives better performance.

Table 2: *EERs (%) on the development set for different values of r used in MAP adaptation in GMM-UBM system.*

r	S1	S2	S3	S4	S5	Avg.
0	**0.09**	**1.74**	**0.00**	**0.00**	**0.70**	**0.51**
2	0.10	1.78	0.01	0.00	0.73	0.52
4	0.10	1.80	0.01	0.00	0.76	0.53
6	0.10	1.84	0.01	0.00	0.79	0.55
8	0.11	1.88	0.01	0.00	0.81	0.56
10	0.11	1.90	0.01	0.00	0.85	0.57

4.2. GMM-SVM Results

GMM-SVM results with different number of Gaussians are summarized in Table 3. Relevance factor, $r = 0$, is used for computing the mean supervectors. Similar to GMM-ML, UBM with 1024 Gaussians gives the smallest average EER. This is probably because of the choice $r = 0$. In our experiments it was found that when large r is used, fewer Gaussians gives higher accuracy, as expected. For example, average EERs of 1.23% and 1.73% were obtained for 16 and 512 Gaussians, respectively with $r = 2$. However, similar to GMM-UBM, $r = 0$ shows the best performance.

Table 3: *EERs (%) for each spoofing attack on the development set using UBMs with different number of Gaussians in GMM-SVM system.*

# Gauss.	S1	S2	S3	S4	S5	Avg.
32	0.56	1.14	0.47	0.49	1.20	0.77
64	0.59	1.33	0.38	0.37	1.10	0.75
128	0.34	0.99	0.24	0.26	0.75	0.52
256	**0.24**	0.89	0.18	**0.18**	0.53	0.41
512	0.31	0.73	0.15	0.20	0.52	0.38
1024	0.28	**0.71**	**0.14**	0.18	**0.51**	**0.36**

4.3. GLDS-SVM Results

In the experiments with GLDS-SVM, we evaluate three different polynomial expansion orders, $m = 1$, $m = 2$ and $m = 3$ (see Table 4). As expected, $m = 1$ provides poor performance since 1st order expansion corresponds to time averaging of MFCCs. The lowest EERs are obtained when 3rd order expansion is used. One may claim that further increasing the polynomial expansion would improve accuracy. However, using a 4th order expansion will yield GLDS supervectors of dimensionality 1929501. Given that we have 16375 training utterances, we found it computationally impractical to train SVMs using 4th order expansion in our Linux server.

Table 4: *EERs (%) on the development set for different expansion orders (m) in GLDS-SVM system.*

m	S1	S2	S3	S4	S5	Avg.
1	10.49	9.45	9.07	9.20	13.03	10.25
2	0.27	0.43	0.33	0.31	1.12	0.49
3	**0.02**	**0.14**	**0.02**	**0.06**	**0.38**	**0.12**

4.4. I-vector Results

In the experiments on the development set with i-vector system, we first train UBMs with different number of Gaussians to determine the best configuration for synthetic speech detection task. Length normalized 400 dimensional i-vectors are used in these preliminary experiments and the average EERs for different scoring methods described in Section 2.4 are shown in Table 5. UBM consisting of 512 Gaussians yields the smallest EERs for i-vector and score averaging methods. However for i-vector scoring based on SVM back-end, 128 Gaussians give slightly smaller EER. In general, SVM back-end is superior to cosine scoring. Next, the number of Gaussians is fixed to 512 and the i-vector dimensionality is varied. Average EERs of 16.38%, 10.04% and 9.60% are obtained using 200, 400 and 600 dimensional i-vectors, respectively, using cosine scoring with i-vector averaging.

Table 5: *Average EERs (%) using UBMs with different number of Gaussians on development set with I-vector system (400 dimensional length-normalized i-vectors are used).*

# Gauss.	SVM	I-vector Avg.	Score Avg.
64	5.81	15.94	15.99
128	**5.59**	12.16	12.12
256	5.85	13.61	13.56
512	5.73	**10.04**	**9.94**
1024	6.94	12.17	12.06

The EERs when WCCN is applied to 600 dimensional length-normalized i-vectors are given in Table 6. Applying WCCN yields 75% relative improvement over the baseline cosine scoring (EER reduced from 9.60% to 2.37%). This could be because the success of WCCN for normalizing the within-class variations caused by changes in speech synthesis/voice conversion techniques. SVM shows considerably better performance than that of cosine scoring without WCCN whereas cosine scoring yields slightly better accuracy when WCCN is applied.

In the last experiment on development set, we apply linear score fusion for all the seven systems utilized in the experiments (GMM-ML, GMM-UBM, GMM-SVM, GLDS-SVM and three i-vector systems) with their optimum parameters. The Bosaris toolkit [29] is used to train the fusion weights. The EERs after score fusion are shown in Table 7.

Table 6: *Average EERs (%) with/without WCCN on development set using 600 dimensional length-normalized i-vectors.*

WCCN	SVM	I-vector Avg.	Score Avg.
–	4.84	9.60	9.60
✓	**2.61**	**2.37**	**2.40**

Table 7: *EERs (%) for the development set after score fusion.*

S1	S2	S3	S4	S5	Avg.
0.00	0.09	0.00	0.00	0.12	0.04

4.5. Results On Evaluation Set

The results on evaluation set with optimized parameters for each classifier are given in Table 8. GLDS kernel using SVM again yields the smallest EER for known attacks on evaluation set. However, for the unknown attacks, GMM-ML produces the lowest EER. In general, generative models (GMM-ML and GMM-UBM) outperform our discriminative classifiers (GMM-SVM and GLDS-SVM) for unknown attacks. Since we have enough amount of training data for natural and synthetic speech classes, GMM parameter estimation successfully captures the distribution of the classes in the feature space. When features from an unseen acoustic class appear in the recognition phase, it will yield low likelihood ratio score given in (1) because neither natural nor the synthetic class are emphasized in the score level for the data from an unknown acoustic class. Another interesting observation from Table 8 is that, score fusion improves the accuracy for known attacks in comparison to best individual system GLDS-SVM whereas its effects for unknown attacks are controversial. The fusion weights, trained on the development data, may inaccurately balance classifiers for unseen attacks.

Table 8: *Average EERs (%) for known and unknown attacks on evaluation set.*

Classifier	Known	Unknown	Avg.
GMM-ML	0.50	**5.52**	**3.01**
GMM-UBM	0.40	6.61	3.50
GMM-SVM	0.26	6.98	3.62
GLDS-SVM	**0.11**	9.40	4.75
I-vector (SVM)	2.66	9.78	6.22
I-vector Avg.	2.46	9.41	5.94
I-vector Score Avg.	2.45	9.41	5.93
Fused	0.04	7.38	3.71

5. Conclusion

We compared five different classifiers for synthetic speech detection task using the ASVspoof 2015 dataset. Our experimental results using standard MFCC features indicate that classifiers used in speaker and language recognition give promising results on synthetic/converted speech detection. On the development set, discriminative methods (GLDS-SVM and GMM-SVM) outperformed generative methods (GMM-ML and GMM-UBM) but the opposite was observed in the evaluaton set, particularly for unknown attacks. Interestingly, state-of-the-art speaker recognition method, i-vector, yields the highest EERs in both development and evaluation sets. Applying WCCN yields considerable improvement in the i-vector system. Finally, we found that detection of synthetic speech (S3 and S4) was easier than that of converted speech (S1, S2 and S5) independent from the classifier.

6. Acknowledgements

This work was funded from Academy of Finland (proj. no. 253120 and 283256).

7. References

[1] A. K. Jain and K. Nandakumar, "Biometric authentication: System security and user privacy," *IEEE Computer*, vol. 45, no. 11, pp. 87–92, 2012.

[2] J. Vilalba and E. Lleida, "Speaker verification performance degradation against spoofing and tampering attacks," in *Proc. FALA*, 2010, pp. 131–134.

[3] R. G. Hautamäki, T. Kinnunen, V. Hautamäki, T. Leino, and A. Laukkanen, "I-vectors meet imitators: on vulnerability of speaker verification systems against voice mimicry," in *Proc. INTERSPEECH*, 2013, pp. 930–934.

[4] M. Farrús, M. Wagner, J. Anguita, and J. Hernando, "How vulnerable are prosodic features to professional imitators?" in *Proc. Odyssey*, 2008, p. 2.

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

[6] D. Matrouf, J. Bonastre, and C. Fredouille, "Effect of speech transformation on impostor acceptance," in *Proc. ICASSP*, 2006, pp. 933–936.

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

[8] Z. Wu, S. Gao, E. S. Cling, and H. Li, "A study on replay attack and anti-spoofing for text-dependent speaker verification," in *Proc. APSIPA*, 2014, pp. 1–5.

[9] N. W. D. Evans, T. Kinnunen, J. Yamagishi, Z. Wu, F. Alegre, and P. L. D. Leon, "Speaker recognition anti-spoofing," in *Handbook of Biometric Anti-Spoofing - Trusted Biometrics under Spoofing Attacks*, 2014, pp. 125–146.

[10] 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.

[11] P. L. D. Leon, I. Hernez, I. Saratxaga, M. Pucher, and J. Yamagishi, "Detection of synthetic speech for the problem of imposture," in *Proc. ICASSP*, 2011, pp. 4844–4847.

[12] F. Alegre, R. Vipperla, N. W. D. Evans, and B. G. B. Fauve, "On the vulnerability of automatic speaker recognition to spoofing attacks with artificial signals," in *Proc. EUSIPCO*, 2012, pp. 36–40.

[13] T. Kinnunen, Z. Wu, K. Lee, F. Sedlak, E. Chng, and H. Li, "Vulnerability of speaker verification systems against voice conversion spoofing attacks: The case of telephone speech," in *Proc. ICASSP*, 2012, pp. 4401–4404.

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

[15] Z. Wu, X. Xiao, E. Chng, and H. Li, "Synthetic speech detection using temporal modulation feature," in *Proc. ICASSP*, 2013, pp. 7234–7238.

[16] F. Alegre, A. Amehraye, and N. W. D. Evans, "Spoofing countermeasures to protect automatic speaker verification from voice conversion," in *Proc. ICASSP*, 2013, pp. 3068–3072.

[17] A. Sizov, E. Khoury, T. Kinnunen, and Z. W. S. Marcel, "Joint speaker verification and anti-spoofing in the i-vector space," *IEEE Trans. Information Forensics and Security*, no. 99, 2015.

[18] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Trans. Speech and Audio Processing*, vol. 3, no. 1, pp. 72–83, 1995.

[19] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society: Series B*, vol. 39, pp. 1–38, 1977.

[20] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[21] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995.

[22] W. M. Campbell, J. P. Campbell, D. A. Reynolds, E. Singer, and P. A. Torres-Carrasquillo, "Support vector machines for speaker and language recognition," *Computer Speech & Language*, vol. 20, no. 2-3, pp. 210–229, 2006.

[23] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 308–311, 2006.

[24] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech & Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[25] A. O. Hatch, S. S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for svm-based speaker recognition," in *Proc. ICSLP*, 2006.

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

[27] P. Rajan, A. Afanasyev, V. Hautamäki, and T. Kinnunen, "From single to multiple enrollment i-vectors: Practical PLDA scoring variants for speaker verification," *Digital Signal Processing*, vol. 31, pp. 93–101, 2014.

[28] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilçi, M. Sahidullah, and A. Sizov, "ASVspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge," in *accepted to INTERSPEECH*, 2015.

[29] "Bosaris toolkit [software package]," [Online:] https://sites.google.com/site/bosaristoolkit, 2015.

[30] M. Sahidullah, T. Kinnunen, and C. Hanilçi, "A comparison of features for synthetic speech detection," in *accepted to INTERSPEECH*, 2015.

[31] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech Communication*, vol. 52, no. 1, pp. 12–40, 2010.

[32] "Wall Street Journal Corpus," [Online:] http://www.ldc.upenn.edu, 2015.

[33] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.