Int J Speech Technol (2016) 19:585–591
DOI 10.1007/s10772-016-9352-6

CrossMark

Audio steganalysis using deep belief networks

Catherine Paulin[1] · **Sid-Ahmed Selouani**[1] · **Éric Hervet**[2]

Received: 28 October 2015 / Accepted: 6 July 2016 / Published online: 21 July 2016

Abstract This paper presents a new steganalysis method that uses a deep belief network (DBN) as a classifier for audio files. It has been tested on three steganographic techniques: StegHide, Hide4PGP and FreqSteg. The results were compared to two other existing robust steganalysis methods based on support vector machines (SVMs) and Gaussian mixture models (GMMs). Afterwards, another classification task aiming at identifying the type of steganographic applied or not to the speech signal was carried out. The results of this four-way classification show that in most cases, the proposed DBN-based steganalysis method gives higher classification rates than the two other steganalysis methods based on SVMs and GMMs.

Keywords Audio steganography · Audio steganalysis · DBN · MFCCs · SVMs · GMMs

1 Introduction

Steganography is the art of hiding secret messages into digital covers such as images, audio or video files, etc. A secret message is hidden in a way that it is undetectable to anyone who does not know it is there. Thus, it is widely used as a secure mean of communication between two parties. A third party intercepting a message between the first two parties has no means of knowing that the message contains secret information. A file containing a hidden message is called the carrier.

Steganalysis aims at detecting secret messages embedded in data through steganography. There are two kinds of steganalysis implementations: Active steganalysis and passive steganalysis. In active steganalysis, the carrier's messages are modified to stop the communication if a third party detects a secret message. In passive steganalysis, the third party examines the data to determine if it contains a hidden message or not, and stops it from being communicated if there is a secret message in it. A message containing secret information is referred to as marked, and one without any secret information is referred to as unmarked. This paper presents an implementation of a new passive steganalysis approach with audio carriers.

Audio or speech steganalysis consists of performing an acoustic analysis of the input signals to get the best parameters to represent them. These parameters are used as inputs to a classifier so that the signal is classified as marked or unmarked. This is illustrated in Fig. 1.

In Ozer et al. (2003), a combination of audio quality metrics and SVMs is used to build the steganalyzer. It has been tested on four watermarking and two steganographic techniques. First, each method is tested individually with results ranging from 87 and 100 %. Then the detection for the ensemble of the watermarking techniques is tested with a detection rate of 69 %, and the ensemble of the steganographic techniques with a detection rate of 73 %.

In Johnson et al. (2005), Principal Component Analysis (PCA) is performed to get a statistical model of audio files. The signals are first decomposed with short-time Fourier

✉ Catherine Paulin
ecp8266@umoncton.ca

Sid-Ahmed Selouani
sid-ahmed.selouani@umoncton.ca

Éric Hervet
eric.hervet@umoncton.ca

[1] Campus de Shippagan, Université de Moncton, Moncton, NB, Canada

[2] Campus de Moncton, Université de Moncton, Moncton, NB, Canada

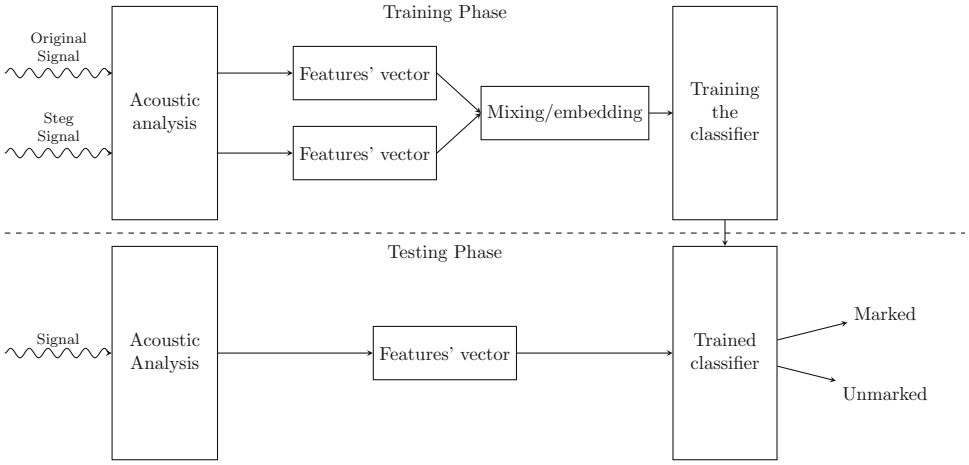

Fig. 1 A block diagram illustrating the steganalysis process

Transforms, then statistical regularities are extracted using PCA to produce the input to SVMs. This technique is tested on two steganographic approaches: Least Significant Bit (LSB) and Hide4PGP with different capacities (Repp 1996). At full capacity, the detection rates are 100 % with LSB and 83.1 % with Hide4PGP.

In Altun et al. (2005), a new technique based on Diminishing Marginal Distortions (DMDs) is presented. It tests the signals with two watermarks tests and measures the morphological distortion induced by each test. Those distortions are used in a neural network for classification. Detection rates vary between 80 and 100 % depending on the strength of the embedded watermark.

In Kraetzer and Dittmann (2007), mel-frequency cepstral coefficients (MFCCs) are introduced to extract significant features from the speech signals. Those coefficients are used as input to a SVM classifier with radial basis function (RBF) kernels. This technique has been tested on a number of steganographic methods including StegHide (Hetzl 2003).

Following the work of Kraetzer and Dittmann (2007), Sung et al. designed second-order derivative-based MFCCs combined with SVMs using RBF kernels for classification Liu et al. (2009). Steganographic techniques tested with this method include Hide4PGP, LSB and StegHide. The results of this modified MFCC approach outperform those from Kraetzer and Dittmann (2007).

In Rekik et al. (2012), line spectral frequencies (LSFs) parameters are used to code the audio signals and use them in an autoregressive time delay neural network for classification. These techniques were tested on three steganographic techniques including stegHide and Hide4PGP on the Noizeus database (Hu and Loizou 2007). They were tested with 50 and 100 % of hiding capacity and got results varying between 52.45 and 82.73 %, with the best result at 100 % of hiding capacity.

In Janicki et al. (2014), a method using MFCCs and GMMs to classify audio signals for transcoding steganography (Transteg) is implemented. MFCC parameters are used first for acoustic analysis. Afterwards, it uses GMMs to create two models: One for the unmarked signals, and one for the marked signals. The probability of the MFCC parameters belonging to each model is then calculated for each signal. The two resulting probabilities are compared in order to classify the signal so that the probability of belonging to a model is the highest. This method has been tested on five speech corpus including TIMIT (Garofolo et al. 1993). Various configurations of Transteg were tested, with the highest detection probability at 94.6 %, and the lowest detectability at 63.3 %.

In Yürüklü et al. (2014), an approach that uses surrogate data based delay vector variance (DVV) features to detect the existence of a stego-signal is presented. It uses a combination of Sequential Forward floating search method (SFFS) and the SVMs as a classifier. This method has been tested on nine watermarking and steganographic techniques, including StegHide and Hide4PGP, using the TIMIT corpus. It gives good results with the highest detection probability at 100 % and the lowest at 77.3 %.

In Ghasemzadeh and Arjmandi (2014), a combination of Reversed-Mel cepstrum Based Audio Steganalysis and SVMs is performed to propose a new steganalysis method. The method has been tested on two steganographic techniques, Hide4PGP and StegHide, and two watermarking methods. The classification rates vary between 91 and 99 %.

In this paper, we propose to use the mel-frequency cepstral coefficients for the acoustic analysis and a deep belief network as a steganalyzer.

The rest of this paper is organized as follows: Sect. 2 describes steganographic techniques used to test the new steganalysis tool. Sect. 3 presents the deep belief network theory, while Sect. 4 proceeds with the description of the

experiment methodology and the discussion of the obtained results. Section 5 concludes the paper.

2 Steganographic techniques

Three techniques of message embedding are tested in this paper. In the next sections, we present these steganographic techniques.

2.1 StegHide

StegHide has been widely used in steganalysis domain and is a free software available online (Hetzl 2003). StegHide hides data in a cover signal by altering its least significant bit (LSB) (Artz 2001). It can be used on JPEG, BMP, WAV and AU files. In this paper, it will be used to hide a signal in WAV files at its full capacity.

2.2 Hide4PGP

Hide4PGP is another free software that has been widely used in steganalysis domain (Repp 1996). It can embed large messages in WAV and BMP by spreading the data evenly throughout the file. In WAV files, it can hide 4 bits/sample. It has also been used to hide signal in WAV files at its full capacity in this paper.

2.3 FreqSteg

The third steganographic technique hides secret information in the high frequencies of the carrier signal. It will be reffered to as FreqSteg for the rest of this paper. FreqSteg relies on the fact that human ears cannot detect frequencies higher than 18 KHz. For signals with a sample rate of 44.1 KHz, the highest frequency recorded is 22.05 KHz, therefore the secret message will be hidden between frequencies of 18 and 22.05 KHz.

Because the sampling rate of the databases we used was different from 22.05 KHz, the hiding process percentage has been modified following these steps:

1 Pass the carrier signal through a Low-Pass Filter with a cut-off frequency of 70 % of the original frequency.
2 Pass the secret signal through a Band-Pass Filter that keeps the frequency over 0.02 % of the original frequency.
3 Generate a cosine signal at a frequency of 93.75 % of the carrier's signal frequency.
4 Multiply the secret signal and modulate the signals together.
5 Sum the resulting signal from the previous step with the resulting carrier signal.

This technique is detailed in Swanson et al. (2002) and is illustrated in Fig. 2.

3 Deep belief network

A deep belief network (DBN) is one of the main components of deep learning. It is a feed-forward neural network with weights and neurons first initialized using stack restricted Boltzmann machines. In the next sections, we present the restricted Boltzmann machines, and then how to use them to get the initialized DBN.

3.1 Restricted Boltzmann machines

Restricted Boltzmann machines (RBM) is a neural network with two layers, one visible and one hidden as showed in Fig. 3.

The energy function of this system is given by:

$$E(\mathbf{x},\mathbf{h}) = -\mathbf{b}^t\mathbf{x} - \mathbf{c}^t\mathbf{h} - \mathbf{h}^t\mathbf{W}\mathbf{x} \quad (1)$$

where $\mathbf{x}$ and $\mathbf{h}$ represent respectively the units vector of the visible and hidden layers, and $\mathbf{b}$ and $\mathbf{c}$ are the bias for those respective layers. $\mathbf{b}^t$ is the transpose of $\mathbf{b}$. The weights between the two layers are represented by matrix $\mathbf{W}$. The probability distribution of this system is given by:

$$P(\mathbf{x},\mathbf{h}) = \frac{e^{-E(\mathbf{x},\mathbf{h})}}{Z} \quad (2)$$

where Z is the partition function of the system. It gives the energies' sum for all possible configurations of the system. It is defined by:

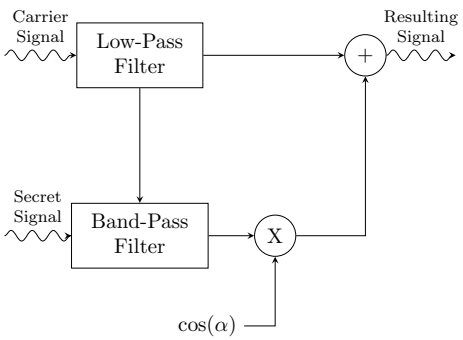

Fig. 2 Overview of the FreqSteg steganographic technique

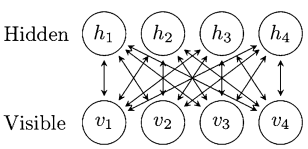

Fig. 3 Restricted Boltzmann machines

$$Z(\mathbf{x},\mathbf{h}) = \sum_{\mathbf{x},\mathbf{h}} e^{-E(\mathbf{x},\mathbf{h})} \tag{3}$$

The conditional probability of the hidden layer knowing the visible layer is:

$$\begin{aligned} P(\mathbf{h}|\mathbf{x}) &= \frac{e^{-E(\mathbf{x},\mathbf{h})}}{\sum_h e^{-E(\mathbf{x},\mathbf{h})}} \\ &= \frac{e^{\mathbf{b}^t\mathbf{x}+\mathbf{c}^t\mathbf{h}+\mathbf{h}^t\mathbf{W}\mathbf{x}}}{\sum_{\mathbf{h}} e^{\mathbf{b}^t\mathbf{x}+\mathbf{c}^t\mathbf{h}+\mathbf{h}^t\mathbf{W}\mathbf{x}}} \\ &= \frac{e^{\mathbf{c}^t\mathbf{h}+\mathbf{h}^t\mathbf{W}\mathbf{x}}}{\sum_{\mathbf{h}} e^{\mathbf{c}^t\mathbf{h}+\mathbf{h}^t\mathbf{W}\mathbf{x}}} \\ &= \frac{\prod_j e^{\mathbf{c}_j\mathbf{h}_j+\mathbf{h}_j\mathbf{W}_j\mathbf{x}}}{\prod_j \sum_{\mathbf{H}} e^{\mathbf{c}_j\mathbf{h}_j+\mathbf{h}_j\mathbf{W}_j\mathbf{x}}} \\ &= \prod_j P(\mathbf{h}_j|\mathbf{V}) \end{aligned} \tag{4}$$

In the same manner, we find that:

$$P(\mathbf{x}|\mathbf{h}) = \prod_i P(\mathbf{x}_i|\mathbf{h}) \tag{5}$$

Therefore, the activation probability of a hidden neuron is given by:

$$\begin{aligned} P(\mathbf{h}_j = 1|\mathbf{x}) &= \frac{e^{\mathbf{c}_j\cdot 1+1\cdot\mathbf{W}_j\mathbf{x}}}{e^{\mathbf{c}_j\cdot 0+0\cdot\mathbf{W}_j\mathbf{x}} + e^{\mathbf{c}_j\cdot 1+1\cdot\mathbf{W}_j\mathbf{x}}} \\ &= sigm(\mathbf{c}_j + \mathbf{W}_j\mathbf{x}) \end{aligned} \tag{6}$$

where *sigm* is the sigmoid function. The activation probability for a visible neuron is:

$$P(\mathbf{x}_j = 1|\mathbf{h}) = sigm(\mathbf{b}_i + \mathbf{W}_i\mathbf{h}) \tag{7}$$

As we train this model, we want it to generate data that resembles the training data. Therefore, we want to minimize the negative log probability of the training data:

$$\frac{\partial}{\partial\theta}(-logP(\mathbf{x})) = \frac{\partial}{\partial\theta}\left(-log\left(\sum_{\mathbf{h}} P(\mathbf{x},\mathbf{h})\right)\right) \tag{8}$$

By developing the second part of Equation 8, we get :

$$\begin{aligned} \frac{\partial}{\partial\theta}(-logP(\mathbf{x})) = &\sum_{\mathbf{h}} P(\mathbf{h}|\mathbf{x})\frac{\partial}{\partial\theta}E(\mathbf{x},\mathbf{h}) \\ &- \sum_{\mathbf{x},\mathbf{h}} P(\mathbf{x},\mathbf{h})\frac{\partial}{\partial\theta}E(\mathbf{x},\mathbf{h}) \end{aligned} \tag{9}$$

Therefore, with our parameters, we get:

$$\begin{aligned} \frac{\partial}{\partial\mathbf{W}}(-logP(\mathbf{x})) &= \sum_{\mathbf{h}} P(\mathbf{h}|\mathbf{x})(-\mathbf{h}^t\mathbf{x}) - \sum_{\mathbf{x},\mathbf{h}} P(\mathbf{x},\mathbf{h})(-\mathbf{h}^t\mathbf{x}) \\ \frac{\partial}{\partial\mathbf{b}}(-logP(\mathbf{x})) &= \sum_{\mathbf{h}} P(\mathbf{h}|\mathbf{x}(-\mathbf{x}) - \sum_{\mathbf{x},\mathbf{h}} P(\mathbf{x},\mathbf{h})(-\mathbf{x}) \\ \frac{\partial}{\partial\mathbf{c}}(-logP(\mathbf{x})) &= \sum_{\mathbf{h}} P(\mathbf{h}|\mathbf{x})(-\mathbf{h}) - \sum_{\mathbf{x},\mathbf{h}} P(\mathbf{x},\mathbf{h})(-\mathbf{h}) \end{aligned} \tag{10}$$

The first part of those equations is easily calculated because it is the sum on the hidden layer only. The second part is not as trivial to compute because we need the sum on all possible states. Therefore, to compute it, the Contrastive Divergence algorithm is used (Hinton et al. 2006). It consists of using the training data as the first iteration of the visible layer and then calculate the first iteration of the hidden layer with that visible layer. We then compute the second iteration of the visible layer with the first iteration of the hidden layer and finally the second iteration of the hidden layer with the second iteration of the visible layer. We use the first iteration of the visible and hidden layer for the first part of those equations, and the second iteration of the visible and hidden layer for the second part Palm (2012).

3.2 Building the DBN

The deep belief network is made by stacking RBMs together, one RBM per hidden layer desired. The first RBM is trained with the contrastive divergence where the visible layer is composed of the training data. The second RBM is trained with the contrastive divergence where the visible layer is the hidden layer from the last RBM trained, and so on. The RBM will then provide the initial weights of the DBN. For technical purposes, all hidden layers contain the same number of units. This network is trained afterwards as a feed-forward network with backpropagation training algorithm.

4 Experiments and results

4.1 Data

The experiments have been carried out on the Noizeus database. It is an English corpus of six speakers. Thirty sentences have been recorded and originally sampled at 25 KHz and downsampled at 8 KHz (Hu and Loizou 2007). These sentences are from 2 to 3 and a half seconds. After the embedding process on the corpus, an acoustic analysis is perfomed on each audio file. In this process, each signal is divided into 16 ms frames and will resort in a database of approximately 10,000 frames. Each frame will then be assigned into either the marked or the unmarked categorie. Afterwards, All frames will be mixted randomly and 2/3 will going into the training process. The rest will serve as the testing data.

4.2 Experimental methodology

In a first set of experiments, the steganographic techniques are tested individually. They are used to embed data in 50 % of the Noizeus database. They are hidden at the full

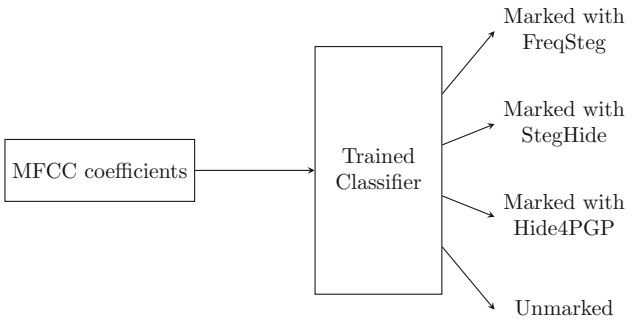

Fig. 4 Task aiming at identifying the steganographic technique

capacity of the cover files. Afterwards, the mel-frequency cepstrum is used to extract features from the resulting speech signals. The signals are framed into 16 ms and the hamming window is also applied. We tested a number of MFCC coefficients ranging from 10 to 25 and found the optimal number of MFCC coefficients for the DBN, the SVM and the GMM based steganalyzers.

The retained MFCC coefficients are the inputs for each classifier. These frames are classified into two groups, marked and unmarked.

In a second set of experiments, we combined the three steganographic techniques together to get a four-way classification. The classifiers are trained to detect which steganographic techniques the data was embedded with or if there is no data embedded. For this classification, we also tested a number of MFCC coefficients ranging from 10 to 25. This classification is illustrated in Fig. 4.

4.3 Experimental results

4.3.1 One-way classification

The results obtained after applying each technique can be divided into four categories:

1. *True positive (TP)* When the utterrance is classified as marked with the staganalysis tool and has the marked label.
2. *False positive (FP)* When the utterance is classified as marked with the staganalysis tool and has the unmarked label.
3. *True negative (TN)* When the utterance is classified as unmarked with the staganalysis tool and has the unmarked label.
4. *False negative (FN)* When the utterance is classified as unmarked with the staganalysis tool and has the marked label.

We present the results obtained on the Noizeus database. The tested MFCC coefficients from 10 to 25 are presented in Fig. 5. The classification rate clearly varies with differents MFCC coefficients which justified the need to get the best MFCC coefficients for each steganalyzer.

Table 1 provides the best results obtained by each steganalyzer corresponding to the optimal number of MFCCs. The GMM-based system reaches its best performance for 256 Gaussians. The numbers in bold face represent the highest classification rate obtained for each steganographic technique. The same goes for Table 3. The optimized parameters for the

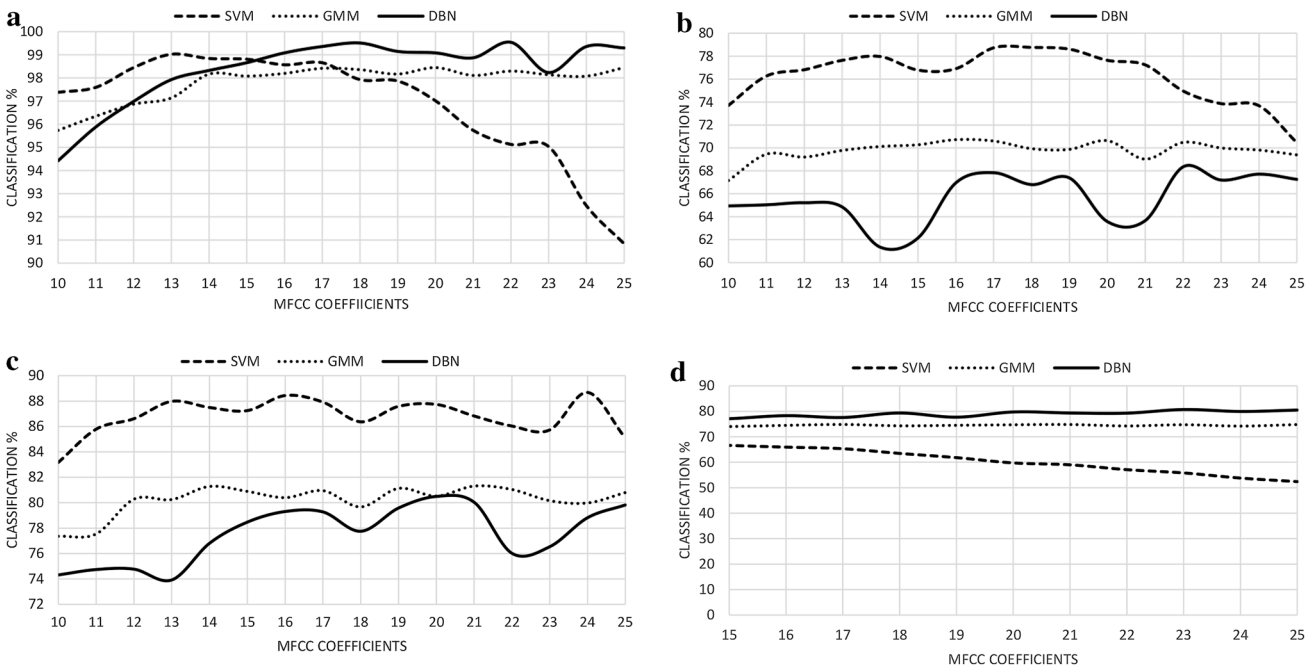

Fig. 5 Classification results for each steganographic technique using the Noizeus database

Table 1 Classification results obtained by SVM, GMM, and DBN steganalyzers after applying the three steganographic techniques on Noizeus speech utterances

Noizeus	Classifier	MFCCs	TP	TN	FP	FN	Total (%)
FreqSteg	SVMs	13	1649	1605	21	11	99.03
	GMMs	25	1632	1584	42	28	98.45
	DBN	22	1650	1617	9	10	**99.42**
StegHide	SVMs	18	1325	1263	363	335	**78.76**
	GMMs	16	1167	1122	504	493	70.72
	DBN	22	1254	1023	603	406	69.29
Hide4PGP	SVMs	24	1456	1458	168	204	**88.68**
	GMMs	21	1334	1307	319	325	81.31
	DBN	22	1327	1336	290	333	81.04

Table 2 Optimized parameters for the DBN-based steganalyzer optained with the cross-validation method

Steganographic techniques	Configuration	RBM			DBN			
	IL-HLs-OL	α	Momentum	Epoch	Epoch	Drop Out Rate	α	Momentum
FreqSteg	22-20-2	0	0	100	2000	0.1	2	0.5
StegHide	22-30-2	0	0.4	100	1000	0.2	2	0.5
Hide4PGP	22-40-2	0	0	100	1200	0.3	2	0.5
Four-Way	23-40-40-2	0	0.6	100	1000	0.2	2	0.3

DBN is obtained with the cross-validation method. They are presented in Table 2. In the configuration, IL represents the number of neurons in the Input layer, HLs represents the number of neurons for each hidden layers and finaly OL reprensents the number of neurons for the output layer (Table 2).

In the case of the FreqSteg steganographic technique, the DBN outperforms the GMMs and gives slightly better results than the SVMs. For StegHide and Hide4PGP, the SVMs outperforms the GMMs and DBN by at least 8 %. The DBN performs only slightly lower than GMMs.

4.3.2 *Four-way classification*

The results of combining all steganographic techniques to make a four-way classifier can be divided into eight categories:

1. *True FreqSteg (TFS)* When the utterrance is classified as FreqSteg with the staganalysis tool and has the FreqSteg label.
2. *False FreqSteg (FFS)* When the utterance is not classified as FreqSteg with the staganalysis tool but has the FreqSteg label.
3. *True StegHide (TSH)* When the utterance is classified as StegHide with the staganalysis tool and has the StegHide label.
4. *False StegHide (FSH)* When the utterance is not classified as StegHide with the staganalysis tool but has the StegHide label.
5. *True Hide4PGP (TH4)* When the utterrance is classified as Hide4PGP with the staganalysis tool and has the Hide4PGP label.
6. *False Hide4PGP (FH4)* When the utterance is not classified as Hide4PGP with the staganalysis tool but has the Hide4PGP label.
7. *True Unmark (TUM)* When the utterance is classified as unmarked with the staganalysis tool and has the unmark label.
8. *False Unmark (FUM)* When the utterance is not classified as unmarked with the staganalysis tool but has the Unmark label.

Table 3 Results of the four-way classification on Noizeus speech utterances obtained by SVM, GMM and DBN steganalysis systems

Classifier	TFS	FFS	TSH	FSH	TH4	FH4	TUM	FUM	Total (%)
SVMs	1530	130	768	892	717	943	1626	80	69.04
GMMs	1525	135	961	699	920	740	1539	87	74.86
DBN	1596	64	1076	584	1081	579	1579	47	**80.71**

These results are given in Table 3.

The SVMs got the highest result with 11 MFCCs. The GMMs gave the highest result with 512 Gaussians and 17 MFCCs. The DBN obtained its best performance with 23 MFCCs. The DBN-based steganalyzer outperforms the SVMs and the GMMs by at least 5 %.

5 Conclusion

In this paper, we presented a new steganalysis method based on the DBN to distinguish between marked and unmarked steganographic signals. It has been compared with two other steganalyzers: SVMs and GMMs. The first experiment consisted of testing these steganalyzers on three steganographic techniques: StegHide, Hide4PGP and FreqSteg. We also tested the capacity of the proposed DBN-based steganalyzer to automatically identify the type of steganographic technique applied to the original speech signals. In this complex task, the DBN outperfoms both the SVM and GMM based steganalyzers.

References

Altun, O., Sharma, G., Celik, M. U., Sterling, M., Titlebaum, E. L., & Bocko, M. (2005). Morphological steganalysis of audio signals and the principle of diminishing marginal distortions. *In ICASSP*, *2*, 21–24.

Artz, D. (2001). Digital steganography: hiding data within data. *IEEE Internet Computing*, *5*(3), 75–80.

Garofolo, J. S., et al. (1993). *TIMIT: acoustic-phonetic continuous speech corpus LDC93S1. Web download*. Philadelphia: Linguistic Data Consortium.

Ghasemzadeh, H., & Arjmandi, M. K. (2014). Reversed-mel cepstrum based audio steganalysis. In *2014 4th international eConference on computer and knowledge engineering (ICCKE)*, (pp. 679–684). IEEE.

Hetzl, S. (2003). *StegHide steganography*. http://www.steghide.sourceforge.net/.

Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, *18*(7), 1527–1554.

Hu, Y., & Loizou, P. C. (2007). Subjective comparison and evaluation of speech enhancement algorithms. *Speech Communication*, *49*(7), 588–601.

Janicki, A., Mazurczyk, W., & Szczypiorski, K. (2014). Steganalysis of transcoding steganography. *Annals of Telecommunications*, *69*(7–8), 449–460.

Johnson, M. K., Lyu, S., & Farid, H. (2005). Steganalysis of recorded speech. In *Proceedings of the electronic imaging 2005*, (pp. 664–672). International Society for Optics and Photonics.

Kraetzer, C., & Dittmann, J. (2007). Mel-cepstrum-based steganalysis for voip steganography. In *Proceedings of the electronic imaging 2007*. (pp. 664–672) . International Society for Optics and Photonics.

Liu, Q., Sung, A. H., & Qiao, M. (2009). Temporal derivative-based spectrum and mel-cepstrum audio steganalysis. *IEEE Transactions on Information Forensics and Security*, *4*(3), 359–368.

Ozer, H., Avcibas, I., Sankur, B., & Memon, N. D. (2003). Steganalysis of audio based on audio quality metrics. In *Proceedings of the electronic imaging 2003*. (pp. 55–66). International Society for Optics and Photonics.

Palm, R. B. (2012). Prediction as a candidate for learning deep hierarchical models of data. *Master's thesis, Technical University of Denmark*.

Rekik, S., Selouani, S.-A., Guerchi, D., & Hamam, H. (2012). An autoregressive time delay neural network for speech steganalysis. In *2012 11th international conference on information science, signal processing and their applications (ISSPA)*. (pp. 54–58). IEEE.

Repp, H. (1996). *Hide4PGP Steganography*. http://www.heinz-repp.onlinehome.de/Hide4PGP.htm.

Swanson, E., Ganier, C., Holman, R., & Rosser, J. (2002). *Freqency domain steganography*. https://www.clear.rice.edu/elec301/Projects01/smokey_steg/group.html.

Yürüklü, E., Koçal, O. H., & Dilaveroğlu, E. (2014). A new approach for speech audio steganalysis using delay vector variance method. *Uludağ Üniversitesi Mühendislik Fakültesi Dergisi*, *19*(1), 27–36.