https://doi.org/10.26552/com.C.2012.2.63-69

A. Utku Yargicoglu – H. Gokhan Ilk *

SPEECH CODER IDENTIFICATION USING CHAOTIC FEATURES BASED ON STEGANALYZER MODELS

In this study a steganalyzer model based on chaotic features is adapted to codec identification problem. As conventional steganalyzers detect suspected packages buried in bitstreams using statistical analysis, the proposed "speech coder identification" model also reveals the statistical differences of the outputs produced by different speech codecs. Essentially, the design of a coder/codec identifier is equivalent to a classifier training where the chaotic features extracted from the suspicious bitstream samples are used as inputs. During training, weight values which dissociate the codec types best are investigated. Finally, performances of the trained classifiers are evaluated by using test sets. According to the test results, for a closed group which is formed from nine different output types, polynomial SVM classifiers have identified more than 97% of the samples correctly.

Keywords: *Coder identification; codec recognition; suspicious bit stream*

1. Introduction

Codec or coder identification which is used interchangeably throughout this manuscript can be defined as identifying the type of codec from an unknown bit stream obtained from Internet, transmitted through air or stored in some media. Generally the type of codec, which will be used in voice, audio or video transmission, is nominated during the channel establishment stage. Anyone, who is able to monitor the transferred data and aware of the connection control protocol, can reveal the type of codec that is being used. In the case of storage media, voice data is usually stored in files which start with a header composed of attributes like the type of codec used. In some cases however there may be too many alternative connection control protocols that must be taken care of or monitoring of the bit stream might start after connection is established. For the storage media, file headers may be missing, corrupted or fake. For scenarios like these, it could still be possible to find the correct definition of the codec type being used by just analyzing the recorded bit stream.

The main motivation of this study is to employ steganalyzers, in order to identify speech coders, using bits obtained from an unknown source. During the training stage of a steganalyzer, firstly the statistical properties, which dissociate data hidden samples from regular data are investigated. Secondly for the "best" dissociated properties the threshold values, which separate the classes, are calculated. From the nature of the classifier design, decisions are either true positive (TP), false positive (FP), true negative (TN) or false negative (FN). For the steganalysis of suspicious speech, Kocal et al. [1] have proposed steganalyzers that decide by processing chaotic features. The outputs of different speech – voice coders would also have different statistical properties and as a result the design methodology of a speech steganalyzer can be adapted to speech codec identification problem.

2. Codec Identifier

As speech codec identification or speech codec type recognition is introduced to the literature with this paper, the term "codec identifier" needs to be defined. Codec identifier is a new generation bit analyzing method that is used to recognize the type of codec being used in a flowing or a stored bit stream. Note that codec identification can be performed for any bitstream which belongs to speech, audio or video. In this study the proof of concept is carried out on speech.

3. Design principles of the codec identifier

Codec identifier aims to recognize the type of codec used in a suspicious bit stream. Recognition can be handled using several different ways which all benefit from some sort of distinguishing information. However collecting, extracting and transforming all possible types of useful information may not be practical or feasible. In order to develop an implementable speech codec identifier model, some design criteria need to be specified along with the assumptions of the bit stream.

* **A. Utku Yargicoglu[1], H. Gokhan Ilk[2]**
[1] E-mail: auyargicoglu@yahoo.com.tr
[2] Electronics Engineering Department, Faculty of Engineering, Ankara University, 06100, Besevler, Ankara, Turkey, E-mail: ilk@ieee.or

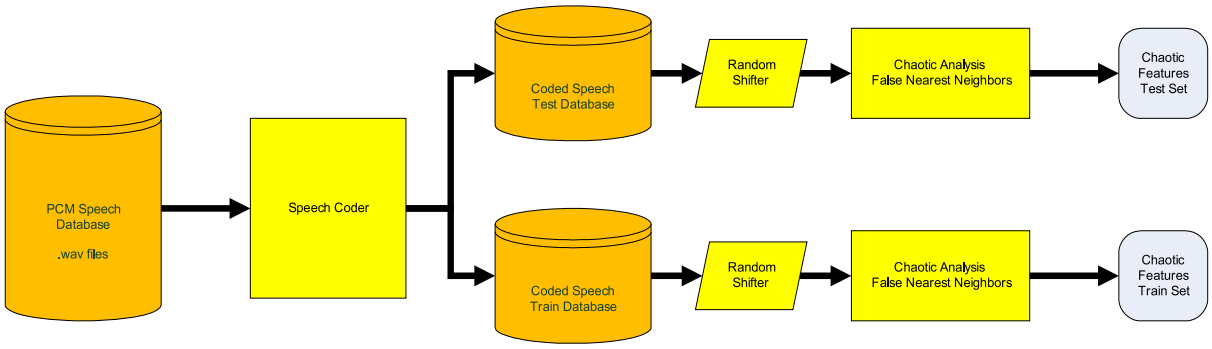

Fig. 1 Calculation of the chaotic features for a speech coder

Statistical Differences: It is assumed that different speech coders produce bit streams with different statistical properties.

Bit rate independency: Outputs of the coders are analyzed without any bit rate information. Different coders may produce different amounts of data in a unit amount of time, codec identifier proposed in this paper does not need this information, it only takes the output bit stream of the coder, divides bit stream into frames and analyzes these frames.

Bit allocation independency: Codec identifier analyzes output stream by only using the calculated chaotic features. The output of a speech coder is not a random sequence; bit fields that form the output bit stream should be grouped according to the information they convey, such as line spectral frequencies (LSF), pitch etc. The proposed method does not need or implicitly use this information. On the other hand, the order of grouping and/or logical relationships between these fields would have direct or indirect effects on the statistical properties and they should somehow reveal themselves in the calculated features.

4. Model Derivation from Steganalyzers

The idea behind the codec identification model design is quite simple: Any tool that is able to distinguish the statistical differences between the codec outputs should be able to detect the type of the codec used. Steganalyzers are sensitive to statistical differences and therefore codec identification should be possible by using existing steganalyzer models.

Codec identification is performed by using a general codec identifier model which is derived from the steganalyzer model given in [1]. In the steganalyzer model in question, statistical differences between two groups (group of samples which carry hidden data and group of samples which do not carry hidden data) are investigated in the chaotic feature domain. There are two types of features obtained from the evaluated samples: False Nearest Neighbors (FNN) [2] and Lyapunov exponents [3]. Existence of the hidden data may stretch any chaotic feature, since every data hiding method causes separation for these features and each data hiding method has its own specialized steganalyzer. The lengths NFNN FNN and NLY Lyapunov feature vectors vary according to dimension and delay [2, 3].

5. Feature Selection for Codec Identification

Codec identification is based on the statistical differences between outputs of the classified codecs. Chaotic features calculated from bit stream samples can be accepted as digest values of the bit stream samples. If the outputs can be identified according to the codecs in use, then these digest values should be separable according to the codec type being used.

In Fig. 1, calculation of test and train chaotic feature sets which belong to a speech coder type is described. First all of the samples stored in a speech database are coded by a speech coder and the output (coded samples) is divided into two smaller databases. Each sample in these databases is randomly shifted and pushed into chaotic analysis to produce its chaotic features. As all the samples are processed, two chaotic feature sets (test and train chaotic feature sets) are produced. Since the codec identifier is supposed to work on all possible portions of suspicious bitstream samples, where the beginning and end are unknown, for a fair testing scenario chaotic analysis should not be always calculated from the beginning of the coded sample. Random shifter simply throws away TR bytes from the beginning and shortens the sample.

In the case of codec identification, type of a codec used causes significant separations almost in every feature. Unlike steganalysis where specialized steganalyzers are trained, this phenomenon enables the design of a universal classifier which can potentially classify or identify all types of speech codecs and may give the correct classification by using only FNN as features.

As this work can be described as a proof of concept work, the relationship between the number of chaotic features (N_{FNN}), window sizes (The length of bitstream where FNN values are calculated for) and codec identification performance is not investigated in a detailed fashion. Therefore, all training and tests are performed for $N_{FNN} = 15$ case., On the other side theoretically,

- If N_{FNN} increases, then
  - Identification performance increases,
  - Complexity increases,
  - Larger train sets are required.
- If N_{FNN} decreases, then
  - Identification performance decreases,
  - Complexity decreases,
  - Smaller train sets are sufficient.
- If the window size increases, then
  - Identification performance increases,
  - Complexity increases.
- If the window size decreases, then
  - Identification performance decreases,
  - Complexity decreases.
- If N_{FNN} increases while the size of the training set is kept unchanged
  - Identification performance increases up to an optimum $N_{FNN\ opt}$
  - Identification performance decreases after exceeding the optimum $N_{FNN\ opt}$

Moreover for most practical scenarios, N_{FNN} is certainly an application specific value and applying trial – error procedures may probably be the best way to determine its value.

In Fig. 2, 2D points, which are obtained after principle component analysis (PCA) of 15 dimensional FNN feature vectors calculated from 1024 byte samples (N_{FNN} = 15, N_{LY} = 0), are illustrated. For simplicity, only the outputs of four different types of speech codecs (AMR 4.5K [4], G.726 16K [5], G.726 24K and G.729 32K [6]) are taken into consideration. For each 1024 length speech sample, a FNN feature 15 item vector is calculated. Each item in the vector gives false nearest neighbor fraction for specific dimension and delay. The item's value is always positive and changes between 0 and 1. Then all the calculated FNN vectors are collected together in a collection matrix. The collection matrix is composed of 2048 vectors, which means it carries 512 vectors per codec. After that, 2048×15 collection matrix is projected onto (or converted to) a 2048×2 matrix. The values of the projected matrix alter between the values −1 and 1. Finally all the vectors (or now they are coordinates) belonging to each codec are shown

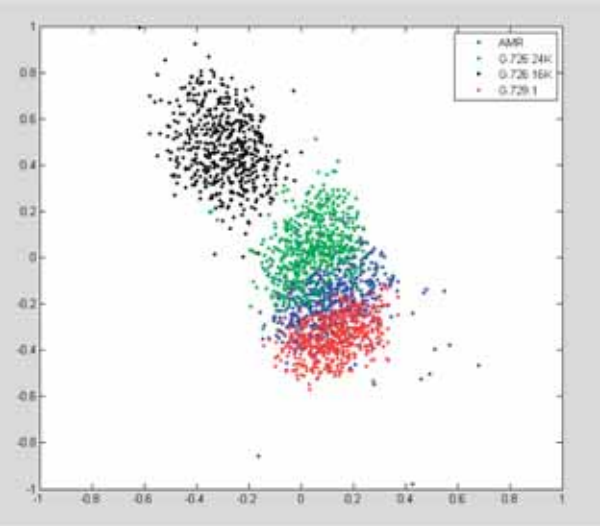

Fig. 2 Separation of projected feature vectors obtained from suspicious bit stream samples

with a different color. As illustrated in the figure, the chaotic features of four codecs are still separable from each other even after a 2-D projection.

6. Speech Codec Identifier Model: Training and Testing

The codec identifier proposed in this paper needs to be trained in order to identify unknown bit stream samples as illustrated in Fig. 3. For training purposes firstly sufficient amount of chaotic type features are calculated for each codec type to form the combined feature database. Then the entire combined features database, which is composed of several chaotic features vectors obtained from the universe of speech coders, may be normalized according to their mean and variance. The parameters of the normalization process are saved and returned as an output. These normalization parameters will be used to normalize the chaotic feature vectors of suspicious bitstream samples. Finally relying on the assumption

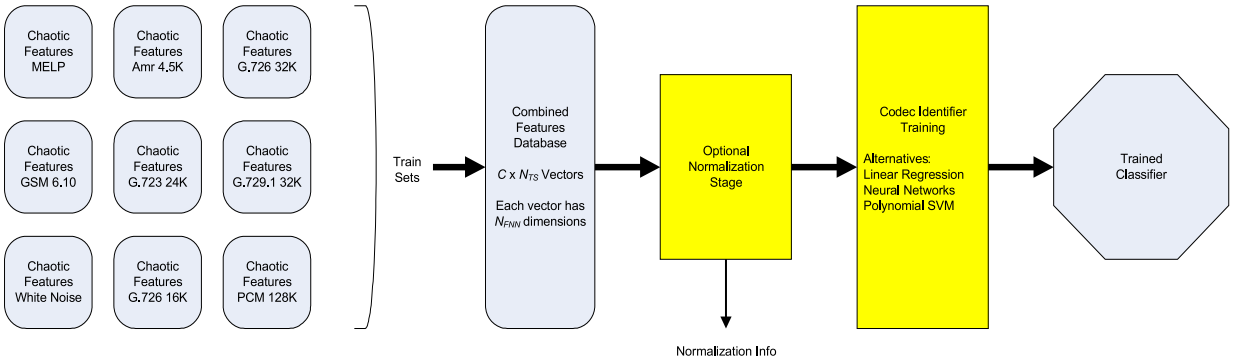

Fig. 3 Training of the codec identifier model

that different codecs would somehow have different distribution of chaotic type features, the normalized feature database is used for classifier training.

After the chaotic analysis stage, N_{FNN} features ($N_{FNN} = 15$) are calculated per sample where the combined training database is actually a two dimensional matrix with N_T rows and N_{FNN} columns. ($N_T = C \times N_{TS}$, N_{TS} is the number of samples per codec, C is the number of codecs to be identified, and N_T is the total number of vectors in the combined training database). For every vector of all codecs, mean (μ) and standard deviation (σ) values are computed and stored as shown in Equation 1 and 2 respectively.

$$\mu = \frac{1}{N_T C}\sum_{i=1}^{N_T C} v_i \tag{1}$$

$$\sigma = \sqrt{\frac{1}{N_T C - 1}\sum_{i=1}^{N_T C}(v_i - \mu)^2} \tag{2}$$

Vector elements are normalized with these stored mean and standard deviation as shown in Equation 3.

$$v_i = \frac{v_i - \mu}{\sigma} \tag{3}$$

After normalization, the combined training database is constructed. This database is used for the classifier training. Training can be defined as determination – calculation of the weights of chaotic features where the codec types can be distinguished in the "optimum" way. The classifier can be designed by using either LR (linear regression) [7], polynomial SVM (Support vector machine) [8] or neural networks [9].

Once the codec identifier has been trained, identification of an unknown bit stream sample (testing) is performed by using its optionally normalized chaotic features as illustrated in Fig. 4. If a normalization process is going to be employed, chaotic feature vector of the unknown bitstream is normalized according to Equation 3. The mean and standard deviation values are obtained from the normalization info which was calculated during training.

7. Performance of the Proposed Codec Identification Model

For all possible codec types, PCM speech files in the database are coded and partitioned into train and test databases. These databases are fed into chaotic analysis to produce test and train chaotic features data sets.

The PCM database is composed of 2560 speech samples, which are chosen from NTIMIT database [10], where each sample's duration may vary. 512 of the speech samples are used in training database, where the rest (remaining 2048 speech samples) form the test database. Note that a speech sample is either in the training or test data set.

The chaotic analysis is applied by executing TISEAN library's [11] false nearest neighbors software for five different embedding dimensions (d_e = 1, 2, 3, 4 and 5). The software produces three chaotic features per embedding dimension and therefore the concatenated feature vector of dimension 15 is calculated for each speech sample.

The codec identification model implemented in this study intends to classify 9 different types of outputs. Namely, AMR 4.5K, G.726 24K, G.726 16K, G.726 32K, G.729.1 32K, GSM 6.10 [12], the Federal and NATO standard MELP (Mixed Excitation Linear Prediction) [13] and white Gaussian noise (WGN). For classification and normalization purposes, five alternative combinations of three types of classifiers are examined as described in Table 1.

The performance of the training process is calculated according to correct identification ratio (CIR) which is defined as the ratio of total number of true positives divided by the total number of bitstream samples tested. (Equation 4)

$$CIR = \frac{\#TP}{\#TP + \#FP} \tag{4}$$

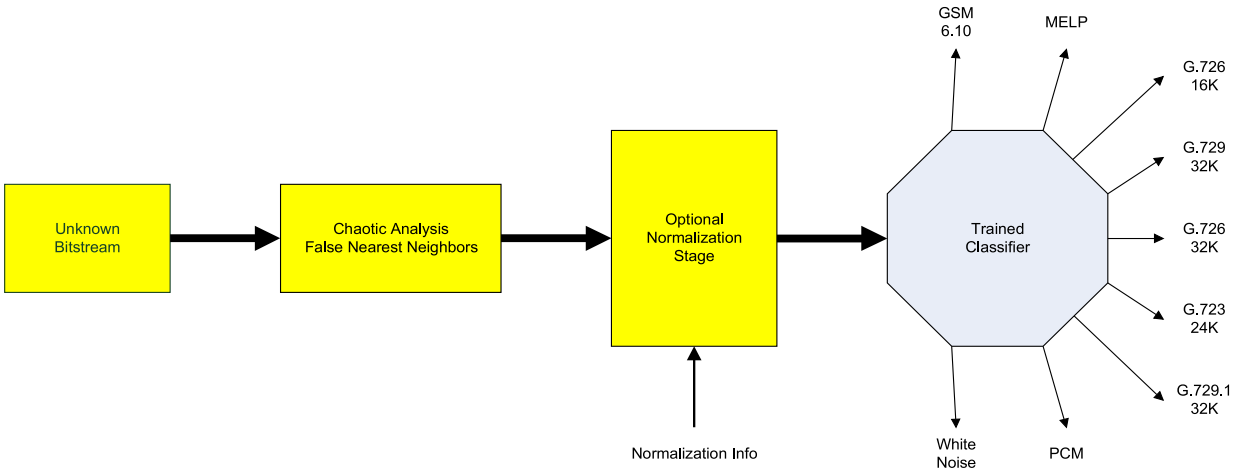

Fig. 4 Codec identification of a suspicious bit stream sample

Classification and normalization combinations used in the proposed codec identifier Table 1.

Classifier Type	Applied Input Normalization
Neural Network	No normalization
Neural Network	Variance
Linear Regression	No normalization
Polynomial SVM	No normalization
Polynomial SVM	Variance

In Table 2, test performance of the trained codec identifiers are presented. With respect to the test results, it is observed that three factors strongly impact the trained identifiers' CIRs:

Classifier Type: Since the chaotic features belonging to different type of codecs could not be properly separated by using planes, linear classifier does not offer high CIR. Non-linear SVM and neural network classifiers yield better and acceptable CIRs.

Window Size: The evaluation length chosen for the speech samples is an important parameter. As the evaluation length, which is the windowed bitstream sample, is increased the confidence of its chaotic features also increases. Unfortunately, because of some practical restrictions like sample collection time, maximum transmitted packet size and calculation complexity of chaotic features, window sizes may not be easily enlarged. Therefore only three of the window size options, 512, 1024 and 1536 are taken into consideration. 512 is chosen since neural network and SVM classifiers start to possess CIRs above 50%. 1536 is an important stopping criterion because it is both divisible by 512 and it is very close to maximum ethernet data size of 1500 and most of the classifiers exhibit better than 90% CIRs. As a matter of fact, in both training and testing cases, chaotic features which are considered as digest – hash values, can be acquired from variable length bitstream samples, training and testing are not needed to be done by taking equal sized bitstream samples as inputs.

Input Normalization: Statistical properties of the codecs influence almost every chaotic feature with varying penetration quantities. Normalization of these features as a preprocessing prior to classification stage improves the convergence rate of the training and reduces the round off errors caused from numerical calculations. Normalization improves performance in all cases.

Performance results of trained codec identifiers Table 2

Classifier Type	Applied Input Normalization	Correct identification ratios for alternative windows sizes in %		
		512	1024	1536
Neural Network	No normalization	53.3	90.5	91.8
Neural Network	Variance	56.2	91.6	95.6
Linear Regression	No normalization	18.5	24.5	34.0
Polynomial SVM	No normalization	75.0	88.5	90.3
Polynomial SVM	Variance	84.4	97.5	98.7

In Table 3, confusion matrix of the most successful codec identifier combination (Polynomial SVM trained and tested with normalized chaotic features calculated from 1536 byte length samples) is presented. AMR is the most accurately identified codec type and MELP 2400 is the least. The highest confusion is reported between MELP 2.4 Kb/s and G.726 24 Kb/s voice coders at around 2%.

Confusion matrix for the proposed speech codec identifier for eight common speech coders and white Gaussian noise bit stream Table 3.

		Claimed Identity								
Actual Codecs		AMR 4.5K	G.726 24K	G.726 32K	G.729.1 32K	G.726 16K	PCM 128K	GSM 6.10 13.2K	MELP 2400	WGN
	AMR 4.5K	1.0000	0	0	0	0	0	0	0	0
	G.726 24K	0	0.9707	0	0	0	0	0.0049	0.0244	0
	G.726 32K	0	0	0.9956	0	0	0.0029	0.0010	0	0.0005
	G.729.1 32K	0.0005	0	0	0.9990	0	0	0	0.0005	0
	G.726 16K	0.0005	0.0005	0	0.0005	0.9961	0	0.0005	0.0010	0.0010
	PCM 128K	0	0	0.0024	0	0	0.9976	0	0	0
	GSM 6.10 13.2K	0.0010	0.0151	0.0005	0	0.0020	0.0015	0.9653	0.0142	0.0005
	MELP 2400	0	0.0205	0.0005	0	0.0059	0	0.0093	0.9639	0
	GWN	0	0	0	0.0020	0.0024	0.0005	0.0005	0	0.9946

Unless inter bit relationships and bit distributions are not same, chaotic features of two different codec's outputs are expected to dissociate. But this dissociation does not always mean that, unsimilar bitstreams' features dissociate more. Training is performed by using every codecs output, and there may be many optimum solutions that depend on the training set contents used.

8. Differences between Steganalysis and Codec Identification

Although the proposed codec identification model is a derivation of the chaotic feature based steganalyzer model there are several differences, as reported below:

Ineffective feature elimination: In steganalysis existence of hidden data may change only few of the features significantly (in other words most of the features may nearly be orthogonal to secret data existence) as a result of this circumstance, for the purpose of reaching maximum decision accuracy impotent features should be discarded. For codec identification however, statistical properties of the codec influence more chaotic features and therefore there is no need for elimination stages.

Feature types: Since existence of secret data does not reveal itself in every chaotic feature in a distinctive manner, the performance of the steganalysis gets better as the number of investigated chaotic features is increased. Consequently chaotic features are the collection of false nearest neighbor results and Lyapunov coefficients in steganalysis. The test results however tell us that usage of false nearest neighbors is sufficient for an acceptable CIR in the codec identification case.

Non linear classifiers: As steganalyzers possess much lower CIRs than codec identifiers, usage of non linear classifiers in steganalysis may not enhance CIR and may force the steganalyzer to memorize the applied training sets. Conversely, with respect to test results presented in this paper codec identifiers with non linear classifiers have significantly higher CIRs.

9. Conclusion

In this study a codec identification model, which identifies the codec type being used, is proposed. The identification model is derived from a steganalyzer model proposed by Kocal et al. [1], where decisions are based on chaotic features calculated from suspicious bit stream samples. The identification model is examined in two steps. Firstly training is described, and then test results and performance limits follow.

In the training, a chaotic feature database is produced from coded speech samples which hosts adequate number of bit stream samples belonging to different speech coding algorithms to be identified. From the assumption of different codecs produce outputs with different statistical properties, calculated chaotic features are shown to be distinct according to the codec being used. From this chaotic feature database, weights which optimize the separation are computed for the classifier. Certainly, the term "weights" has different meanings for different classifier types; in the linear regression case, weights mean a vector, while it means the excitement ratios of all the inputs of all the neurons in a neural network.

During the test, the trained codec identification model is tested using a number of coded speech samples. For each coded speech sample, initially chaotic features are calculated; then these calculated features are fed into the trained codec identifier where decision correct identification ratio of identification model is checked. According to the test results, SVM based codec identifiers accomplish better than 97% correct identification ratio for 1024 and 1536 byte size windows for nine different outputs. Similarly neural network based identifiers achieve above 90% success rate. On the other hand, codec identifiers based on linear regression could only achieve correct identification ratios between 18-34%.

In summary, the proposed codec identification model proves that the type of a speech coder used in a unknown bitstream can be revealed by using chaotic features. Although the proof of concept is carried out on speech, similar approach can be adapted to audio or video which will improve network monitoring applications.

References

[1] KOCAL, O. H., YURUKLU, E., AVCIBAS, I.: Speech Steganalysis Using Chaotic-type Features, *IEEE Transactions on Information Forensics and Security,* 2008, pp. 651–661.

[2] KENNEL, M. B., ABARBANEL, H. D. I.: False Neighbors and False Strands: A Reliable Minimum Embedding Dimension Algorithm, *Physical Review* E (66), 2002, 026209.

[3] MARTINEZ, F., GUILLAMON, A., ALCARAZ, J. C., ALCARAZ, M. C.: *Detection of Chaotic Behaviour in Speech Signals Using the Largest Lyapunov Exponent,* 14th Int'n Conference on Digital Signal Processing (1) (2002) 317–320.

[4] 3GPP, TS 26.090: Adaptive Multi-Rate (AMR) Speech Transcoding, version 4.0.0, 3rd Generation Partnership Project, 3GPP, 2001-03.

[5] ITU-T, G.726: 40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM)

[6] ITU-T, G.729.1: G.729 based Embedded Variable bit-rate coder: An 8-32 kbit/s scalable wideband coder bitstream interoperable with G.729.

[7] ROJAS, R.: *Neural Networks - A Systematic Introduction,* Springer, 1996.

[8] BURGES, C. J. C.: A Tutorial on Support Vector Machines for Pattern Recognition, *Data Mining and Knowledge Discovery* (2), 1998, pp. 121–167.

[9] HAYKIN, S.: *Neural Networks-A Comprehensive Foundation, second ed.,* Prentice- Hall, New York, 1999.

[10] JANKOWSKI, C., KALYANSWAMY, A., BASSON, S., SPITZ, J.: NTIMIT: *A Phonetically Balanced, Continuous Speech, Telephone Bandwidth Speech Database,* ICASSP-90 Intern. Conference on Acoustics, Speech, and Signal Processing (1), 1990, pp. 109–112.
[11] HEGGER, R., KANTZ, H., SCHREIBER, T.: TISEAN: *Nonlinear Time Series Analysis,* http://www.mpipks-dresden.mpg.de/~tisean/ (accessed December 2009).
[12] ETSI, EN 300 961: Digital Cellular Telecommunications System (Phase 2+) (GSM) Full rate speech transcoding, GSM 06.10 version 8.1.1, 1999.
[13] US DoD, MIL-STD-3005, Department of Defense Telecommunications Systems Standard, 1999.