Choosing an Accurate number of Mel Frequency Cepstral Coefficients for Audio Classification Purpose

Lacrimioara GRAMA, Corneliu RUSU
Signal Processing Group, Basis of Electronics Department,
Faculty of Electronics, Telecommunications and Information Technology, Technical University of Cluj-Napoca,
Cluj-Napoca, Romania
{Lacrimioara.Grama, Corneliu.Rusu}@bel.utcluj.ro

***Abstract*—In this paper, we study several audio classification schemes applied on different number of features for multiclass classification with imbalanced datasets. As features, we proposed the liftering Mel frequency cepstral coefficients, while for classification we use probabilistic methods, instance-based learning algorithms, support vector machines, neural networks, L^∞-norm based classifier, fuzzy lattice reasoning classifier, and trees. The final goal is to find the appropriate number of liftering Mel frequency cepstral coefficients to provide the desired accuracy for audio classification purpose. The best results are obtained using 16 features and *k*-Nearest Neighbor as a classifier. In this case, the correct classification rate is 99.79%, the false alarm rate is 0.05%, the miss rate is 0.21%, the precision is 99.80% and the *F*-measure is 99.79%.**

Keywords— audio classification; MFCC; sinusoidal liftering; Bayesian network; SVM; MLP; kNN; CHIRP; FLR; LMT; Random Forests;

I. INTRODUCTION

In the last few years there is an increased interest for detecting acoustic events in audio signals, within the audio community. The motivation is to develop automatic methods for environmental sound recognition. The problem is challenging for two reasons when compared to speech and music sounds: the variability and the diversity of audio events [1]; there is less research for intruder audio-detection systems than for speech and music processing.

In this paper, we propose an audio signal classification system for environmental sounds, which can be used as an intruder detection system. In standard sound classification systems, the classification of a sound is usually composed of two phases. First, feature extraction is using various techniques to characterize the signal to be classified. Then, for these features, a classifier is selected to assign a pattern to a class. The sound classes chosen for evaluation (birds, chainsaws, gunshots, human voices, tractors) belong to audio events from wildlife intruder detection applications.

In [2], Mel frequency cepstral coefficients (MFCCs) with different windows and hop sizes, and Support Vector Machines (SVMs) with optimized parameters are used. The tests using 5-fold cross validation (CV) give 62% precision, 58% recall and 55% *F*-measure value. In [3], for MFCCs and SVMs, the reported accuracies are 64.3% (one-against-the-rest) and 68.2% (one-against-one) for linear kernel. In [4], the overall accuracy reported is 83%. In [5], for block based MFCCs and SVMs, the highest accuracy is 93.4%. In [6] MFCCs, DELTAs and DELTA-DELTAs were used together with SVMs, for audio signals like the ones used in this paper. For 12 features, they reported 98.66% accuracy using linear kernel and 97% accuracy using radial basis kernel. The environmental sound recognition system in [7] uses MPEG-7 audio low level descriptors and MFCCs for audio signals corresponding to work, restaurant, crowded street, quiet street, shopping mall, car with open window, car with closed window, corridor of university campus, office room, desert and park. For only MFCCs, full MPEG-7, selected MPEG-7 using Principal Component Analysis, and MFCCs combined by selected MPEG-7, the system gives accuracies of 85%, 89%, 91% and 93%, respectively. We have also proposed previously different solutions for the wildlife intruder detection [8, 9], based on audio pattern. In [8], the averaged binary spectrogram was used together with artificial neural networks (ANNs). The overall accuracy was 97.4%. The accuracy was increased to 98.5% in [9] by improving the spectral signature.

Through this work, we shall study several classification algorithms to determine the effect of different number of features (liftering MFCCs) towards the classification accuracy. The experimental results will prove that MFCCs can be used together with different classifiers, in the context of source sound detection, to obtain high correct classification rates. To weight the MFCCs for obtaining an equal variance, we shall apply the sinusoidal liftering. For classification phase, we shall use probabilistic methods (Bayesian Networks (BN)), instance-based learning algorithms (KStar, *k*-NN), SVMs, ANNs (Multilayer Perceptron (MLP)), L^∞-norm based classifier (CHIRP), fuzzy lattice reasoning (FLR) classifier, and trees (Logistic Model Trees (LMT), Random Forests (RF)).

The rest of the paper is organized as follows. In Section II some theoretical background regarding liftering MFCCs is

978-1-5090-4011-7/17/$31.00 ©

recalled and classifiers used are illustrated. The proposed audio classification system is overviewed in Section III, while the obtained results are the subject of Section IV. Finally, conclusions are presented in Section V.

II. Theoretical Backround

A. Mel Frequency Cepstral Coefficients

MFCCs give a good discriminative performance; they combine the advantages of cepstrum with a frequency scale based on critical band. To extract the MFCCs one should follow a few steps.

To enhance high-frequency spectrum and to reduce noise, a first order high pass finite impulse response filter (pre-emphasis filter) is applied, with a_{pre} between 0.95 ÷ 0.98 [10]:

$$H_{\text{pre}}(z) = 1 - a_{\text{pre}} z^{-1} \tag{1}$$

The audio signal must be transformed into statistically stationary blocks (frames), with duration between 20 ÷ 40 ms. To avoid losing information at the end of frames, a 25% ÷ 75% overlap can be used. To prevent abrupt changes at the end of frames, a Hamming or a Hanning window is applied to smooth the signal. Each frame should be converted from time domain into frequency domain, thus FFT radix-2 is applied to each frame and the magnitude is evaluated. Because human ears act as filters, concentrate only on certain frequencies, the frequency band should be divided using triangular filterbanks, spaced on the Mel-scale: more filters for low frequencies and less for high frequencies (Fig. 1 – top), spread over the entire frequency range (0 ÷ Nyquist rate). Typically, the number of triangular filters is 20 ÷ 40 [11]. The coefficients magnitude is multiplied by the corresponding filter gain and the results are accumulated [12]. The logarithm of the amplitude spectrum (at the output of Mel filterbank) is then computed. The log compressed filterbank energies are decorrelated using the DCT to produce cepstral coefficients [12]:

$$c_n = \sqrt{\frac{2}{M}} \sum_{i=1}^{M} \ln S_i \cos\left[\frac{\pi n}{M}(i - 0.5)\right], \quad n = \overline{1, N} \tag{2}$$

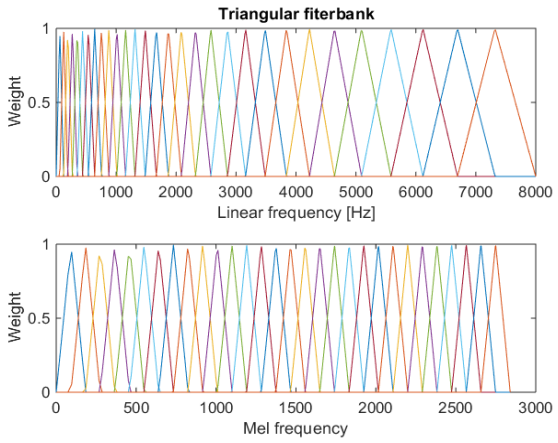

Fig. 1. Filterbank – 30 triangular filters spaced on the Mel-scale (0 – 8 kHz).

where c_n is the n^{th} MFCC, M is the number of filterbank channels, S_i is the magnitude response of the i^{th} filterbank channel, and N is the number of coefficients we want to compute. The 0^{th} MFCC is dropped. Liftering weights the cepstral coefficients to obtain an equal variance. In this way, it improves the recognition performance [13].

Finally, the n^{th} parameters from each frame are added together. To obtain the features vector, they are normalized by the number of frames of the audio signal.

B. Classifiers

For this research, nine classification algorithms were used.

Bayesian Networks (BN) are probabilistic based classifiers, which combine principles from graph and probability theory with computer science and statistics [14]. They are directed acyclic graphs representing a joint probability distribution over a set of random variables. The dual nature of a BN makes learning as a two-stage process: first learn a network structure, then learn the probability tables [15].

Support Vector Machines (SVMs) are supervised learning methods employed for classification or regression, based on the concept of decision planes which define decision boundaries. In the case of classification, SVMs use an iterative algorithm to construct an optimal hyperplane to minimize an error function [16]. For SVMs the LibSVM library was used [17].

Multilayer perceptron (MLP) is a feed-forward ANN that maps input datasets onto appropriate outputs. The inputs are weighted and then summed at each hidden layer node; this weighted sum is used as the input to an activation function. It implements backpropagation for training [16].

For *k-Neareast Neighbor* (*k*-NN) algorithm the RseslibKNN library was employed [18]. Various distance measures are applicable to data. It implements fast neighbor search in large datasets.

KStar is a nearest-neighbor method with a generalized distance function based on transformations [16]; it utilized an entropy-based distance function.

Composite Hypercubes on Iterated Random Projections (CHIRP) is a nonparametric, ensemble classifier based on L^{∞}-norm. It employs an iterative sequence of three stages: projecting, binning, and covering [16]. Its computational effort is sub-linear in number of instances and number of variables and sub-quadratic in number of classes [19].

Fuzzy Lattice Reasoning (FLR) classifier generates a rule-based inference engine from data, based on fuzzy lattice scheme. Learning is carried out both incrementally and fast by computing disjunctions of join-lattice interval conjunctions; a join-lattice interval conjunction corresponds to a hyperbox in R^N [20].

Logistic Model Trees (LMT) use logistic regression models at the leaves. When the logistic regression functions fit at a node, the CV is used to determine the optimal number of iterations to run; it employs the same number throughout the tree instead of CV at each node. The splitting criterion can be based on C4.5's information gain or on the Logit Boost

residuals [16]. LMT makes use of the minimal cost-complexity pruning to produce a compact tree structure.

For *Random Forests* (RF), random vectors are generated to grow an ensemble of trees and those trees are let to vote for the most popular class [16]. To avoid correlation among trees a subset of features is randomly selected to find a best split at each node. When the training set of one tree is built by sampling with replacement, about third of the cases are left out of the bootstrap sample from the original data. This out-of-bag (OOB) data is used to get a running unbiased estimate of the classification error [21]. After each tree is built, all the data are run down the tree, and proximities are computed for each pair of cases. At the end of the run, the proximities are normalized by the number of trees [22].

III. The Audio Classification System

A. Database

The audio signal database considered for the current work is an improved version of the one used in [8, 9]. The sampling rate for all the audio signals in the database is F_s = 16 kHz; all of them are quantized on 16 bits. None of them are studio recordings. We have five classes: birds (654 audio files), chainsaws (356 audio files), gunshots (120 audio files), human voice (207 speech sounds), and tractors (260 audio files).

B. Feature extraction

For feature extraction phase, we use the MFCCs. In (1) a_{pre} is set to 0.97. Each audio signal is divided into 25 ms (400 samples) frames with 15 ms overlap (60%), and a Hamming window is applied. For 25 ms frame length, FFT-512 is employed; 30 triangular filters are used; the frequency range is 0 ÷ 8 kHz (see Fig. 1). We consider a sinusoidal liftering:

$$c'_n = \left(1 + \frac{L}{2}\sin\frac{\pi n}{L}\right) c_n, \qquad n = \overline{1, N} \tag{3}$$

where L is the number of the liftering parameters. Because we want to make a comparison of the influence of the number of MFCCs in the classification accuracy, N was chosen to be 10, 12, 14, 16, 18, 20, 24, and 28, respectively.

The features matrix **F**, for all 1597 audio signals in the database, is as defined as in (4):

$$\mathbf{F}_{1597xN} = \begin{bmatrix} c'_{1,1} & c'_{1,2} & \dots & c'_{1,N} \\ c'_{2,1} & c'_{2,2} & \dots & c'_{2,N} \\ \dots & \dots & \dots & \dots \\ c'_{1597,1} & c'_{1597,2} & \dots & c'_{1597,N} \end{bmatrix} \tag{4}$$

Each row of matrix **F** represents the features vector of one audio signal, and N defines the number of MFCCs. The features extraction step was carried out in MATLAB.

C. Classification

The best number of features to be used together with a given classifier, is determined using a specialized tool for data mining, named WEKA [23]. We have selected several classifiers, the ones present in Section II-C.

For BN, we use a global K2 search algorithm with 10-fold CV as illustrated in [16]. The maximum number of parents are set to 5, thus a Bayes Net Augmented BN is learned. For estimating the conditional probability tables of the BN, a simple estimator, as presented in [24], is used. It estimates probabilities directly from data. A value of 0.5 is set which can be interpreted as the initial count on each value.

In the case of SVMs, the radial basis kernel is chosen. The values employed for the parameters are: the cost C value is 32, γ value is 0.0009, and the tolerance of the termination criterion is 0.0001. These values were determined using a grid search algorithm [25].

To improve performance for the MLP classifier a momentum of 0.2 for updating weights from the previous iteration is set. This smooth the search process by making changes in direction less abrupt [18]. The learning rate is 0.3 and the number of training epochs is 500. The activation function is sigmoid.

The k-NN classifier learns the optimal number of nearest neighbors by optimizing the classification accuracy of the training set. The maximum possible value while learning the optimum is 100-NN, and 5-NN is considered to take part in selection of decision for a classified object. The method of voting is the inverse square distance. For measuring distance between data objects, City and Simple Value Difference is used.

For KStar, the entropy-based blending is employed with the value of 10 for the parameter for global blending.

In the case of the CHIRP classifier, we run it 7 times and score a testing instance based on simple-majority, equally-weighted vote.

For FLR classifier the vigilance parameter is set to 0.75 and the number of rules is optimized.

For LMT we minimize the root mean squared error instead of the misclassification error. The minimum number of instances at which a node, considered for splitting, is 15.

For RF we consider 100 iterations for bagging. The forest of decision trees is build based on different attributes in the nodes. Different trees have access to a different sub collection of the features set, or to a different sub collection of data. The OOB estimate is used.

IV. Experimental Results

For intruder detection applications, besides de correct classification rate (accuracy) (*CCR*), some other metrics are also important, such as the false alarm rate (false positive rate/fall out rate) (*FAR*) or the miss rate (false negative rate) (*MR*). In general, we want a high *CCR*, and low values for *FAR* and *MR*. To compare our results with the ones reported in the

audio signal classification literature, as performance measures we have also used the precision (*P*) and the *F*-measure (*F*1-score) (*F*). Because we faced out with multiclass audio classification with imbalanced datasets, we decided to use the weighted average of the above mentioned metrics. In this sense, we have conducted 72 experiments (8 different number of features x 9 classifiers).

To see the influence of different number of features using the above classifiers, we have evaluated the average value of the aforementioned metrics (*AvCCR*, *AvFAR*, *AvMR*, *AvP*, *AvF*), together with their corresponding standard deviation (*StdDev*), over 10 runs of stratified 10-fold CV (same splits into training/test set for each classifier are used). Each set of 10 CVs folds is averaged, and then, to obtain the result, the averaged values for each run are averaged. Stratified *k*-fold CV is important for imbalanced data sets, especially for classification problems. Stratification reduces the estimate's variance [16].

The results obtained are presented in Table I. On the header row, together with the name of the classifier, the running time is also illustrated; it represents the total time needed for all eight different number of features considered. The classifiers were run on an Intel Core i7, 8GB.

In Table I we have highlighted with yellow the results with *AvCCR* ∈ [98.0%, 98.99%], with green the results for *AvCCR* ∈ [99.0%, 99.74%], and with blue for *AvCCR* ∈ [99.75%, 100.0%]. In all three cases, we have low *AvFAR* (below 0.41%), and *AvMR* (below 1.95%) values. It is also important to have a small *StdDev*. In all highlighted cases the *StdDev* is no larger than 0.33, i.e. for accuracy.

The evolution of algorithms with *AvCCR* over 96%, during all experiments, is illustrated in Fig. 2, based on the *AvCCR*. Taking in account the metrics used, we can conclude that the three best classifiers for our purpose are: *k*-NN, SVMs, and RF. From Table I we can point out that these three classifiers are relatively fast classifiers.

These three classifiers are illustrated separately in Fig. 3. On the *x*-axis, we have the number of features, on the *y*-axis the *AvFAR*, while on the *z*-axis the *AvMR*. The *AvCCR* is illustrated based on the colormap on the right and based on dimension (higher dimension means higher accuracy), while for each classifier a different marker is used. Each quadruple point (features number, *AvFAR*, *AvMR*, *AvCCR*) is also labelled by the name of the classifier.

We can notice that, regardless the number of features, for *k*-NN the *AvCCR* is higher than 99.60, for SVMs the *AvCCR* is higher than 98.68%, and for RF is higher than 98.89%.

In the case of *k*-NN, the best classification is obtained for MFCC-16: *AvCCR* = 99.79%, *AvFAR* = 0.05%, *AvMR* = 0.21%, *AvP* = 99.80% and *AvF* = 99.79%. For SVMs, the best classification is attained for MFCC-28: *AvCCR* = 99.76%, *AvFAR* = 0.06%, *AvMR* = 0.24%, *AvP* = 99.77% and *AvF* = 99.75%. In the case of RF, the best classification is obtained for MFCC-28: *AvCCR* = 99.38%, *AvFAR* = 0.27%, *AvMR* = 0.62%, *AvP* = 99.41% and *AvF* = 99.37%.

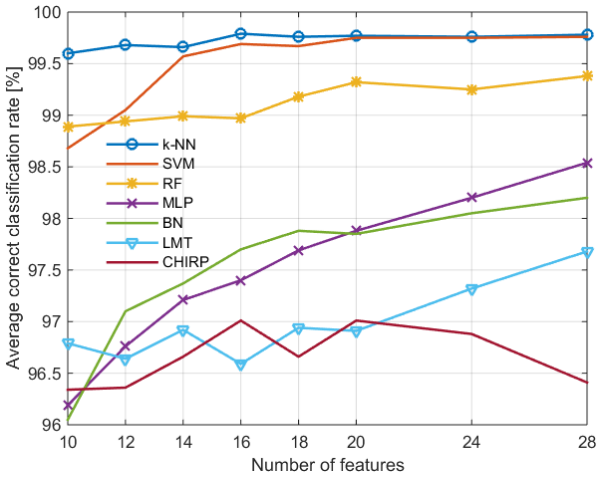

Fig. 2. Average correcct classification rate evolution for all experiments.

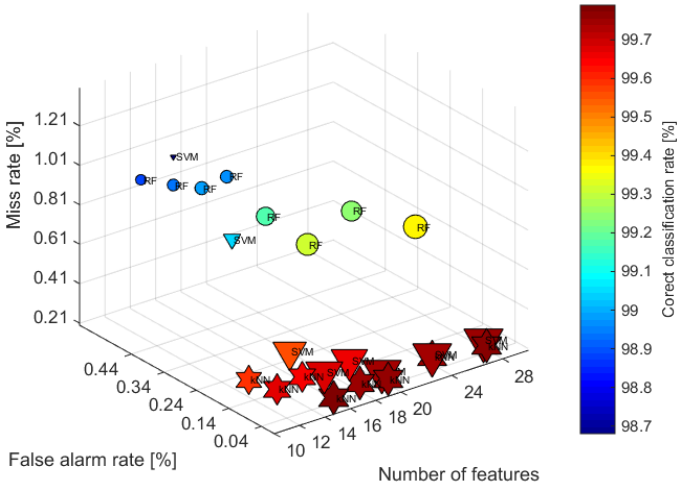

Fig. 3. (Number of features, *AvFAR*, *AvMR*, *AvCCR*) for *k*-NN (hexagram), SVMs (downward-pointing triangle) and RF (circle) classifiers.

For an intruder detection application, the best choice, from the results presented, is *k*-NN and MFCC-16. The results are almost identical if we are using a larger number of features. Similar results are obtained for SVMs and MFCC-28, and almost identical results are obtained for 20 or 24 features.

V. Conclusions

In this paper, we have studied different audio classification algorithms which were applied on different number of features, for multiclass audio classification with imbalanced datasets. The purpose was to find the adequate number of features and the best classifier for an environmental event detection. The proposed audio classification scheme can be used as an intruder detection system, since the sound classes (birds, chainsaws, gunshots, human voices, tractors) belong to audio events from wildlife intruder detection applications.

As features we have used the liftering MFCCs (different values for MFCCs), while for classification we have used BN, *k*-NN, KStar, SVMs, MLP, CHIRP, FLR, LMT, and RF.

TABLE I. AVERAGE CLASSIFICATION METRICS AND THEIR CORESPONDING STANDARD DEVIATION

Features	Classifier	BN 1h56'59''	SVM 7'48''	MLP 41'40''	kNN 13'34''	KStar 1h4'18''	CHIRP 22'18'	FLR 2'40''	LMT 34'15''	RF 14'54''
MFCC-10	AvCCR [%] (StdDev)	96.19 (0.3756)	98.68 (0.1386)	96.05 (0.4359)	99.60 (0.1725)	89.38 (0.5263)	96.34 (0.2800)	89.07 (0.7033)	96.79 (0.4154)	98.89 (0.1974)
	AvFAR [%] (StdDev)	0.97 (0.0012)	0.31 (0.0005)	1.19 (0.0015)	0.08 (0.0004)	4.70 (0.0028)	1.32 (0.0011)	3.03 (0.0031)	0.80 (0.0012)	0.41 (0.0008)
	AvMR [%] (StdDev)	3.81 (0.0038)	1.32 (0.0014)	3.95 (0.0044)	0.40 (0.0017)	10.62 (0.0053)	3.66 (0.0028)	10.93 (0.0070)	3.21 (0.0042)	1.11 (0.0020)
	AvP [%] (StdDev)	96.37 (0.0034)	98.76 (0.0013)	96.27 (0.0041)	99.62 (0.0016)	89.53 (0.0054)	96.49 (0.0028)	90.07 (0.0035)	96.91 (0.0041)	98.94 (0.0018)
	AvF [%] (StdDev)	96.18 (0.0038)	98.65 (0.0014)	95.87 (0.0048)	99.59 (0.0018)	89.10 (0.0056)	96.16 (0.0031)	88.85 (0.0059)	96.69 (0.0041)	98.86 (0.0020)
MFCC-12	AvCCR [%] (StdDev)	96.76 (0.2222)	99.05 (0.1271)	97.10 (0.2716)	99.68 (0.1488)	90.38 (0.6274)	96.36 (0.3114)	89.98 (0.7087)	96.64 (0.2344)	98.94 (0.1851)
	AvFAR [%] (StdDev)	0.84 (0.0007)	0.21 (0.0005)	0.89 (0.0010)	0.07 (0.0004)	4.35 (0.0031)	1.33 (0.0013)	2.77 (0.0025)	0.85 (0.0010)	0.39 (0.0008)
	AvMR [%] (StdDev)	3.24 (0.0022)	0.95 (0.0013)	2.89 (0.0027)	0.32 (0.0015)	9.62 (0.0063)	3.64 (0.0031)	10.02 (0.0071)	3.36 (0.0023)	1.06 (0.0019)
	AvP [%] (StdDev)	96.91 (0.0021)	99.11 (0.0011)	97.25 (0.0027)	99.70 (0.0014)	90.57 (0.0057)	96.50 (0.0030)	90.94 (0.0044)	96.74 (0.0022)	98.99 (0.0018)
	AvF [%] (StdDev)	96.77 (0.0022)	99.03 (0.0013)	96.96 (0.0029)	99.68 (0.0015)	90.14 (0.0063)	96.20 (0.0035)	89.98 (0.0067)	96.58 (0.0023)	98.92 (0.0019)
MFCC-14	AvCCR [%] (StdDev)	97.21 (0.2862)	99.57 (0.1337)	97.37 (0.2720)	99.66 (0.1984)	90.51 (0.6302)	96.66 (0.4794)	90.38 (0.6772)	96.92 (0.5022)	98.99 (0.1873)
	AvFAR [%] (StdDev)	0.74 (0.0010)	0.11 (0.0003)	0.78 (0.0010)	0.07 (0.0004)	4.41 (0.0030)	1.28 (0.0018)	2.78 (0.0026)	0.73 (0.0012)	0.38 (0.0007)
	AvMR [%] (StdDev)	2.79 (0.0029)	0.43 (0.0013)	2.63 (0.0027)	0.34 (0.0020)	9.49 (0.0063)	3.34 (0.0048)	9.62 (0.0068)	3.08 (0.0050)	1.01 (0.0019)
	AvP [%] (StdDev)	97.33 (0.0028)	99.59 (0.0012)	97.51 (0.0025)	99.68 (0.0018)	90.72 (0.0062)	96.78 (0.0046)	91.41 (0.0048)	97.05 (0.0046)	99.04 (0.0017)
	AvF [%] (StdDev)	97.21 (0.0029)	99.56 (0.0013)	97.26 (0.0029)	99.65 (0.0020)	90.28 (0.0065)	96.54 (0.0051)	90.48 (0.0066)	96.92 (0.0051)	98.97 (0.0019)
MFCC-16	AvCCR [%] (StdDev)	97.40 (0.2397)	99.69 (0.1082)	97.70 (0.2067)	99.79 (0.1359)	90.99 (0.8420)	97.01 (0.3126)	91.26 (0.3752)	96.59 (0.3039)	98.97 (0.2858)
	AvFAR [%] (StdDev)	00.66 (0.0008)	0.08 (0.0004)	00.73 (0.0008)	0.05 (0.0005)	4.19 (0.0035)	01.18 (0.0015)	2.83 (0.0030)	0.83 (0.0010)	0.38 (0.0011)
	AvMR [%] (StdDev)	2.61 (0.0024)	0.31 (0.0011)	2.30 (0.0021)	0.21 (0.0014)	9.01 (0.0084)	2.96 (0.0031)	8.74 (0.0038)	3.41 (0.0030)	1.03 (0.0029)
	AvP [%] (StdDev)	97.51 (0.0023)	99.71 (0.0010)	97.79 (0.0020)	99.80 (0.0013)	91.22 (0.0082)	97.14 (0.0030)	92.04 (0.0038)	96.78 (0.0028)	99.02 (0.0026)
	AvF [%] (StdDev)	97.39 (0.0024)	99.69 (0.0011)	97.61 (0.0022)	99.79 (0.0014)	90.80 (0.0086)	96.95 (0.0032)	91.22 (0.0036)	96.62 (0.0030)	98.95 (0.0030)
MFCC-18	AvCCR [%] (StdDev)	97.69 (0.2675)	99.67 (0.1024)	97.88 (0.3105)	99.76 (0.1628)	90.84 (0.7003)	96.66 (0.2809)	91.30 (0.8963)	96.94 (0.3011)	99.18 (0.1940)
	AvFAR [%] (StdDev)	0.59 (0.0014)	0.08 (0.0004)	0.78 (0.0010)	0.05 (0.0005)	4.34 (0.0031)	1.29 (0.0013)	2.91 (0.0042)	0.79 (0.0012)	0.34 (0.0006)
	AvMR [%] (StdDev)	2.31 (0.0027)	0.33 (0.0010)	2.12 (0.0031)	0.24 (0.0016)	9.16 (0.0070)	3.34 (0.0028)	8.70 (0.0090)	0.03.06 (0.0030)	0.82 (0.0019)
	AvP [%] (StdDev)	97.82 (0.0025)	99.68 (0.0009)	97.96 (0.0030)	99.78 (0.0015)	91.13 (0.0070)	96.79 (0.0029)	92.18 (0.0075)	97.13 (0.0028)	99.22 (0.0018)
	AvF [%] (StdDev)	97.69 (0.0027)	99.67 (0.0010)	97.81 (0.0032)	99.76 (0.0017)	90.65 (0.0072)	96.57 (0.0030)	91.30 (0.0089)	96.96 (0.0030)	99.17 (0.0020)
MFCC-20	AvCCR [%] (StdDev)	97.88 (0.1555)	99.75 (0.0886)	97.85 (0.3010)	99.77 (0.1750)	90.81 (0.5679)	97.01 (0.2759)	91.88 (0.5299)	96.91 (0.1866)	99.32 (0.1158)
	AvFAR [%] (StdDev)	0.57 (0.0007)	0.06 (0.0004)	0.74 (0.0008)	0.04 (0.0004)	4.43 (0.0027)	1.28 (0.0011)	2.83 (0.0033)	0.77 (0.0008)	0.29 (0.0004)
	AvMR [%] (StdDev)	2.12 (0.0016)	0.25 (0.0009)	2.15 (0.0030)	0.23 (0.0018)	9.19 (0.0057)	2.99 (0.0028)	8.12 (0.0053)	3.09 (0.0019)	0.68 (0.0012)
	AvP [%] (StdDev)	97.98 (0.0016)	99.76 (0.0008)	97.92 (0.0029)	99.79 (0.0016)	91.15 (0.0054)	97.12 (0.0027)	92.64 (0.0040)	97.11 (0.0020)	99.35 (0.0011)
	AvF [%] (StdDev)	97.89 (0.0015)	99.75 (0.0009)	97.79 (0.0031)	99.77 (0.0018)	90.65 (0.0059)	96.91 (0.0028)	91.86 (0.0054)	96.93 (0.0019)	99.31 (0.0012)
MFCC-24	AvCCR [%] (StdDev)	98.20 (0.2711)	99.75 (0.0886)	98.05 (0.2694)	99.76 (0.1853)	90.49 (0.2957)	96.88 (0.3029)	92.62 (0.5686)	97.32 (0.3397)	99.25 (0.1624)
	AvFAR [%] (StdDev)	0.51 (0.0010)	0.06 (0.0004)	0.62 (0.0008)	0.06 (0.0006)	4.70 (0.0019)	1.35 (0.0015)	2.81 (0.0047)	0.73 (0.0014)	0.31 (0.0008)

Features	Classifier	BN 1h56'59''	SVM 7'48''	MLP 41'40''	kNN 13'34''	KStar 1h4'18''	CHIRP 22'18'	FLR 2'40''	LMT 34'15''	RF 14'54''
	AvMR [%] (StdDev)	1.80 (0.0027)	0.25 (0.0009)	1.95 (0.0027)	0.24 (0.0019)	9.51 (0.0030)	3.12 (0.0030)	7.38 (0.0057)	2.68 (0.0034)	0.75 (0.0016)
	AvP [%] (StdDev)	98.29 (0.0024)	99.76 (0.0008)	98.12 (0.0023)	99.77 (0.0017)	90.91 (0.0031)	96.97 (0.0029)	93.21 (0.0064)	97.48 (0.0033)	99.29 (0.0015)
	AvF [%] (StdDev)	98.20 (0.0026)	99.75 (0.0009)	98.01 (0.0027)	99.75 (0.0019)	90.34 (0.0030)	96.81 (0.0032)	92.59 (0.0063)	97.33 (0.0034)	99.24 (0.0018)
MFCC-28	AvCCR [%] (StdDev)	98.54 (0.3206)	99.76 (0.0806)	98.20 (0.1630)	99.78 (0.1422)	89.74 (0.3275)	96.41 (0.5275)	93.37 (0.3422)	97.68 (0.2818)	99.38 (0.1457)
	AvFAR [%] (StdDev)	0.43 (0.0012)	0.06 (0.0003)	0.52 (0.0006)	0.05 (0.0005)	5.21 (0.0023)	1.57 (0.0028)	2.86 (0.0032)	0.59 (0.0012)	0.27 (0.0005)
	AvMR [%] (StdDev)	1.46 (0.0032)	0.24 (0.0008)	1.80 (0.0016)	0.22 (0.0014)	10.26 (0.0033)	3.59 (0.0053)	6.63 (0.0034)	2.32 (0.0028)	0.62 (0.0015)
	AvP [%] (StdDev)	98.62 (0.0029)	99.77 (0.0008)	98.27 (0.0016)	99.80 (0.0013)	90.26 (0.0029)	96.52 (0.0052)	93.78 (0.0033)	97.86 (0.0029)	99.41 (0.0013)
	AvF [%] (StdDev)	98.54 (0.0032)	99.75 (0.0008)	98.18 (0.0016)	99.78 (0.0014)	89.57 (0.0033)	96.35 (0.0053)	93.28 (0.0038)	97.71 (0.0028)	99.37 (0.0015)

Regardless the number of features, the correct classification rate, for *k*-NN classifier is higher than 99.60, for SVMs classifier is higher than 98.68%, and for RF classifier is higher than 98.89%. The best audio classification scheme is obtained with 16 features and *k*-NN as a classifier. In this case the correct classification rate is 99.79%, the false alarm rate is 0.05%, the miss rate is 0.21%, the precision is 99.80% and the *F*-measure is 99.79%.

ACKNOWLEDGMENT

This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS/CCCDI–UEFISCDI, project number PNIII-P2-2.1-BG-2016-0378, 54BG/2016, within PNCDI III.

REFERENCES

[1] S. E. I Kucukbay, M. Sert, "Audio Event Detection Using Adaptive Feature Extraction Scheme," 7th International Conferences on Advances in Multimedia (MMEDIA), pp. 44-49, 2015.

[2] S. E. Kucukbay, M. Sert, "Audio-based event detection in office live environments using optimized mfcc-svm approach," IEEE International Conference on Semantic Computing (ICSC) , pp. 475–480, 2015.

[3] B. Uzkent, B. D. Barkana1, H. Cevikalp, "Non-Speech Environmental Sound Classification Using SVMs with a New Set of Features," International Journal of Innovative Computing, Information and Control, vol. 5, issue B, pp. 3511-3524, 2012.

[4] G. Kour, N. Mehan, "Music Genre Classification using MFCC, SVM and BPNN," International Journal of Computer Applications (0975 – 8887), vol. 112, no. 6, pp. 12-14, 2015.

[5] V. Ghodasara, S. Waldekar, D. Paul, G. Saha, "Acoustic Scene Classification Using Block Based MFCC Features," Detection and Classification of Acoustic Scenes and Events, 2016.

[6] M. V. Ghiurcau, C. Rusu, R. C. Bilcu, J. Astola, "Audio based solutions for detecting intruders in wild areas," Signal Processing, vol. 92, issue 3, pp. 829-840, 2012.

[7] D. Giannoulis, E. Benetos, D. Stowell, M. D. Plumbley, "IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events - Development Dataset for Event Detection Task, subtask 1 - OL," IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), pp.1-4, 2013.

[8] L.Grama, C. Rusu, "Averaged Binary Sparsogram for Wildlife Intruder Detection," Acta Technica Napocensis – Electronics and Telecommunications, vol. 56, issue 3, pp. 42-47, 2015.

[9] L.Grama, C. Rusu, "Spectrograms, Sparsograms and Spectral Signatures for Wildlife Intruder Detection," 8th Conference on Speech Technology and Human-Computer Dialogue (SpeD), pg. 1-4, Bucharest, Oct. 2015.

[10] F. Bimbot et al, "A Tutorial on Text-Independent Speaker Verification," EURASIP Journal on Advances in Signal Processing, vol. 2004, issue 4, pp. 430-451, 2004.

[11] S. Young et al. The HTK Book (for HTK Version 3.4.1). Engineering Department, Cambridge University, available: http://htk.eng.cam.ac.uk.

[12] D. Asutosh, M. R. Jena, K. K. Barik, "Mel-Frequency Cepstral Coefficient (MFCC) - a Novel Method for Speaker Recognition," Digital Technologies, vol. 1, issue 1, pp. 1-3, 2014.

[13] L. Rabiner, R. W. Schafer. Theory and Applications of Digital Speech Processing. Pearson: international edition, 2011.

[14] T. Koski, J. Noble. Bayesian Networks: An Introduction. Wiley Series in Probability and Statistics, 2009.

[15] J. Han, M. Kamber. Data Mining: Concepts and Techniques. Elsevier: The Morgan Kaufmann Series in Data Management Systems, second ed., 2006.

[16] I. Witten, E. Frank, M. A. Hall, C. J. Pal. Data Mining: Practical Machine Learning Tools and Techniques. Elsevier: The Morgan Kaufmann Series in Data Management Systems, fourth ed., 2016.

[17] C. C. Chang, C. J. Lin. LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011, available: http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[18] A.Wojna, L. Kowalski. RSESLIB Programmer's Guide, http://rseslib.mimuw.edu.pl/rseslib.pdf, 2017.

[19] L. Wilkinson, A. Anand, D. N. Tuan, "CHIRP: A New Classifier Based on Composite Hypercubes on Iterated Random Projections," 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 6-14, 2011.

[20] I. N. Athanasiadis, "The Fuzzy Lattice Reasoning (FLR) Classifier for Mining Environmental Data," Computational Intelligence Based on Lattice Theory, pp. 175–193, 2007.

[21] R. M. Elbasiony, E. A. Sallam, T. E. Eltobely, M. M. Fahmy, "A hybrid network intrusion detection framework based on random forests and weighted k-means," Ain Shams Engineering Journal, Volume 4, Issue 4, December 2013, Pages 753-762, ISSN 2090-4479.

[22] L. Breinman, A. Cutler, "Random Forests - Statistical Methods for Prediction and Understanding," available: https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm.

[23] WEKA – The University of Waikato, "Weka 3: Data Mining Software in Java", available http://www.cs.waikato.ac.nz/ml/weka, version 3.9.0.

[24] R. R. Bouckaert. Bayesian Network Classifiers in Weka for Version 3-5-7, http://www.cs.waikato.ac.nz/~remco/weka.bn.pdf, 2008.

[25] L. Grama, L. Tuns, C. Rusu, "On the optimization of SVM kernel parameters for improving audio classification accuracy," 14th International Conference on Engineering of Modern Electric Systems (EMES), pp. 224-227, Oradea, June 2017.