DETECTING INDOOR SOUND EVENTS

Toma TELEMBICI, Lacrimioara GRAMA
Signal Processing Group, Basis of Electronics Department,
Faculty of Electronics, Telecommunications and Information Technology, Technical University of Cluj-Napoca,
Cluj-Napoca, Romania
tomaatelembici@yahoo.com; Lacrimioara.Grama@bel.utcluj.ro

Abstract: In this paper we present the problem of context awareness for a service robot, based on acoustic analysis. To describe the audio signals, we proposed the liftering Mel frequency cepstral coefficients as features, while for classification the k-Nearest Neighbor is used. The results obtained are illustrated for different number of features, various filtering methods prior classification, different metrics, voting procedures and weighting methods, respectively. The best results are obtained using 37 features, City and Simple Value Difference metric, Inverse Distance voting, Accuracy Based weighting method, and *k*=3. The correct classification rate is improved from 98.25% to 99.21%, by applying resampling to data before classification.

Keywords: *MFCC; kNN; service robot; audio classification.*

I. INTRODUCTION

In this paper, we propose an improvement of the audio signal classification system for environmental sound events presented in [1]. The improvement in the average correct classification rate is due to resampling data prior classification. This system can be used for a service robot to achieve a good understanding of the context. Especially for elderly who live alone, a service robot is a need. The robot should be aware of the environment inside the house.

Similar work was done in [2-4]. In [2] and [3] the NAR audio dataset [5] is used. In [2] for 40 log-spaced bands in the case of Mel frequency Cepstral Coefficients (MFCC) (they do not report the number of features used) and Support Vector Machines (SVM) they have obtained an accuracy of 91.5% for 10-fold cross validation. The best accuracy for SVM was 97% obtained in their case using as features the interpolated MFCC and Time and Time-Frequency Features: Energy, Zero Crossing Rate (ZCR), Spectral Decrease, Spectral Flatness, Spectral Slope.

In [3], for 10-fold cross validation and using as features only the MFCC, an accuracy of 95.82% is obtained (for 39 features), using SVM as a classifier, and 93.23% using *k*NN (for 27 features). They have obtained an improvement in accuracies by splitting the test data into training (80%) and test (20%) data: 97.62% in the case of SVM for 37 and 39 features, and 96.43% in the case of *k*NN for 17 features.

In [4] the indoor activities are monitored using audio signal, with a performance of 94.9%, using SVM with radial basis kernel, for 123 features: ZCR, the first 12 MFCC, the frame energy, 20 linear prediction coefficients, the harmonics to noise ratio, etc.

Through this work, we shall study several classification algorithms based on k-Nearest Neighbor (*k*NN) to determine the effect of different number of features on the classification accuracy. We shall obtain the adequate metric, weighting method, voting procedure, and the optimal value of *k*, such that the overall correct classification rate to be as high as possible. The experimental results will prove that MFCC together with *k*NN can be used in the context of environmental sound event detection, to obtain high correct classification rates. We shall also prove that the average correct classification rates can be improved by resampling data before classification.

The rest of the paper is organized as follows. In Section II the audio database is presented and the MFCC and the *k*NN are recalled. The experimental results are the subject of Section III. Finally, conclusions are dragged in Section IV.

II. AUDIO CLASSIFICATION SCHEME OVERVIEW

In any audio classification scheme, we should follow two main steps: feature extraction and classification. For the feature extraction phase, we have used the MFCC features (more exactly the liftering MFCC), while for classification we have used *k*NN and stratified 10-fold cross validation.

A. Audio Database

The audio database was recorded with the aid of the TIAGo service robot [6]. The data stream was recorded to contain only a certain class of sound events. It was subsequently split into signals containing only one acoustic event. The sampling rate for all the audio signals in the database is *Fs*=48 kHz; all of them are quantized on 16 bits. None of them are studio recordings.

There are 21 classes, each of which with 30 audio files, that correspond to 3 different scenarios. The “kitchen” scenario contains 8 classes (chair, tap water, drop water, shower water, porcelain dish, cutlery, plastic bag rush, cardboard drop), the “room” scenario contains 8 classes (page turn, Velcro, zip open, zip close, door knock, door open, door close) and the “appliances” scenario contains 5 classes (washing machine, microwave open, microwave close, microwave alarm, toaster alarm).

Manuscript received May 25, 2018; revised June 24, 2018

B. Feature extraction – Mel Frequency Cepstral Coefficients

MFCC features are the most commonly used in speaker recognition. From the audio signal to MFCC there are some steps in order to achieve the coefficients, as illustrated in Fig.1.

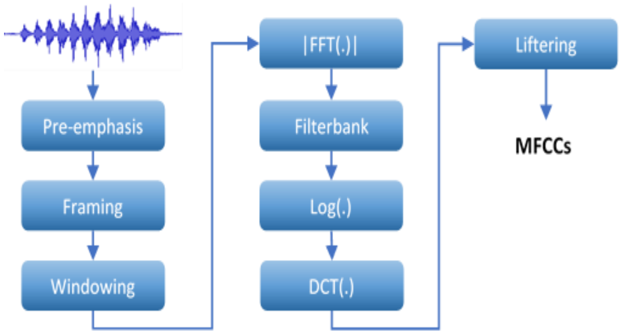


Figure 1. Features extraction stage.

The audio signal is first pre-emphasized using an FIR filter with the pre-emphasize parameter 0.97. After that each audio signal is divided into 25 ms frames with a 60% overlap, and a Hamming window is applied. After that a 512-point Fast Fourier Transform (FFT) was used; 40 triangular filterbanks, spaced on the Mel scale are used; the frequency range is from 0 to 24 kHz. The magnitudes obtained are multiplied by the corresponding filter gain and the results are accumulated. The logarithm of amplitudes at the output of Mel filterbank is evaluated, to compress dynamic range of values. After that, the Discrete Cosine Transform (DCT) is applied on the log of the Mel spectrum, to convert it back to time. For an equal variance, the cepstral coefficients can be weighted, using a sinusoidal liftering:

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

where L is the number of the liftering parameters. *N* was chosen to be 10, 12, 14, …, 38. For each audio signal, the energy was also taken into account, thus for every signal we have extracted 11, 13, 15, …, 39 features. The feature extraction phase was implemented in MATLAB.

C. Classification – K-Nearest Neighbor

For *k*NN algorithm the RseslibKNN library was employed [7]. Various distance measures are applicable to data. It implements fast neighbour search in large datasets. *k* in *k*NN is the number of instances that we take into account for determination of affinity with classes. Detailed description of *k* coefficient optimization algorithm and voting of the nearest neighbors, the analysis of metrics and attribute weighting methods can be found in [8].

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. The methods of voting are [9]:

- Inverse Square Distance (the votes of nearest neighbors are inversely proportional to square of their distances from classified object);
- Inverse Distance (the votes of nearest neighbors are inversely proportional to their distances from classified object);
- and Equal Distance (the votes of all nearest neighbors are equally important).

For measuring distance between data objects next methods were used [10],

- City and Hamming (combination of Manhattan metric for numeric attributes and Hamming metric for symbolic attributes);
- City and Simple Value Difference (combination of Manhattan metric for numeric attributes with Simple Value Difference metric for symbolic attributes; Simple Value Difference is a kind of difference between decision distributions of a pair of attribute values in training set);
- Interpolated Value Difference (combination of Simple Value Difference for symbolic attributes with its version for numeric attributes; the numeric version of this metric is based on dividing the range of values into intervals, counting decision distributions in the intervals from the training set and approximating decision distribution for any numeric value using linear interpolation between the two intervals nearest to a given value);
- and Density Based Value Difference (combination of Simple Value Difference for nominal attributes with its adaptation to numerical attributes that takes into account distribution of numerical value density; computations of decision distribution for every numeric value is based on some neighborhood of this value; the neighborhood is established by the number of nearest values occurring on a given attribute in the training set; decision distribution for a given value is calculated from a subset of training objects whose values on a given attribute fall into calculated neighbourhood)

The weighting methods used are [11]:

- Distance Based (iterative correction of weights to optimize distances to correctly classified training objects);
- Accuracy Based (iterative correction of weights to optimize training objects classification);
- and Perceptron (optimizing weights by the method of perceptron training).

As *k*NN is very popular in voice recognition, it comes as a help for the service robot. In order not to use different classifiers, we have adapted *k*NN to work properly not only with voice but also with indoor surrounding sounds.

III. EXPERIMENTAL RESULTS

As we have already mentioned, for classification we have used the *k*NN method, implemented using Rseslib [12] from WEKA [13]. For all considered cases we run 10 times stratified 10-fold cross validation. A single 10-fold cross-validation might not be enough to get a reliable error estimate. Different 10-fold cross-validation experiments with the same learning method and dataset often produce different results, because of the effect of random variation in choosing the folds themselves. Stratification reduces the variation, but it certainly does not eliminate it entirely. When looking for an accurate error estimate, it is standard procedure to repeat the cross-validation process 10 times, and average the results. Obtaining a good measure of performance is a computation-intensive undertaking [11].

After the features were extracted for each signal, we have tried different filtering methods of extracted features before classification, to see the influence in the average correct classification accuracy (Fig. 2) [1]. For each type of filtering all considered number of features are used (from 11 to 39 with a step of 2).

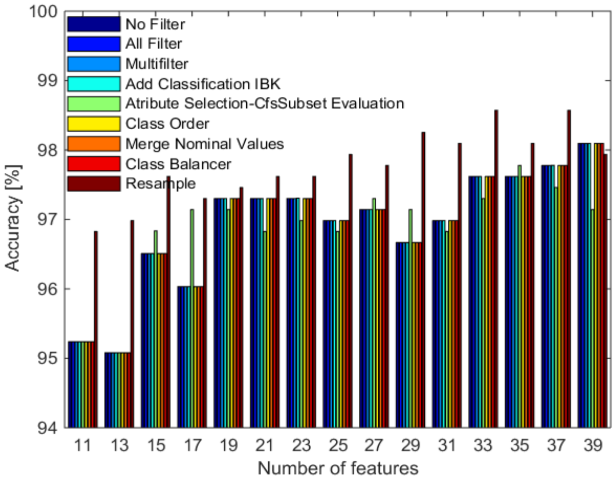


Figure 2. Influence of filtering in the overall classification accuracy (before classification phase) [1].

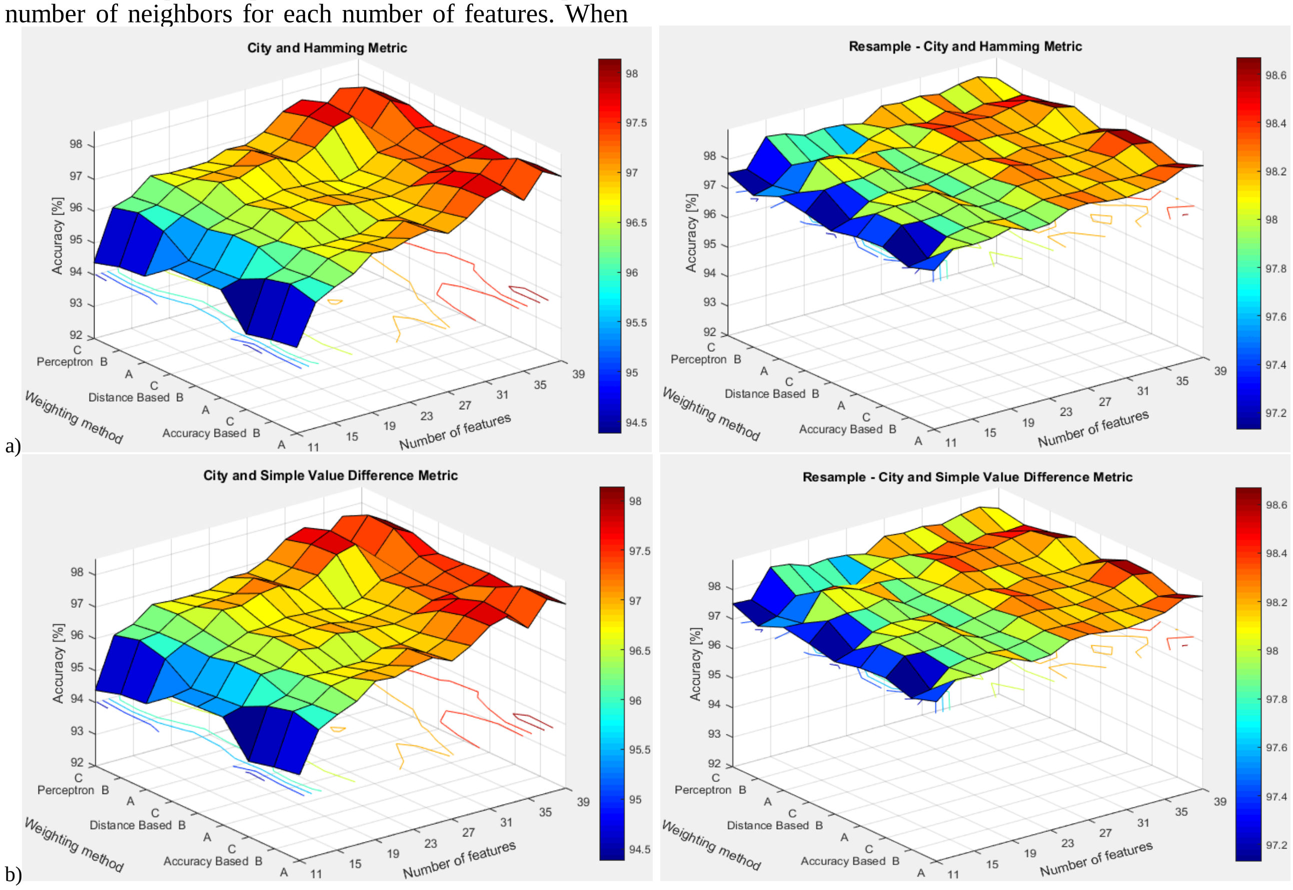
From Fig. 2 we can see that resampling the features prior to classification, the accuracy is higher, for almost all the considered number of features, from 11 to 37, except for 39 features. The average correct classification rate using resampling is 98.57%, for 33 and 37 features, respectively.

Another important phase was to establish the optimal number of neighbors for each number of features. When there is no filter involved prior to classification, in our previous work [1], for our data, *k* reaches seven times 1, two times 5 and six times 3. When resampling is involved, *k* reaches five times 6, ones 5, two times 4, ones 3, ones 2, and six times 1.

In Fig. 3 the average classification accuracy is illustrated for all considered number of features using different metrics (City and Hamming, City and Simple Value Difference, Density Based Value Difference, Interpolated Value Difference), different weighting methods (Distance Based, Accuracy Based, Perceptron), and different voting procedures (Inverse Square Distance, Inverse Distance, Equal Distance), respectively. On the graphs, A stands for inverse square distance, B for inverse distance and C for equal distance.

Because we have observed an increasing of the correct classification rate by resampling data (Fig. 2), in experiments form Fig. 3 we have analysed the differences between results without filters and the ones obtained with resample filters. The results obtained without filtering data prior classification are detailed in [1].

Having a comparison between the two City and Hamming metric experiments (Fig. 3 a)), we can notice that with resample we obtain a low variation between the values and a much higher top value and average value. In the case of no filter the highest average correct classification rate 98.14% is obtained for 37 MFCC features, Accuracy Based and Perceptron weighting method and Inverse Distance voting procedure. In the case of resampling the highest average correct classification rate 98.67% is obtained for 37 MFCC features, Accuracy Based and Perceptron weighting method and Inverse Distance voting procedure.



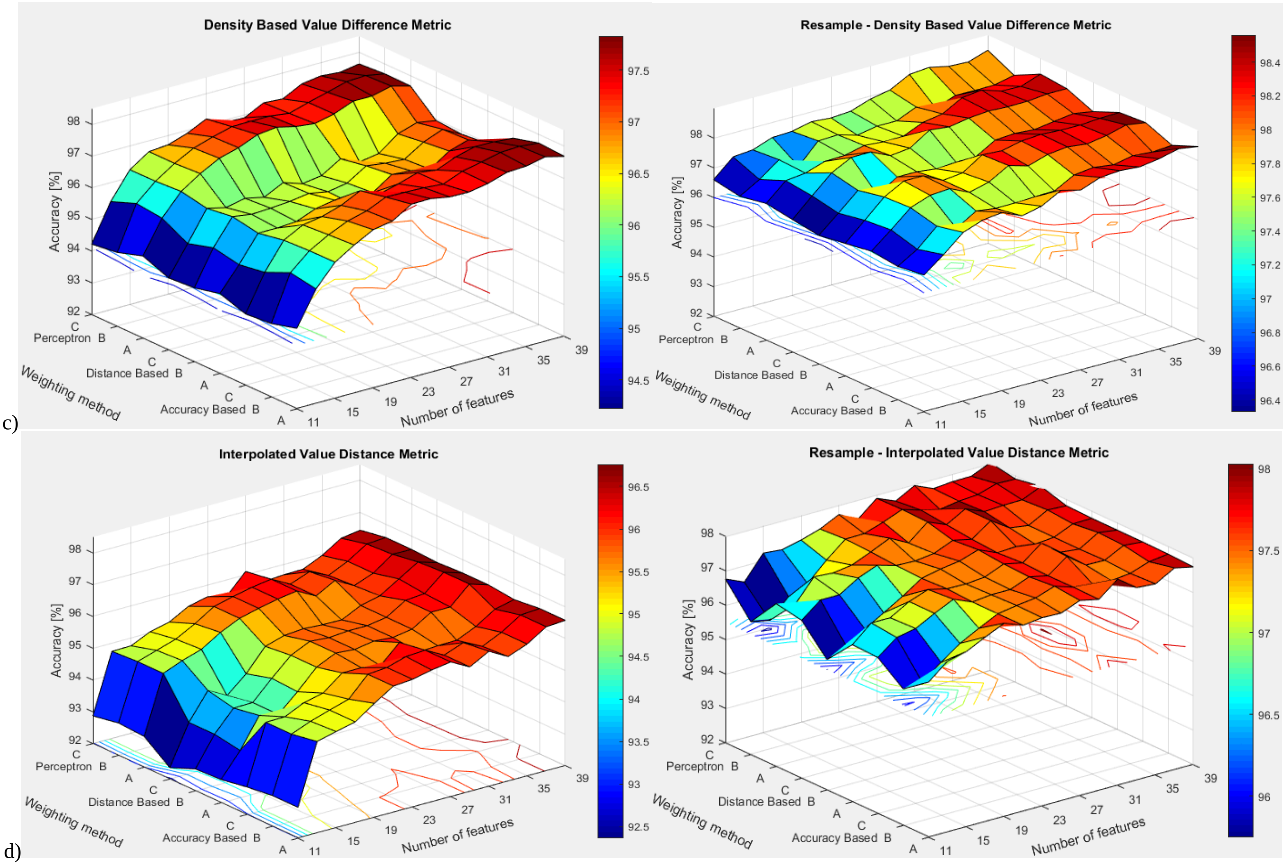


Figure 3. Overall classification accuracy based on number of features, voting procedure and weighting method for: a) City and Hamming, b) City and Simple Difference, c) Density Based Value Difference, d) Interpolated Value Difference metrics: no filter (left), resample (right).

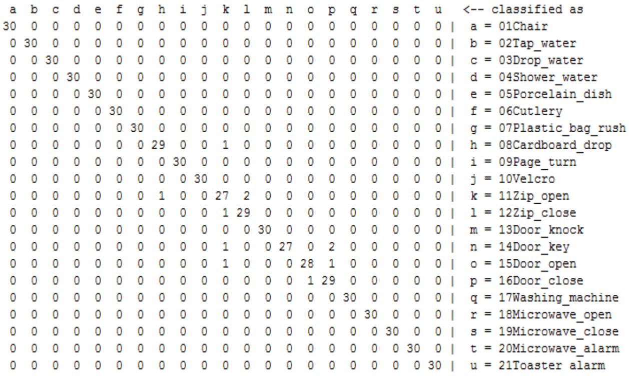
For City and Simple Difference metric (Fig. 3 b)), in the case of no filter the highest average correct classification rate 98.14% is obtained for 37 MFCC features, Accuracy Based and Perceptron weighting method and Inverse Distance voting procedure. In the case of resampling the highest average correct classification rate 98.67% is obtained for 37 MFCC features, Accuracy Based and Perceptron weighting method and Inverse Distance voting procedure.

For Density Based Value Difference metric (Fig. 3 c)), in the case of no filter the highest average correct classification rate 97.83% is obtained for 37 MFCC features, Accuracy Based and Perceptron weighting method and Inverse Square Distance voting procedure. In the case of resampling the highest average correct classification rate 98.56% is obtained for 37 MFCC features, Accuracy Based and Perceptron weighting method and Inverse Square Distance voting procedure.

For Interpolated Value Difference metric (Fig. 3 d)), in the case of no filter the highest average correct classification rate 96.76% is obtained for 37 MFCC features, Distance Based weighting method and Inverse Square Distance and Inverse Distance voting procedure. In the case of resampling the highest average correct classification rate 98.03% is obtained for 39 MFCC features, Accuracy Based and Perceptron weighting method and Equal voting procedure.

When no filter is applied to data prior classification, for a low number of features, i.e. 11, 13 or 15, is better to use the Distance Based voting no matter the metric or weighting method, for 19, 21, 23, 25, 27, 29 or 31, best results are obtained using the Inverse Square Distance voting procedure together with Accuracy Based or Perceptron weighting method no matter the metric used. For a higher number of features, i.e. 33, 35, 37 or 39, best results are obtained for Inverse Distance voting procedure together with Accuracy Based or Perceptron weighting method no matter the metric used.

The confusion matrices are illustrated for the best case (one out of ten). The best case when no filter is applied to data (Fig. 4) is obtained for 37 features, City and Simple Value Difference metric, Inverse Distance voting (Accuracy Based/Perceptron weighting method), optimal *k* equal to 3.

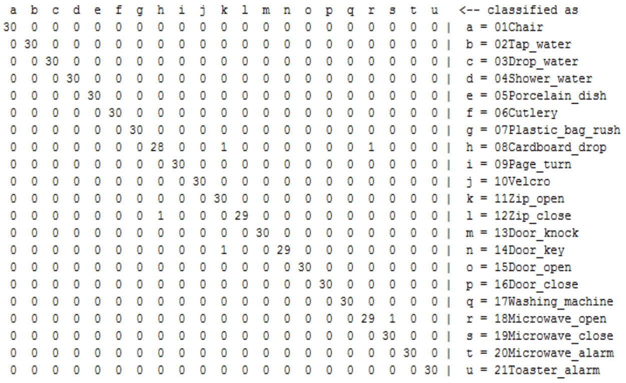


```
  a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p  q  r  s  t  u   <-- classified as
 30  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  a = 01Chair
  0 30  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  b = 02Tap_water
  0  0 30  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  c = 03Drop_water
  0  0  0 30  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  d = 04Shower_water
  0  0  0  0 30  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  e = 05Porcelain_dish
  0  0  0  0  0 30  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  f = 06Cutlery
  0  0  0  0  0  0 30  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  g = 07Plastic_bag_rush
  0  0  0  0  0  0  0 29  0  0  1  0  0  0  0  0  0  0  0  0  0 |  h = 08Cardboard_drop
  0  0  0  0  0  0  0  0 30  0  0  0  0  0  0  0  0  0  0  0  0 |  i = 09Page_turn
  0  0  0  0  0  0  0  0  0 30  0  0  0  0  0  0  0  0  0  0  0 |  j = 10Velcro
  0  0  0  0  0  0  0  1  0  0 27  2  0  0  0  0  0  0  0  0  0 |  k = 11Zip_open
  0  0  0  0  0  0  0  0  0  0  1 29  0  0  0  0  0  0  0  0  0 |  l = 12Zip_close
  0  0  0  0  0  0  0  0  0  0  0  0 30  0  0  0  0  0  0  0  0 |  m = 13Door_knock
  0  0  0  0  0  0  0  0  0  0  1  0  0 27  0  2  0  0  0  0  0 |  n = 14Door_key
  0  0  0  0  0  0  0  0  0  0  1  0  0  0 28  1  0  0  0  0  0 |  o = 15Door_open
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1 29  0  0  0  0  0 |  p = 16Door_close
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 30  0  0  0  0 |  q = 17Washing_machine
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 30  0  0  0 |  r = 18Microwave_open
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 30  0  0 |  s = 19Microwave_close
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 30  0 |  t = 20Microwave_alarm
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 30 |  u = 21Toaster alarm
```

Figure 4. Best case confusion matrix (no filter) [1].

The percent of correctly classified instances is 98.25%. The lowest accuracy is obtained for zip open and door key classes: only 27 out of 30 audio signals are correctly match to the proper class. In the case of zip open, two signals are classified as zip close and one is classified as cardboard drop. The time taken to build model was 5.84 seconds (one of WEKA's output parameters is the time taken to build model).

To see the influence of applying resampling to data before classification, in the average correct classification accuracy, we have performed simulations for the same number of coefficients (37 MFCC), same metric, voting procedure and weighting method (City and Simple Value Difference metric, Inverse Distance voting and Accuracy Based weighting method). The corresponding confusion matrix is illustrated in Fig. 6. We have obtained optimal *k* equal to 2, the percent of correctly classified instances is 99.21%. The lowest accuracy is obtained for cardboard drop: 28 out of 30 audio signals are correctly match to the proper class. In the case of zip close, door key, microwave open, one signal is misclassified. The time taken to build model was 2.76 seconds.



```
  a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p  q  r  s  t  u   <-- classified as
 30  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  a = 01Chair
  0 30  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  b = 02Tap_water
  0  0 30  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  c = 03Drop_water
  0  0  0 30  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  d = 04Shower_water
  0  0  0  0 30  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  e = 05Porcelain_dish
  0  0  0  0  0 30  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  f = 06Cutlery
  0  0  0  0  0  0 30  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  g = 07Plastic_bag_rush
  0  0  0  0  0  0  0 28  0  0  1  0  0  0  0  0  0  1  0  0  0 |  h = 08Cardboard_drop
  0  0  0  0  0  0  0  0 30  0  0  0  0  0  0  0  0  0  0  0  0 |  i = 09Page_turn
  0  0  0  0  0  0  0  0  0 30  0  0  0  0  0  0  0  0  0  0  0 |  j = 10Velcro
  0  0  0  0  0  0  0  0  0  0 30  0  0  0  0  0  0  0  0  0  0 |  k = 11Zip_open
  0  0  0  0  0  0  0  1  0  0  0 29  0  0  0  0  0  0  0  0  0 |  l = 12Zip_close
  0  0  0  0  0  0  0  0  0  0  0  0 30  0  0  0  0  0  0  0  0 |  m = 13Door_knock
  0  0  0  0  0  0  0  0  0  0  1  0  0 29  0  0  0  0  0  0  0 |  n = 14Door_key
  0  0  0  0  0  0  0  0  0  0  0  0  0  0 30  0  0  0  0  0  0 |  o = 15Door_open
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 30  0  0  0  0  0 |  p = 16Door_close
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 30  0  0  0  0 |  q = 17Washing_machine
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 29  1  0  0 |  r = 18Microwave_open
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 30  0  0 |  s = 19Microwave_close
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 30  0 |  t = 20Microwave_alarm
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 30 |  u = 21Toaster_alarm
```

Figure 5. Best case confusion matrix (resample).

IV. CONCLUSION

In this paper we have studied different audio classification schemes using as features different number of MFCC coefficients and *k*NN as a classifier. The purpose was to obtain the adequate number of features, metric, weighting method, voting procedure, and the optimal value of *k*, such that the overall correct classification rate to be as high as possible. We have obtained an accuracy of 98.25%, for City and Simple Value difference metric, Inverse Distance voting, Accuracy Based weighting method, optimal *k* equal to 3, for 37 features, when data is not filter prior classification.

The overall correct classification rate can be improved by resampling data prior classification. For the same number of features (37), same metric, weighting method and voting procedure, by applying resampling, the accuracy is increased to 99.21%.

The database used was one captured by the TIAGo service robot, which consists in 21 classes of audio files that correspond to 3 different scenarios. The proposed audio signal classification system can be used for a service robot to achieve a good understanding of the context. This is a need especially for elderly who lives alone. The robot should be aware of the environment inside the house.

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] T. Telembici, L. Grama, "A Way to Detect Indoor Sound Events," in Novice Insights in Electronics and Telecommunications. SSET 2018, May 2018, pp. 49-50.

[2] M. Janvier, X. Alameda-Pineda, L. Girin, and R. Horaud, "Sound representation and classification benchmark for domestic robots," in IEEE International Conference on Robotics and Automation (ICRA), Hong-Kong, China, pp. 6285–6292, May 2014.

[3] C. Rusu, L. Grama, "Recent Developments in Acoustical Signal Classification for Monitoring," in 5th International Symposium on Electrical and Electronics Engineering (ISEEE), Galati, Romania, pp. 1-10, Oct. 20-22, 2017.

[4] P. Naronglerdrit, I. Mporas, "Recognition of Indoors Activity Sounds for Robot-Based Home Monitoring in Assisted Living Environments," Interactive Collaborative Robotics, Second International Conference, ICR 2017, Hatfield, UK, pp. 153-161, September 12-16, 2017.

[5] NAR dataset."Available: https://team.inria.fr/perception/nard/

[6] PAL Robotics, TIAGo Handbook, version 1.4.2, Barcelona 2016.

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

[8] M. Moshkov, M. Piliszczuk, B. Zielosko, "On Construction of Partial Association Rules," Rough Sets and Knowledge Technology, Springer Berlin Heidelberg, pp. 176-183, 2009.

[9] D. T. Larose, C. D. Larore, Data Mining and Predictive Analytics, Wiley: John Wiley & Sons, Inc., Hoboken, New Jersey, second ed., 2015.

[10] W. Pedrycz, A. Skowron, V. Kreinovich, Handbook of Granular Computing, Wiley: John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, England, 2008.

[11] 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.

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

[13] 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.