Microphone Array Based Surveillance Audio Classification

Dimitri L. O. Silva, Tito Spadini and Ricardo Suyama

***Abstract*— The work assessed seven classifiers and two beamforming algorithms for detecting surveillance sound events. The tests included the use of AWGN with -10 dB to 30 dB SNR and Data Augmentation (DA). The results showed that the combination of *Support Vector Machine* (SVM) and Delay-and-Sum (DaS) scored the best accuracy (up to 86.0%), but had high computational cost (≈ 79 ms), mainly due to DaS and DA. The use of *Stochastic Gradient Descent* (SGD) also seems to be a good alternative since it has achieved good accuracy either (up to 85.3%), but with quicker processing time (≈ 25 ms).**

***Keywords*— Audio classification; Microphone array; Support Vector Machine; Delay-and-Sum; Stochastic Gradient Descent.**

I. Introduction

Several public security systems depend directly on human action in numerous stages of its operation. The monitoring of public areas, for instance, is usually done with the use of cameras spread over the busiest places in large urban centers. In general, these systems depend on an operator to pay attention to the images so that the agencies responsible for security can be activated when events such as thefts, vandalism, and traffic accidents are observed. Considering the amount of information to which the operator is exposed, there is a high probability that surveillance failures will occur, even if the patrol center has a large team [1]. Although the operators are attentive at all times, this type of monitoring has some disadvantages: the images are limited to the direction in which the camera points and have low visibility at dusk and in cases of rain or bright light. Besides, events such as gunshots, alarms, distress calls, among others, are much more noticeable in the auditory field than in the visual [2], [3].

In this sense, the monitoring of risk areas could be done through the use of audio processing techniques, reducing the need for human participation in the surveillance process, and making public security systems more efficient [4]. To support this argument, it is worth recalling two very favorable characteristics concerning these signals: initially, the sound consumes less bandwidth in the transmission of information, reducing the need for high transmission rates, as in the case of high definition images; in addition, sound processing techniques require, in general, less computational power than techniques for video processing and analysis, which would enable the implementation of simpler and therefore less costly embedded systems [3], [5].

Dimitri L. O. Silva, Tito Spadini and Ricardo Suyama. Laboratory of Signals and Systems, UFABC, Santo André - SP. Emails: dimitri.leandro@aluno.ufabc.edu.br, {tito.caco, ricardo.suyama}@ufabc.edu.br.
This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de NívelSuperior - Brasil (CAPES) - Finance Code 001 - and the National Councilfor Scientific and Technological Development – CNPq

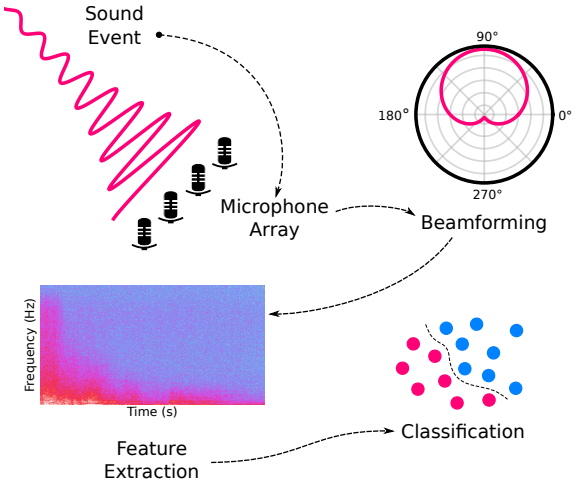

Fig. 1. Illustration of the steps of the proposed system: from audio capture to classification.

The present work is a sequence of what is presented in [6], and aims to evaluate the effectiveness of audio processing techniques for detecting events that are harmful to public safety, based on two important points: the use of machine learning techniques for automatic recognition of sound events; and signal processing in microphone arrays, to improve the signal-to-noise ratio (SNR) and, therefore, obtain better accuracy in the classification stage. Figure 1 summarizes the proposed system.

Following, Section II presents a bibliographic review on the subjects covered in this article, and Section III reveals the main considerations about the algorithms employed. Section IV explains the simulation environment and procedures adopted, while Section V discuss the outcomes. Finally, Section VI concludes the work.

II. Literature Review

In recent years, researchers have shown that the most effective tools for the classification of sound events include the application of deep, convolutional, and recurrent neural networks (DNN, CNN, and RNN) [7], [8], [3], [9], [4]. However, for the current work, the concern with the processing time of the algorithms is fundamental, since, among the future goals, the aim is to create a low-cost system capable of running in real-time. Therefore, the solutions used here address classic machine learning algorithms, seeking to find a balance between accuracy and computational cost. In [10], the authors also worked with audio classification for security systems and used two of the seven algorithms that will be covered in

this article, K Nearest Neighbors (KNN) and Support Vector Machines (SVM). Similar to the previous one, [3] revealed that Random Forests can achieve high accuracy with MIVIA Dataset [11], specifically for surveillance applications, even with low SNR.

Recent work advocates the use of DNN and CNN can perceive patterns in auditors without using many features [9], [7]. Both were able to acquire good results using only Mel Frequency Cepstral Coefficients (MFCC). [8], [3], [4] states that temporal and frequency features, when separately used, can't achieve satisfactory performance, especially in noisy environments, but the combination of those can significantly improve the classification task. The authors [8] and [4] claim that Spectral Chroma, Spectral Contrast, and Tonnetz were primarily related to musical classification, but examine their performance in classification of other sorts of audio signals, attesting they exert a fundamental role in that duty. In [8], [3], and [4], the authors confirm the excellent performance of MFCC and its first and second-order derivatives, Delta and Delta Delta.

In [12] and [9], the authors use beamforming to improve the classification of audios. The first one operates in the field of monitoring systems and, as in the current work, simulates an array of microphones and establishes the position of the sound source randomly. The second one seeks to classify the pronunciation of ten simple English words with 10 dB SNR and fixed position. As in [9], [13] argues that the use of beamforming improves the performance of classifiers, but as long as the direction of the emitting source is known. More recently, authors have used beamforming to ascertain the influence of different sound sources on the total noise of an environment surrounded by factories, highways and train stations, making it possible to determine the most intense source [13]. In all articles mentioned, the authors apply the Delay-and-Sum (DaS) algorithm and achieve good classification performance. In [5], a real-time system for locating sound sources in the ESP32 micro-controller was developed, exposing the feasibility of future works mentioned above.

The current work leads contributions by addressing several procedures separately used in other articles, attempting to unify techniques and results that can be applied to a common objective. The report discusses comparatively the use of beamforming and machine learning techniques, relying on simulations of diffuse noise in a wide SNR range, seeking results both in accuracy and in processing time.

III. Algorithms

This section will briefly present the main fundamentals of beamforming and machine learning algorithms employed in this work.

A. Beamforming

Beamforming algorithms are based on the fact that a wave propagating in a certain direction of space will be captured at different times by each device present in an array of receivers (microphones, in the case of this work) [14]. The simplest of them, Delay-and-Sum, does exactly what its name says: settle the delays between each microphone and then add them up. This causes directivity patterns to be created, amplifying signals from a certain direction, and attenuating from others [15], [16]. To find the lags of each microphone concerning the referential, auto-correlation analysis in the time or frequency domain can be performed [17], [7].

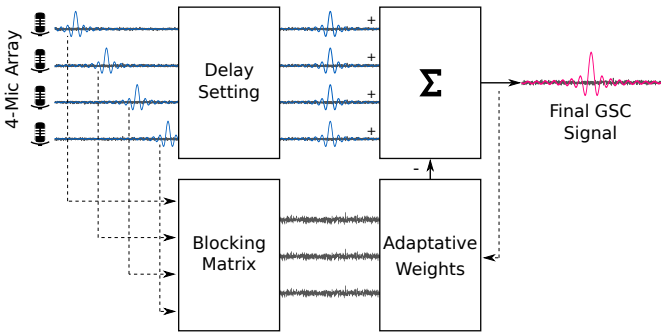

Fig. 2. GSC structure adopted in this work, that considers an arrangement of four microphones.

The so-called Generalized Sidelobe Canceller differs from the previous one for being adaptive. It attempts to minimize the power of noise in the resulting signal coming from some other fixed beamforming algorithm, such as DaS itself. To achieve this goal, GSC drops the signal of interest from the signal captured by each microphone using a Blocking Matrix. Subsequently, these signals are subtracted from the final signal using an adaptive multiplicative factor [15], [18]. Figure 2 shows the structure of the GSC, and the Equation 1,

$$B = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \quad (1)$$

shows the Blocking Matrix employed in this work. The algorithm used in the adaptive section was the Least Mean Squares (LMS).

B. Classifiers

Supervised learning algorithms are based on prior knowledge of training samples and their respective labels. Therefore, given a new sample to be classified, the algorithm must stipulate its label according to the already known data, mathematically described by a vector of characteristics (features) [19].

In this work, seven classifiers will be addressed: K Nearest Neighbors (KNN), Linear Discriminant Analysis (LDA), Perceptron, Quadratic Discriminant Analysis (QDA), Stochastic Gradient Descent (SGD), Support Vector Machine (SVM) and Decision Tree [19].

IV. Simulation Environment and Procedures

This section will describe the dataset and the procedures that were adopted to: increase the database in order to improve the performance of classifiers in noisy environments; find the best configuration for each classifier, so that these algorithms achieve superior performance, and; simulate a microphone

array for each audio in the dataset with different SNR values. In all stages of this work, Python was employed in an Intel (R) Xeon (R) E5-2650 v3 @ 2.30 GHz CPU and 16 GB of RAM.

A. Dataset

The dataset Sound Events for Surveillance Applications (SESA) [20] is composed of 480 audios up to 33 seconds length, divided between the classes *shot*, *explosion*, *alarm*, and *casual*, the latter being composed of audios that could be incorrectly classified as any of the other three classes, but which do not represent security risks, e.g. constructions, fireworks, thunders, horns, among others. All audios are WAV files in mono, sampled at 16 kHz with 8 bits of depth.

Attempting to improve the performance of the classifiers in scenarios in which the noise stands out from the signal of interest, the dataset underwent a process of Data Augmentation. The procedure consisted of adding white Gaussian noise to each audio in the dataset, with SNR ranging from -10 dB to 30 dB with step 5 dB.

B. Features Extraction

TABLE I
FEATURES VECTOR OF EACH AUDIO WINDOW.

Feature	Position
Root Mean Square	0
Spectral Centroid	1
Spectral Bandwidth	2
Spectral Flatness	3
Roll-Off Frequency	4
Zero Crossing Rate	5
MFCC	6–24
Delta	25–44
Delta Delta	45–64
Mel Spectrogram	65–74
Chromagram	75–86
Constant-Q Chromagram	87–98
Chroma Energy Normalized CENs	99–110
Tonnetz	111–117
Spectral Contrast	118–125

In order to perform the features extraction procedure, the files were segmented into 200 ms and 50 % overlap windows. Each window was designed as a feature vector, as shown in Table I, selected based on the discussions related to the works mentioned. After extraction, features were normalized. The work [10] explains the importance of this stage for the performance of the classifiers.

C. Gridsearch

Seven classifiers were submitted to gridsearch, a process that performs, for each classifier, a search for the hyper-parameter values that give the best classification performance. The cross-validation method used was Bootstrap, which consists of several reproductions of separation of data between training and testing, to obtain more reliability in the results achieved with the classifiers [19]. The mode of each window classification determines the final classification of single audio. It is important to regard that gridsearch was performed twice: first using the original dataset in the training phase; then, the augmented.

D. Microphone Arrangement Simulation

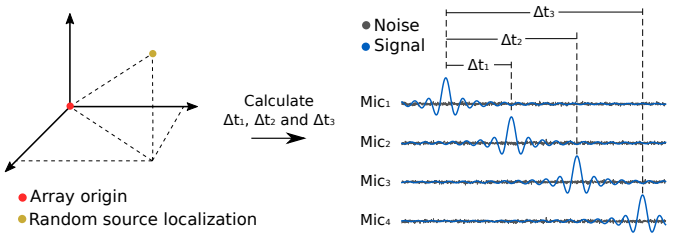

Fig. 3. Procedure adopted to simulate an array of microphones with diffuse noise for each audio and for each SNR.

After checking the best settings for the classifiers used in the original and augmented datasets, a procedure was used to simulate a microphone array so that these techniques could be evaluated. For this, the position of each microphone was designed to respect the positioning characteristics observed in the device called *ReSpeaker 4-Mic Array*, as it will be used in continuations of this work [1].

For each dataset audio, both azimuth and elevation angles were randomly defined so that the lags of each microphone concerning the reference could be calculated according to the array geometry. Following that, the audios were replicated with these same lags. Later on, white noise was added to each replicated signal, simulating a diffuse noise field [15]. This procedure was repeated for SNR values from -10 dB to 30 dB, with a step of 1 dB, as shown in Figure 3.

V. DISCUSSION AND ANALYSIS OF THE RESULTS

After simulating the microphones array with all dataset's audios and for several SNR values, three resulting signals were obtained:

1) **Without beamforming:** consisted of the sum of the signals from the four simulated microphones, without temporal displacement;
2) **With DaS:** the delays between the microphones were repaired before the sum of the signals was performed.
3) **With GSC:** the signal acquired after the DaS from the previous step was used to proceed with the GSC.

Using a hyper-parameter configuration that provides the best accuracy for classifiers for both original and augmented datasets, the calculated signals *without beamforming*, *with DaS*, and *with GSC* were used in the classification test step, where, again, bootstrap cross-validation was employed.

Table II brings comparative results between classifiers, beamforming and datasets for different SNR values. Figures 4 and 5 show in more detail the same results for SVM, the classifier that obtained the best performance in accuracy.

About the beamforming techniques applied, it was observed that they were successful in increasing the accuracy of all classifiers, in addition to having demonstrated consistency by

[1] http://wiki.seeedstudio.com/ReSpeaker_4_Mic_Array_for_Raspberry_Pi/

TABLE II

MEAN ACCURACY AND STANDARD DEVIATION ACHIEVED BEFORE AND AFTER DATA AUGMENTATION.

Classifier	Beamforming	Original Dataset					Data Augmentation				
		-10 dB	*0 dB*	*10 dB*	*20 dB*	*30 dB*	*-10 dB*	*0 dB*	*10 dB*	*20 dB*	*30 dB*
KNN	No Beamforming	0.605 ± 0.021	0.649 ± 0.033	0.710 ± 0.016	0.784 ± 0.017	0.828 ± 0.012	0.697 ± 0.018	0.796 ± 0.014	0.847 ± 0.019	0.840 ± 0.011	0.840 ± 0.007
	DaS	0.607 ± 0.020	0.691 ± 0.025	0.761 ± 0.012	0.851 ± 0.021	**0.878 ± 0.007**	0.769 ± 0.016	0.820 ± 0.007	0.836 ± 0.009	0.840 ± 0.011	0.849 ± 0.003
	GSC	0.619 ± 0.013	0.695 ± 0.015	0.754 ± 0.018	0.855 ± 0.019	0.868 ± 0.013	0.750 ± 0.018	0.822 ± 0.004	0.838 ± 0.019	0.855 ± 0.015	0.838 ± 0.006
LDA	No Beamforming	0.542 ± 0.034	0.670 ± 0.018	0.721 ± 0.026	0.720 ± 0.017	0.702 ± 0.007	0.622 ± 0.011	0.697 ± 0.013	0.739 ± 0.011	0.786 ± 0.012	0.744 ± 0.007
	DaS	0.580 ± 0.015	0.681 ± 0.009	0.744 ± 0.021	0.788 ± 0.003	0.809 ± 0.017	0.645 ± 0.011	0.716 ± 0.018	0.731 ± 0.016	0.767 ± 0.009	0.777 ± 0.007
	GSC	0.598 ± 0.033	0.687 ± 0.019	0.758 ± 0.017	0.782 ± 0.018	0.811 ± 0.003	0.640 ± 0.007	0.704 ± 0.010	0.723 ± 0.008	0.777 ± 0.007	0.775 ± 0.004
Perceptron	No Beamforming	0.657 ± 0.024	0.723 ± 0.026	0.733 ± 0.017	0.771 ± 0.036	0.691 ± 0.045	0.582 ± 0.058	0.710 ± 0.029	0.744 ± 0.037	0.754 ± 0.015	0.676 ± 0.073
	DaS	0.655 ± 0.036	0.735 ± 0.016	0.792 ± 0.025	0.801 ± 0.023	0.819 ± 0.032	0.659 ± 0.055	0.740 ± 0.018	0.763 ± 0.035	0.771 ± 0.046	0.769 ± 0.058
	GSC	**0.670 ± 0.046**	**0.746 ± 0.009**	0.784 ± 0.019	0.807 ± 0.029	0.815 ± 0.025	0.649 ± 0.050	0.735 ± 0.025	0.769 ± 0.029	0.760 ± 0.058	0.767 ± 0.064
QDA	No Beamforming	0.380 ± 0.023	0.659 ± 0.030	0.601 ± 0.007	0.592 ± 0.023	0.554 ± 0.019	0.392 ± 0.003	0.660 ± 0.009	0.672 ± 0.009	0.615 ± 0.009	0.535 ± 0.011
	DaS	0.403 ± 0.038	0.659 ± 0.022	0.687 ± 0.016	0.678 ± 0.009	0.687 ± 0.052	0.428 ± 0.010	0.685 ± 0.006	0.803 ± 0.009	0.758 ± 0.004	0.754 ± 0.007
	GSC	0.400 ± 0.037	0.659 ± 0.025	0.680 ± 0.014	0.687 ± 0.028	0.697 ± 0.043	0.409 ± 0.010	0.678 ± 0.003	0.798 ± 0.007	0.763 ± 0.003	0.737 ± 0.009
SGD	No Beamforming	0.529 ± 0.017	0.607 ± 0.015	0.653 ± 0.021	0.723 ± 0.019	0.699 ± 0.030	0.714 ± 0.023	0.754 ± 0.016	0.796 ± 0.009	0.819 ± 0.008	0.767 ± 0.029
	DaS	0.565 ± 0.018	0.641 ± 0.017	0.731 ± 0.020	0.807 ± 0.015	0.822 ± 0.004	0.708 ± 0.012	0.784 ± 0.011	0.824 ± 0.015	0.853 ± 0.007	0.824 ± 0.012
	GSC	0.577 ± 0.017	0.634 ± 0.009	0.737 ± 0.024	0.817 ± 0.012	0.817 ± 0.007	0.720 ± 0.007	0.775 ± 0.011	0.830 ± 0.007	0.841 ± 0.009	0.824 ± 0.012
SVM	No Beamforming	0.548 ± 0.014	0.700 ± 0.024	0.773 ± 0.016	0.820 ± 0.011	0.811 ± 0.009	0.775 ± 0.014	0.775 ± 0.012	0.843 ± 0.004	0.840 ± 0.007	0.803 ± 0.004
	DaS	0.561 ± 0.022	0.733 ± 0.024	**0.803 ± 0.007**	0.859 ± 0.018	0.851 ± 0.009	**0.819 ± 0.013**	**0.828 ± 0.008**	**0.847 ± 0.010**	0.855 ± 0.013	0.849 ± 0.007
	GSC	0.563 ± 0.011	0.740 ± 0.022	0.803 ± 0.019	**0.859 ± 0.016**	0.847 ± 0.012	0.817 ± 0.012	0.824 ± 0.009	0.845 ± 0.011	**0.860 ± 0.015**	**0.853 ± 0.007**
Tree	No Beamforming	0.552 ± 0.074	0.613 ± 0.056	0.704 ± 0.019	0.729 ± 0.039	0.679 ± 0.041	0.651 ± 0.031	0.754 ± 0.026	0.798 ± 0.037	0.742 ± 0.045	0.668 ± 0.013
	DaS	0.601 ± 0.033	0.687 ± 0.041	0.786 ± 0.034	0.790 ± 0.033	0.811 ± 0.003	0.657 ± 0.028	0.775 ± 0.034	0.819 ± 0.044	0.822 ± 0.026	0.796 ± 0.029
	GSC	0.619 ± 0.036	0.685 ± 0.043	0.786 ± 0.036	0.796 ± 0.023	0.815 ± 0.018	0.662 ± 0.037	0.779 ± 0.044	0.819 ± 0.026	0.826 ± 0.016	0.798 ± 0.033

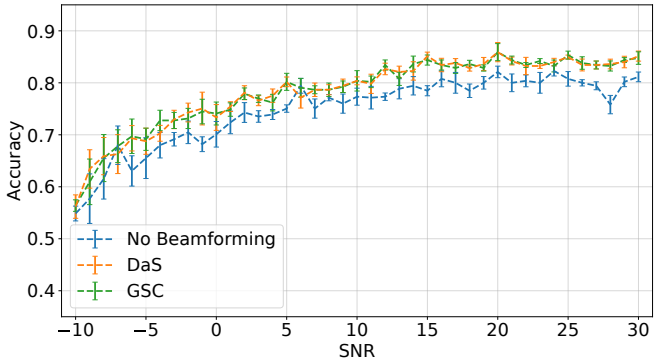

Fig. 4. Mean accuracy and standard deviation of SVM as a function of SNR for different beamforming approaches when the classifier was trained with the original dataset.

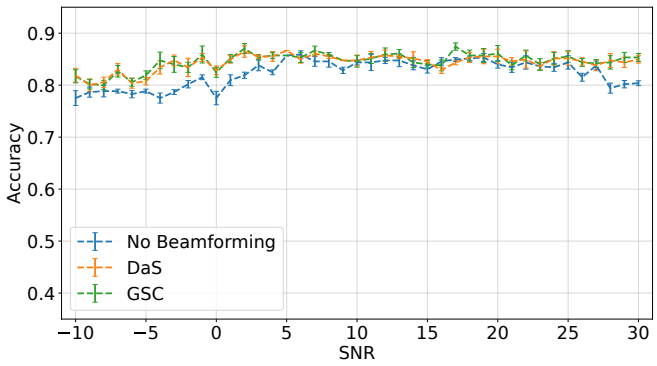

Fig. 5. Mean accuracy and standard deviation of SVM as a function of SNR for different beamforming approaches when the classifier was trained with the augmented dataset.

presenting a practically constant improvement for all tested noise scenarios. The gain was approximately the same for both DaS and GSC.

Regarding the data augmentation at training step, there was an even more positive result: the classifiers showed a remarkably expressive increase in the noisiest scenarios and, in some cases, an irrelevant loss in the less noisy ones. Besides, the standard deviation has decreased, making the classifiers even more robust. When using SVM in combination with beamforming techniques, for example, the accuracy was greater than 80 % in all scenarios. In general, the non-parametric classifiers, SVM and KNN, were the ones that managed to reach the highest accuracy values, corroborating the results obtained in [10].

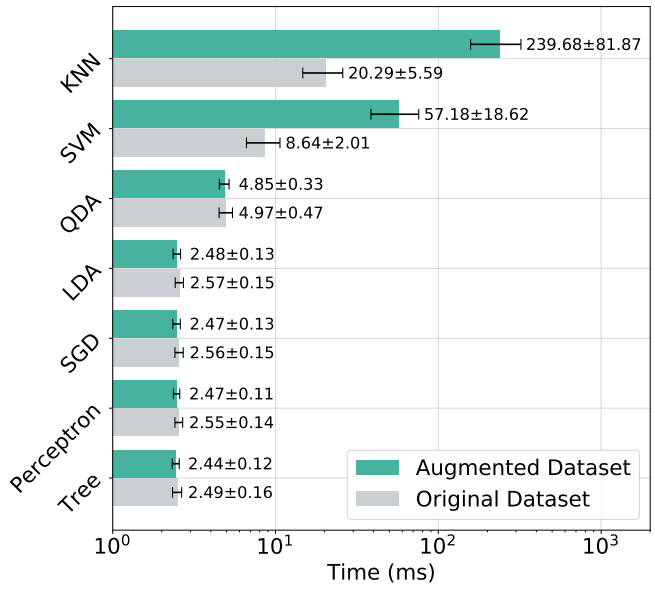

Fig. 6. Average time and standard deviation required to classify all windows of a single audio when training was done with the original and augmented datasets.

As mentioned, the computational cost of each algorithm was also a concern. The tests indicated that beamforming techniques were not influenced as a result of the SNR, nor concerning the data augmentation since they were applied before classification. The average and standard deviation of processing time for these algorithms were:

- *DaS*: (22.14 ± 13.23) ms.
- *GSC*: (397.09 ± 648.65) ms.

Figure 6 shows, in logarithmic scale, the mean and standard

deviation of the time needed to classify all windows from the same audio, before and after data augmentation. As expected, parametric classifiers were not affected after this change. However, those that had the best accuracy score had sudden changes. The first, SVM, increased by 561.80 %, while the second, KNN, 1081.27 %. These changes may compromise future applications in real-time.

Analyzing the results, it appears that beamforming techniques were much more costly when compared to the classifiers. Specifically about GSC, since it obtained a processing 18 times longer than DaS and did not present great accuracy differences, it's usage is not suggested in the next stages of this work.

It is assumed that the most appropriate configuration in terms of accuracy is that which combines the augmented dataset, DaS, and SVM. Even so, concerning the processing time, SGD Classifier also deserves attention, since it has also achieved good accuracy results, and, unlike SVM, its processing time was not affected by data augmentation.

VI. Conclusions

The work presented a comparative study between seven classifiers and two beamforming techniques aiming to find an efficient configuration for the construction of an audio-based public security system. The results revealed that the best performance in accuracy was obtained with the combination of DaS and SVM algorithms. Furthermore, Data Augmentation guaranteed accuracy above 80 % for all examined SNR values. Despite that, SGD Classifier conferred more satisfactory results in processing time, although the classification performance was slightly lower. Besides, it was found that the processing time of beamforming was higher than the classifiers', which may compromise future real-time applications if they do not receive improvements.

In future work, it is advisable to investigate other beamforming techniques and perform analyses with other databases, including noise bases. Also, the aim is to implement the system discussed on an embedded platform that allows the coupling of an array of microphones, and that can run in real-time.

References

[1] M. Crocco, M. Cristani, A. Trucco, and V. Murino, "Audio surveillance: A systematic review," *ACM Comput. Surv.*, vol. 48, no. 4, 2016.

[2] T. D. Räty, "Survey on contemporary remote surveillance systems for public safety," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 5, pp. 493–515, 2010.

[3] M. Mulimani and S. G. Koolagudi, "Extraction of mapreduce-based features from spectrograms for audio-based surveillance," *Digital Signal Processing*, vol. 87, pp. 1 – 9, 2019.

[4] S. U. Hassan, M. Zeeshan Khan, M. U. Ghani Khan, and S. Saleem, "Robust sound classification for surveillance using time frequency audio features," in *2019 International Conference on Communication Technologies (ComTech)*, 2019, pp. 13–18.

[5] G. Fabregat, J. A. Belloch, J. M. Badía, and M. Cobos, "Design and implementation of acoustic source localization on a low-cost iot edge platform," *IEEE Transactions on Circuits and Systems II: Express Briefs*, pp. 1–1, 2020.

[6] T. Spadini, D. L. d. O. Silva, and R. Suyama, "Sound event recognition in a smart city surveillance context," *arXiv preprint arXiv:1910.12369*, 2019.

[7] P. Vecchiotti, G. Pepe, E. Principi, and S. Squartini, "Detection of activity and position of speakers by using deep neural networks and acoustic data augmentation," *Expert Systems with Applications*, vol. 134, pp. 53 – 65, 2019.

[8] Y. Su, K. Zhang, J. Wang, D. Zhou, and K. Madani, "Performance analysis of multiple aggregated acoustic features for environment sound classification," *Applied Acoustics*, vol. 158, p. 107050, 2020.

[9] M. R. Bai, S. Lan, and J. Huang, "Time difference of arrival (tdoa)-based acoustic source localization and signal extraction for intelligent audio classification," in *2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, 2018, pp. 632–636.

[10] N. Surampudi, M. Srirangan, and J. Christopher, "Enhanced feature extraction approaches for detection of sound events," in *2019 IEEE 9th International Conference on Advanced Computing (IACC)*, 2019, pp. 223–229.

[11] P. Foggia and M. Vento, "Reliable detection of audio events in highly noisy environments," July 2015. [Online]. Available: http://dx.doi.org/10.1016/j.patrec.2015.06.026

[12] S. Scardapane, M. Scarpiniti, M. Bucciarelli, F. Colone, M. V. Mansueto, and R. Parisi, "Microphone array based classification for security monitoring in unstructured environments," *AEU - International Journal of Electronics and Communications*, vol. 69, no. 11, pp. 1715 – 1723, 2015.

[13] J. Murovec, J. Prezelj, L. Čurović, and T. Novaković, "Microphone array based automated environmental noise measurement system," *Applied Acoustics*, vol. 141, pp. 106 – 114, 2018.

[14] H. Krim and M. Viberg, "Two decades of array signal processing research: The parametric approach," *Signal Processing Magazine, IEEE*, vol. 13, pp. 67 – 94, 08 1996.

[15] I. McCowan, "Microphone arrays: A tutorial," *Citeseer*, 2001.

[16] Y. Zeng and R. C. Hendriks, "Distributed delay and sum beamformer for speech enhancement via randomized gossip," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 1, pp. 260–273, 2014.

[17] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 4, pp. 320–327, 1976.

[18] G. Rombouts, A. Spriet, and M. Moonen, "Generalized sidelobe canceller based combined acoustic feedback- and noise cancellation," *Signal Processing*, vol. 88, no. 3, pp. 571 – 581, 2008.

[19] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, ser. Springer Series in Statistics. Springer New York Inc., 2001.

[20] T. Spadini, "Sound events for surveillance applications dataset," Oct. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3519845