Comparison of Multiclass SVM Classification Techniques in an Audio Surveillance Application under Mismatched Conditions

Roneel V. Sharan and Tom J. Moir
School of Engineering
Auckland University of Technology
Private Bag 92006, Auckland 1142, New Zealand
Email: roneel.sharan@aut.ac.nz, tom.moir@aut.ac.nz

Abstract—**In this paper, we compare the performance of classification techniques for multiclass support vector machines in an unstructured environment. In particular, we consider the following methods: one-against-all, one-against-one, decision directed acyclic graph, and adaptive directed acyclic graph. The performance is compared in terms of classification accuracy, training time, and evaluation time. An audio surveillance application is looked at under different noise conditions and varying signal-to-noise ratio with mel-frequency cepstral coefficients and other commonly used time and frequency domain features. The results show that while there isn't much difference in the classification accuracy using the four approaches under clean and low noise conditions, the one-against-all method was found to give relatively better classification accuracy in high noise conditions when trained with clean samples only. However, the results were much more even with multi-conditional training. Also, the training time for the one-against-all approach was found to increase significantly as the training data increased fourfold while the one-against-one approach showed a significantly higher evaluation time.**

Keywords—audio surveillance; signal-to-noise ratio; sound recognition; support vector machines

I. INTRODUCTION

Initially intended as a binary classifier, a number of methods have since been developed to use support vector machines (SVMs) for multiclass classification. The most common technique in solving the multiclass problem is to reduce it into multiple binary classification problems. Four of the widely used methods based on this approach are: one-against-all (OAA), one-against-one (OAO), decision directed acyclic graph (DDAG), and adaptive directed acyclic graph (ADAG).

OAA, which is probably the earliest of the multiclass SVM classification techniques [1, 2], distinguishes between one of the class labels against the rest. During classification, the classifier that has the highest output function assigns the class. The OAO approach distinguishes between every pair of classes and classification is done using a max-wins voting strategy [3]. Every classifier assigns the instance to one of the two classes with the vote for the assigned class increased by one. In the end, the class with the most votes assigns the class label. DDAG [4] and ADAG [5] are also based on classification between pair of classes but utilize a decision tree structure in the testing phase.

In [6], Hsu and Lin compare OAA, OAO, DDAG and two altogether methods on large problems and conclude OAO and DDAG as being more suitable for practical use. A similar comparison is done by Seo [7] using OAA, OAO, DDAG together with the approach given by Weston [8] and Crammer [9] for a face recognition application. OAO was found to give the best results followed by DDAG but they suggest DDAG due to its low computational cost. In [10], a bottom-up binary tree architecture, which is similar to the ADAG method, is presented to reduce the number of comparison during testing in an audio classification application. Some other similar work but with some modifications to the architecture and employing different databases can be found in [11-13].

Although the OAA method doesn't seem to be the preferred choice in most cases, in most literature, the difference in terms of classification accuracy is marginal and, as such, the comparison between the methods are largely based on training and evaluation times. However, there is hardly any literature doing such a comparison under noise conditions which is the key contribution in this work. In this work, we compare the performance of OAA, OAO, DDAG, and ADAG multiclass SVM classification methods in an audio surveillance application under different noise conditions and signal-to-noise ratio (SNR).

The rest of this paper is organized as follows. Section II gives an overview of SVMs and the four multiclass classification techniques that we investigate in this work. Section III presents the features used in this work which include mel-frequency cepstral coefficients (MFCCs) and some common time and frequency domain features. Section IV is on the experimentations we carried out and the corresponding results while conclusion and future recommendations are given in Section V.

II. SUPPORT VECTOR MACHINES

A. Basic Theory

A support vector machine determines the optimal hyperplane to maximize the distance between any two given classes. It has been well described in many literature, such as

978-1-4799-4612-9/14/$31.00

in [1, 14-16], and is summarized here. Starting with a case of linearly separable dataset, consider a set of l training samples belonging to two classes, a positive class and a negative class, given as $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\}$, where $\mathbf{x}_i \in R^d$ is a d-dimensional feature vector representing the i^{th} training sample, and $y_i \in \{-1, +1\}$ is the class label of $\mathbf{x}_i$. There can be many possible hyperplanes but the two classes can be said to be optimally separated by the hyperplane if the separation distance, or margin, between the closest vector, known as support vectors, to the hyperplane is maximal.

Any hyperplane in the feature space can be described by the equation $\mathbf{w} \cdot \mathbf{x} + b = 0$, where $\mathbf{w} \in R^d$ is a normal vector to the hyperplane and b is a constant. Selecting two hyperplanes, $\mathbf{w} \cdot \mathbf{x} + b = +1$ and $\mathbf{w} \cdot \mathbf{x} + b = -1$ such that the data points are separated with no data between them in the margin region, the aim then is to maximize the distance between them. The distance between these two hyperplanes is given as $^{\mathbf{2}}/_{\|\mathbf{w}\|}$, therefore, $\|\mathbf{w}\|$ has to be minimized. To prevent the data points from falling into the margin, the following constraints are added: $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, i = 1, 2, \ldots l$. For mathematical convenience, and without altering the solution, $\|\mathbf{w}\|$ is substituted with $½\|\mathbf{w}\|^2$ which becomes a quadratic programming problem. The optimization problem can be solved under the given constraints by the saddle point of the Lagrange functional and, for ease of computation, the primal problem is transformed to a dual problem using classical Lagrangian duality which gives the solution

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i. \tag{1}$$

where α_i are the non-negative Lagrange multipliers. The $\mathbf{x}_i$ for which $\alpha_i > 0$ are called the supported vectors which lie exactly on the margin satisfying $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) = 1$. The offset can then be determined as

$$b = y_i - \mathbf{w} \cdot \mathbf{x}_i \tag{2}$$

using any support vector or averaged over all support vectors.

However, there is no such hyperplane for linearly nonseparable problems to classify every training sample correctly. In such a case, the optimization can be generalized by introducing the concept of *soft margin* implying a hyperplane separating most but not all the points. Introducing non-negative *slack* variables ξ_i which measure the degree of misclassification of data $\mathbf{x}_i$ and a penalty function $\sum_i \xi_i$, the optimization is a trade-off between a large margin and a small error penalty. The optimization problem can be solved as before and the solution is similar to the separable case except for a modification to the Lagrange multipliers: $0 \leq \alpha_i \leq C, i = 1, 2, \ldots l$, where C is a penalty or tuning parameter to balance the margin and training error.

In applications where linear SVM does not give satisfactory results, nonlinear SVM is suggested which aims to map the input vector $\mathbf{x}$ to a higher dimensional space $\mathbf{z}$ through some nonlinear mapping $\phi(\mathbf{x})$ chosen *a priori* to construct an optimal hyperplane. The *kernel trick* [16] is applied to create the nonlinear classifier where the dot product is replaced by a nonlinear kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$ which computes the inner product of the vectors $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$.

The typical kernel functions are: polynomial, $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^r$ where r is the degree of the polynomial; Gaussian radial basis function (RBF), $K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2\right)$, where $\sigma > 0$ is the width of the Gaussian function; and multilayer perception, $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(a_1(\mathbf{x}_i \cdot \mathbf{x}_j) - a_2)$, where a_1 and a_2 are two given parameters known as *scale* and *offset* respectively.

The classifier for a given kernel function with the optimal separating hyperplane is then given as

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right). \tag{3}$$

B. Multiclass Classification

1) One-Against-All SVM

Consider an M-class problem with l training samples: $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\}$, where $\mathbf{x}_i \in R^d$ is a d-dimensional feature vector representing the i^{th} training sample, and $y_i \in \{1, 2, \ldots, M\}$ is the class label of $\mathbf{x}_i$. In the OAA-SVM approach, M binary SVM classifiers are constructed and evaluated where each classifier separates one class from all the other classes combined. That is, the i^{th} classifier is trained with all the training samples from the i^{th} class as positive labels and all the remaining samples as negatives labels.

During classification, a sample $\mathbf{x}$ is classified in the class with the largest value of the decision function

$$f(\mathbf{x}) = \arg\max_{i=1,2,\ldots,M} \left(\mathbf{w}^i \cdot \phi(\mathbf{x}) + b^i\right). \tag{4}$$

The disadvantage of OAA-SVM is the high mismatch in the training samples between the positive and negative classes while some literature [4, 17] also shows that the training and evaluation times are relatively high.

2) One-Against-One SVM

For an M-class problem, OAO-SVM constructs and evaluates $M(M-1)/2$ classifiers where each SVM is trained on samples from two classes at a time, that is, using training samples from the i^{th} and j^{th} class. During classification, the class label of a test sample is predicted as

$$f(\mathbf{x}) = \arg\max_{i=1,2,\ldots,M} \sum_{j=1, j\neq i}^{M} \operatorname{sgn}\left(\mathbf{w}^{ij} \cdot \phi(\mathbf{x}) + b^{ij}\right). \tag{5}$$

While OAO-SVM has much more uniform training samples in the positive and negative classes when compared to OAA-SVM, its disadvantage is the inefficiency of classifying data because the number of SVM classifiers grows super linearly with an increase in the number of classes. DDAG and ADAG techniques remedy this disadvantage using a decision tree architecture.

3) Decision Directed Acyclic Graph

The structure of a rooted binary DAG by Platt et al. [4] is shown in Fig. 1. A rooted binary tree has nodes arranged in a triangle. The single root node is at the top, two nodes in the second layer, and so on with M leaves in the last layer where M is the number of classes. The i^{th} node in layer $j < M$ is connected to the i^{th} and $(i+1)^{th}$ node in the $(j+1)^{th}$ layer.

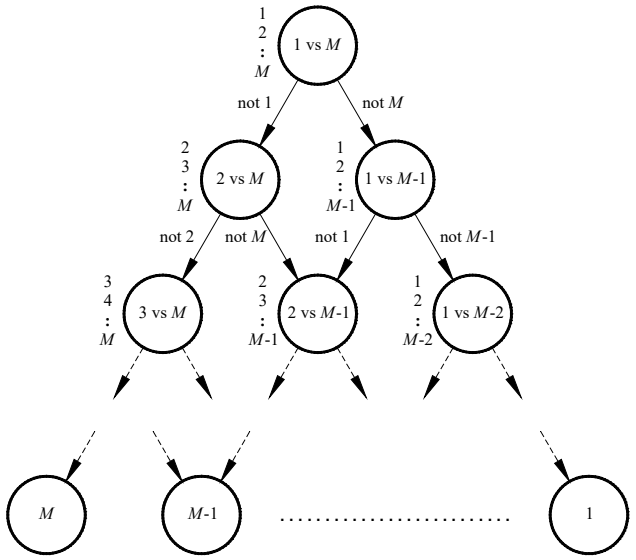

Figure 1. DDAG for an M-class problem. The root node is at the top of the tree and there are M-leaves at the bottom of the tree. Evaluation starts at the root node from where each class is removed from the class order list at each node. Only one class is left at the leaf node which is the decision function.

The evaluation of a DDAG starts at the root node and the node is exited through the left edge if the outcome is zero and the right edge otherwise. The binary function at the next node is then evaluated and this continues until the leaf node is reached, which is the value of the decision function. The DDAG operates on a class order list which is initialized at the root node. The list is updated at each subsequent node where one class is eliminated from the list. The evaluation at each node corresponds to the first and last classes in the list. There is only one class left in the list after $M-1$ evaluations. As mentioned in [4], the choice of the class order in the list is arbitrary and in their experimentation, a class list in numerical/alphabetical order was used since a few different combination of class order did not show significant changes in the accuracy.

Similar to OAO-SVM, DDAG-SVM creates $M(M-1)/2$ nodes during training phase but only $M-1$ nodes are evaluated during testing. As such, DDAG outperforms OAO in terms of computation speed. However, as pointed out by Kijsirikul et al. in [5], the node evaluations for the correct class is unnecessarily high which creates high cumulative error. On average, the number of times a correct class has to be tested against other classes scales linearly with M. In a worst case scenario, if the correct class is evaluated at the root node, it will be tested $M-1$ times, that is, tested against all the other classes, before being correctly classified.

4) Adaptive Directed Acyclic Graph

Adaptive DAG is proposed by Kijsirikul et al. in [5] aimed at overcoming the shortcomings of DDAG-SVM. Similar to DDAG, for an M-class problem, $M(M-1)/2$ binary classifiers are trained and $M-1$ evaluations are required during testing. However, an ADAG has a reversed triangular structure when compared to a DDAG as shown in Fig. 2 for an M-class problem where M is assumed to be an even number for now.

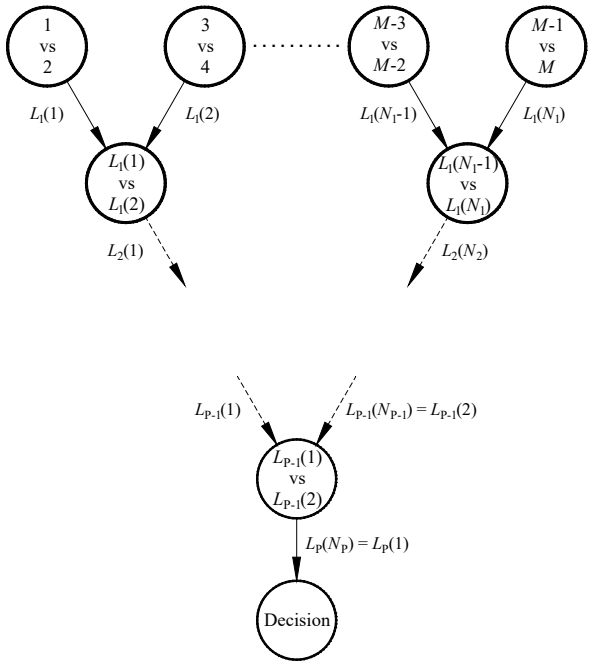

Figure 2. ADAG structure for an M-class problem (M assumed to be even) where L_p is the p^{th} layer, N_p is the number of nodes in the p^{th} layer, $L_p(q)$ is the output of the q^{th} node in the p^{th} layer, $q = 1, 2, \dots, N_p$, and $p = 1, 2, \dots, P$; $p = 1$ is the top layer.

Similar to DDAG, ADAG is implemented using a class order list, each node evaluates two classes, and a class is eliminated at each node. The classification starts at the top layer and based on the outcome of the binary function, the outgoing edge from the node passes the preferred class information to the next node. The top layer has $M/2$ nodes, the second layer has $M/2^2$ nodes, and so on. In general, the number of nodes in each layer is equal to $M/2^p$ where $p = 1, 2, \dots, P$ is the layer number starting from the top layer.

The elimination process continues at each node with the number of classes reducing by half in each layer until the final node, the output of which is the decision function. While the same number of evaluations are required as in DDAG-SVM, the number of evaluations that the correct class has to go through is $\lceil \log_2 M \rceil$, which is also equal to the number of layers, when compared to a maximum of $M-1$ evaluations for the correct class in DDAG-SVM. In the case of odd number of classes, the last class in the list is not evaluated at a node until the number of classes in the list becomes even.

III. Feature Selection

In this work, we consider MFCCs and some other commonly used time and frequency domain features which are described below.

A. MFCC

Firstly, the Discrete Fourier Transform (DFT) is applied to the windowed signal as

$$X_t(k) = \sum_{n=0}^{N-1} x(n) w(n) e^{\frac{-2\pi i k n}{N}}, \; k = 0, \dots, N-1 \qquad (6)$$

where N is the window length, $x(n)$ is the time-domain signal, $X_t(k)$ is the k^{th} harmonic corresponding to the frequency $f(k) = kF_s/N$ for the t^{th} frame, F_s is the sampling frequency, and $w(n)$ is the window function.

Next, a triangular mel filterbank is applied to the linear spectra and the energy in each filter is added. The discrete cosine transform (DCT) of the log power of these values are then computed from which the MFCCs are obtained.

B. Zero-Crossing Rate (ZCR)

Zero-crossing rate is the number of time-domain zero-crossings within a frame and is a simple measure of the frequency content of a signal given as

$$ZCR = \frac{1}{2(N-1)}\sum_{n=1}^{N-1}\left|\operatorname{sgn}\left[x(n+1)\right]-\operatorname{sgn}\left[x(n)\right]\right| \quad (7)$$

where $\operatorname{sgn}[\cdot]$ is a sign function: $\operatorname{sgn}[x(n)] = 1, x(n) \geq 0$; $\operatorname{sgn}[x(n)] = -1, x(n) < 0$.

C. Short-Time Energy (STE)

Short-time energy is the total spectrum power of a frame given as

$$STE = \log\left(\int_0^{w_0}\left|X(w)\right|^2 dw\right) \quad (8)$$

where $X(w)$ denotes the DFT coefficients, $|X(w)|^2$ is the power at the frequency w, and w_0 is the half sampling frequency or Nyquist frequency.

D. Sub-Band Energy (SBE)

Sub-band energy is the ratio between sub-band power and the total power in a frame given as

$$SBE = \frac{1}{STE}\int_{L_j}^{H_j}\left|X(w)\right|^2 dw \quad (9)$$

where L_j and H_j are the lower and upper bound of sub-band j respectively with the frequency spectrum divided into four sub-bands: $[0, {}^{w_0}/_8], [{}^{w_0}/_8, {}^{w_0}/_4], [{}^{w_0}/_4, {}^{w_0}/_2], [{}^{w_0}/_2, w_0]$.

E. Spectral Centroid (SC)

Spectral centroid, also called brightness, is the frequency centroid of the spectrum or the balancing point of the spectral power distribution and is given as

$$SC = w_c = \frac{\int_0^{w_0} w\left|X(w)\right|^2 dw}{\int_0^{w_0}\left|X(w)\right|^2 dw}. \quad (10)$$

F. Bandwidth (BW)

Bandwidth is the square root of the power-weighted average of the squared difference between the spectral components and frequency centroid given as

$$BW = \sqrt{\frac{\int_0^{w_0}\left(w-w_c\right)^2\left|X(w)\right|^2 dw}{\int_0^{w_0}\left|X(w)\right|^2 dw}}. \quad (11)$$

G. Spectral Roll-Off (SR)

Spectral roll-off is the frequency below which a certain amount of power spectrum lies and can be determined as

$$SR = \max\left\{K \mid \sum_{k=0}^{K}\left|X_t(k)\right|^2 < A\sum_{k=0}^{N}\left|X_t(k)\right|^2\right\} \quad (12)$$

where A is an empirical constant ranged between zero and one (commonly used value is 0.95) and normally half the size of the DFT is used.

IV. Experimental Evaluation

A description of the database of sounds used in this work is given first followed by an overview of the noise conditions and the experimental setup. We then present the classification accuracy for the multiclass SVM classification techniques using MFCCs as the only features and, for comparison, MFCCs combined with the time and frequency domain features which is a common approach in most sound recognition applications. We also compare the training and evaluation time for the multiclass SVM classification techniques.

A. Description of Sound Database

The sound database consists of 10 classes: *alarms*, *children voices*, *construction*, *dog barking*, *footsteps*, *glass breaking*, *gunshots*, *horn*, *machines*, and *phone rings*. The sound files are largely obtained from the Real World Computing Partnership (RWCP) Sound Scene database in Real Acoustic Environment [18] and the BBC Sound Effects library [19]. All signals in the database have 16-bit resolution and a sampling frequency of 44100 Hz. The choice of the sound classes is similar to most other audio surveillance applications, [20] in particular.

B. Noise Conditions

The performance of the different features and classification methods are investigated under three different noise environments taken from the NOISEX-92 database [21]: *speech babble*, *factory floor 1*, and *destroyer control room*. The signals are resampled at 44100 Hz and the overall performance is measured in clean conditions and at 20dB, 10dB, and 0dB SNR.

C. Experimental Setup

For all experiments, features were extracted from a Hamming window of 512 points (11.61 ms) with 50% overlap. The four multiclass SVM classification techniques: OAA, OAO, DDAG, and ADAG are compared in each of the experiments. All results reported are using a nonlinear SVM with a Gaussian RBF kernel as it was found to give the best results during preliminary experiments. The penalty parameter C and σ for the Gaussian RBF kernel were tuned using cross validation. For DDAG and ADAG, the class order list in numerical order was used. Results using K-Nearest Neighbor (KNN) classification with Euclidean distance measure are also presented for comparison.

The system is trained with two-third of the clean samples with all remaining data used for testing. Under multi-

TABLE I. COMPARISON OF CLASSIFICATION ACCURACY - TRAINING USING CLEAN SAMPLES ONLY

Classification Method	MFCC Only					MFCC+ZCR+STE+SBE+SC+BW+SR				
	Clean	20dB	10dB	0dB	Average	Clean	20dB	10dB	0dB	Average
OAA-SVM	98.43	90.81	69.03	41.56	**74.96**	99.21	93.21	74.22	40.77	**76.85**
OAO-SVM	98.16	90.52	65.65	36.48	72.70	99.21	92.10	68.80	37.59	74.42
DDAG-SVM	98.16	91.28	63.43	35.58	72.11	99.21	92.65	68.62	35.84	74.08
ADAG-SVM	98.16	91.89	65.12	37.12	73.08	99.21	92.68	70.40	36.60	74.72
KNN	96.59	87.17	57.63	31.73	68.28	97.64	89.06	61.10	32.52	70.08

TABLE II. COMPARISON OF CLASSIFICATION ACCURACY - MULTI-CONDITIONAL TRAINING

Classification Method	MFCC Only					MFCC+ZCR+STE+SBE+SC+BW+SR				
	Clean	20dB	10dB	0dB	Average	Clean	20dB	10dB	0dB	Average
OAA-SVM	97.90	93.82	91.83	94.14	**94.42**	98.16	93.38	95.10	96.33	95.74
OAO-SVM	96.59	93.76	92.07	91.08	93.37	97.90	93.61	94.72	96.85	**95.77**
DDAG-SVM	96.59	93.79	92.13	91.08	93.39	97.90	93.58	94.72	96.68	95.72
ADAG-SVM	96.33	93.70	92.27	90.99	93.32	97.90	93.61	95.13	96.33	95.74
KNN	96.33	86.56	80.84	90.38	88.52	97.38	88.66	86.24	94.14	91.60

conditional training, two-third data from clean samples and at 0dB SNR are used for training while all remaining data is used for testing. With MFCCs as the only features, the feature vector for each frame is 36-dimensional: 12 MFCCs with the 0^{th} component excluded, using a 23-filterbank system, plus deltas and accelerations. When combined with the time and frequency domain features considered in this work, we get a 45-dimensional feature vector with the 9 additional features as follows: ZCR, STE, SBE (four subbands), SC, BW, and SR.

The overall size of the feature vector for a signal is $36 \times F$ using MFCCs only and $45 \times F$ with the inclusion of the time and frequency domain features, where F is the number of frames in the sound signal, which is different in each case. After data normalization, the final feature vector is represented by concatenating the mean and standard deviation for each dimension. As such, the final feature vector is 72-dimensional with MFCCs only and 90-dimensional when combined with the time and frequency domain features.

D. Results

The classification accuracy with MFCCs only and its combination with the time and frequency domain features is given in Table I. With MFCCs only, the minimum classification accuracy in clean conditions is 98.16% for the SVM methods and is 96.59% for KNN. However, the classification accuracy reduces greatly with the addition of noise, especially at 10dB and 0dB SNR with the highest classification accuracy at 69.03% and 41.56%, respectively. Also, there is only a slight increase in the average classification accuracy with the addition of the time and frequency domain features considered in this work. Possibly different combination of features need to be experimented with as addition of new features does not necessarily increase the classification accuracy as seen in [22].

In addition, the multiclass SVM classification techniques give a better overall classification accuracy than KNN in both the cases. While generally there isn't a significant difference in the classification accuracy using the four methods in clean and low noise (20dB SNR) conditions, the OAA-SVM approach does better at 10dB and 0dB SNR.

As presented in Table II, much better classification accuracy is obtained under noisy conditions with multi-conditional training. Using MFCCs only, at 0dB SNR, a maximum classification accuracy of 94.14% is achieved with OAA-SVM which increases to 96.85% with OAO-SVM with the addition of the time and frequency domain features. Also, the multiclass SVM classification methods once again give better results than KNN but the results are much more even with multi-conditional training for the SVM methods.

In Table III and IV, we compare the training and evaluation time for the results given in Table I and II respectively. The OAO, DDAG, and ADAG approaches have the same training procedure and time. The training time for these three methods is slightly higher than OAA when using only the clean samples for training as given in Table III. However, with multi-conditional training, results given in Table IV, the training time for OAA method increases significantly, about 847% for MFCCs only and 754% with the combined features, as the training data increases fourfold. As for the evaluation time, the DDAG and ADAG methods are the fastest while the evaluation time for the OAO method is significantly greater than the other methods. The DDAG and ADAG methods provide a good trade-off between classification accuracy and training and evaluation time.

V. CONCLUSION

Of the four multiclass SVM classification techniques considered in this work, the OAA approach gives the best overall performance as far as the classification accuracy is concerned, when trained with clean samples only. It also performs better than the other methods in high noise conditions. However, the SVM methods give similar classification accuracy with multi-conditional training. The classification accuracy for the ADAG method is dependent on the class order list and a few methods have been proposed in literature to get an optimal order. Some of these techniques were tested with clean data and while the classification accuracy increased slightly, that wasn't always the case with the addition of noise to the signal. As a result, the overall classification accuracy largely remained unchanged and is something that we plan to explore further.

TABLE III. COMPARISON OF TRAINING AND EVALUATION TIME - TRAINING USING CLEAN SAMPLES ONLY

Classification Method	MFCC Only						MFCC+ZCR+STE+SBE+SC+BW+SR					
	Training Time (s)	Testing Time (s)					Training Time (s)	Testing Time (s)				
		Clean	20dB	10dB	0dB	Average		Clean	20dB	10dB	0dB	Average
OAA-SVM	0.398	1.989	17.683	17.852	17.671	13.799	0.453	2.099	18.538	18.455	18.386	14.369
OAO-SVM	0.458	6.434	60.136	58.949	58.066	45.897	0.482	6.578	59.701	60.336	59.356	46.492
DDAG-SVM	0.458	1.318	11.883	11.872	11.730	9.201	0.482	1.339	12.208	12.127	12.081	9.439
ADAG-SVM	0.458	1.355	12.154	12.003	12.913	9.606	0.482	1.350	12.216	12.139	12.227	9.483

TABLE IV. COMPARISON OF TRAINING AND EVALUATION TIME - MULTI-CONDITIONAL TRAINING

Classification Method	MFCC Only						MFCC+ZCR+STE+SBE+SC+BW+SR					
	Training Time (s)	Testing Time (s)					Training Time (s)	Testing Time (s)				
		Clean	20dB	10dB	0dB	Average		Clean	20dB	10dB	0dB	Average
OAA-SVM	3.769	2.828	24.922	8.377	24.831	15.240	3.869	2.962	25.890	8.718	26.137	15.927
OAO-SVM	1.589	7.156	65.082	22.247	65.020	39.876	1.600	7.156	64.632	21.542	64.598	39.482
DDAG-SVM	1.589	1.462	13.265	4.489	13.187	8.101	1.600	1.488	13.305	4.385	13.391	8.142
ADAG-SVM	1.589	1.499	13.665	4.489	13.558	8.303	1.600	1.500	14.326	4.562	13.429	8.454

While multi-conditional training significantly improves the classification accuracy over training with clean samples only, the number of training samples and the training time increase as a result. This work focused on comparison of multiclass SVM classification techniques but the addition of more noise robust features could be considered in future work.

REFERENCES

[1] V. N. Vapnik, *Statistical learning theory*. New York: Wiley, 1998.

[2] L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, L. D. Jackel*, et al.*, "Comparison of classifier methods: a case study in handwritten digit recognition," in *Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 2 - Conference B: Computer Vision & Image Processing*, 1994, pp. 77-82.

[3] U. H. G. Kreßel, "Pairwise classification and support vector machines," in *Advances in Kernel Methods - Support Vector Learning*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA: MIT Press, 1999, pp. 255-268.

[4] J. C. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin DAGs for multiclass classification," in *Advances in Neural Information Processing Systems 12 (NIPS-99)*, S. A. Solla, T. K. Leen, and K.-R. Müller, Eds. Cambridge MA: MIT Press, 2000, pp. 547-553.

[5] B. Kijsirikul, N. Ussivakul, and S. Meknavin, "Adaptive directed acyclic graphs for multiclass classification," in *PRICAI 2002: Trends in Artificial Intelligence*. vol. 2417, M. Ishizuka and A. Sattar, Eds. Berlin Heidelberg: Springer, 2002, pp. 158-168.

[6] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks,* vol. 13, no. 2, pp. 415-425, 2002.

[7] N. Seo, "A comparison of multi-class support vector machine methods for face recognition," The University of Maryland, Research Report, 6 Dec. 2007.

[8] J. Weston and C. Watkins, "Multi-class support vector machines," Department of Computer Science, Royal Holloway, University of London, Egham, UK, Technical Report CSD-TR-98-04, 1998.

[9] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *Journal of Machine Learning Research,* vol. 2, pp. 265-292, 2001.

[10] G. Guo and S. Z. Li, "Content-based audio classification and retrieval by support vector machines," *IEEE Transactions on Neural Networks,* vol. 14, no. 1, pp. 209-215, 2003.

[11] G. Madzarov, D. Gjorgjevikj, and I. Chorbev, "Multi-class classification using support vector machines in binary tree architecture," in *International Scientific Conference*, Gabrovo, 2008, pp. 413-418.

[12] S. Xia, J. Li, L. Xia, and C. Ju, "Tree-structured support vector machines for multi-class classification," in *Advances in Neural Networks – ISNN 2007*. vol. 4493, D. Liu, S. Fei, Z. Hou, H. Zhang, and C. Sun, Eds. Berlin Heidelberg: Springer, 2007, pp. 392-398.

[13] C. Peng and L. Shuang, "An improved DAG-SVM for multi-class classification," in *Fifth International Conference on Natural Computation (ICNC '09)*, 2009, pp. 460-462.

[14] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning,* vol. 20, no. 3, pp. 273-297, Sep. 1995.

[15] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Transactions on Neural Networks,* vol. 10, no. 5, pp. 988-999, 1999.

[16] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, Pittsburgh, Pennsylvania, USA, 1992, pp. 144-152.

[17] G. Madzarov and D. Gjorgjevikj, "Evaluation of distance measures for multi-class classification in binary SVM decision tree," in *Artificial Intelligence and Soft Computing*. vol. 6113, L. Rutkowski, R. Scherer, R. Tadeusiewicz, L. Zadeh, and J. Zurada, Eds. Berlin Heidelberg: Springer, 2010, pp. 437-444.

[18] S. Nakamura, K. Hiyane, F. Asano, T. Nishiura, and T. Yamada, "Acoustical sound database in real environments for sound scene understanding and hands-free speech recognition," in *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC 2000)*, Athens, Greece, 2000, pp. 965–968.

[19] *BBC Sound Effects Library*. Available: http://www.leonardosoft.com

[20] A. Rabaoui, M. Davy, S. Rossignol, and N. Ellouze, "Using one-class SVMs and wavelets for audio surveillance," *IEEE Transactions on Information Forensics and Security,* vol. 3, no. 4, pp. 763-775, 2008.

[21] A. Varga and H. J. M. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Communication,* vol. 12, no. 3, pp. 247-251, Jul. 1993.

[22] E. Alexandre, L. Cuadra, M. Rosa, and F. Lopez-Ferreras, "Feature selection for sound classification in hearing aids through restricted search driven by genetic algorithms," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 15, no. 8, pp. 2249-2256, 2007.