ISSN 2442-6571


# Incremental multiclass open-set audio recognition

Hitham Jleed [a,1,*], Martin Bouchard [a,2]

[a] School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Canada
[1] h.jleed@ieee.org; [2] bouchm@uottawa.ca
* corresponding author

ARTICLE INFO

**Article history**
Received March 14, 2022
Revised April 27, 2022
Accepted May 13, 2022
Available online July 31, 2022

**Keywords**
Incremental learning
Open-Set recognition
Support vector machine
Audio recognition

ABSTRACT

Incremental learning aims to learn new classes if they emerge while maintaining the performance for previously known classes. It acquires useful information from incoming data to update the existing models. Open-set recognition, however, requires the ability to recognize examples from known classes and reject examples from new/unknown classes. There are two main challenges in this matter. First, new class discovery: the algorithm needs to not only recognize known classes but it must also detect unknown classes. Second, model extension: after the new classes are identified, the model needs to be updated. Focusing on this matter, we introduce incremental open-set multiclass support vector machine algorithms that can classify examples from seen/unseen classes, using incremental learning to increase the current model with new classes without entirely retraining the system. Comprehensive evaluations are carried out on both open set recognition and incremental learning. For open-set recognition, we adopt the openness test that examines the effectiveness of a varying number of known/unknown labels. For incremental learning, we adapt the model to detect a single novel class in each incremental phase and update the model with unknown classes. Experimental results show promising performance for the proposed methods, compared with some representative previous methods.



## 1. Introduction

Learning assumptions can be described based on the human perspective. The human auditory learning process is inherently incremental and depends on experience [1]. People can recognize certain sounds among different possible sound inputs, which can include sounds that have not been explicitly heard before. The central auditory system models sound events or groupings of sound events. Eventually, it classifies and labels sounds with prior knowledge (closed set) or labels a new kind of sound as an unknown class if never heard before (open set). The new sound is stored in the memory for future processing (incremental learning). In machine learning, incremental learning is an algorithm that cancontinuously learn new information from new sounds while maintaining its previous knowledge [2]. However, incremental learning is a notoriously difficult task and remains a challenge.

Traditional recognition systems normally rely on the hypothesis that audio classes in testing and training data are sourced from the same database and distributed evenly. Considering these assumptions, some algorithms have accomplished remarkable achievements in a variety of machine learning

applications. The algorithms become experts at recognizing what they have been provided. However, in real applications, the label set often expands as new classes occur during the test phase, in which the robustness of these methods can be drastically weakened. A few related research topics have been addressed to remedy this problem. Some researchers have used the open-set recognition (OSR) framework [3][4], where there is an option to abandon the decision-maker for the inputs that do not meet certain criteria. The classification discriminates among the known classes and identifies new class samples, but it cannot update the model with these new classes because it does not have an incremental learning procedure. An incremental learning procedure is a process that can update an existing model with new knowledge as new data is continuously found. In this work, we utilize open- set recognition based on our previous work [5], which uses the Peak-Side-Ratio (PSR) on scaled posterior probabilities. A known class produces a high PSR value, whereas an audio sample is considered as an unknown class when it produces a low PSR value. In incremental learning, we take advantage of the kernelized Support Vector Machine (SVM) [6], where support vectors are collected to optimize the separating hyperplane based on the Karush–Kuhn–Tucker (KKT) condition.

We perform extensive experiments on the DCASE [7] and Freesound [8] audio datasets and compare our approach against previous representative methods such as [9], [10] and [11]. The results indicate that our algorithm can achieve improved performance. Our approach differs as it aims to overcome two challenges, which are: (1) how to make it possible for the model to categorize examples of known classes into their corresponding classes and to identify examples of new classes, and (2) how to gradually add these classes without re-training the model when these classes have enough data. We propose two versions of incremental SVM algorithms with label incremental learning to deal with open-set recognition and new emerging labels. Both algorithms do not retrain the whole model during the incremental procedures. The first algorithm is incremental open-set multiclass SVM (IOmSVM). The algorithm reuses the classifier's old models as negatives and uses new samples to generate a new hyperplane to build a new model for the incremented class. The second method is based on SVM and the k-nearest neighbors (kNN) algorithm: IOmSVM+kNN. The algorithm selects the closest classes boundaries to define the new class hyperplane with information from the kNN algorithm.

This paper is structured as follows. Section 2 presents a literature review. Section 3 clarifies some concepts of machine learning classifiers (Section 3.1) and describes the detailed principle of our proposed methods (Section 3.2). Also, show the evaluation metrics that are utilized to assess the methods' performance (Section 3.3). In Section 4, experimental results and performance are given. A conclusion then follows in Section 5.

## 2. Related Works

In this section, we review efforts published in the literature that explicitly deal with SVM for audio recognition (2.1), open-set scenarios via SVM (2.2), and incremental learning via SVM (2.3).

### 2.1. SVM for Audio Recognition

Audio signals are surrounding us. They convey detailed information about where we are and what is going on around us. The goal of audio recognition algorithms is to distinguish audio signals from each other. The use of SVM classifiers has been popular in audio recognition because SVMs are effective in high-dimensional spaces and provide a distinct margin of separation between classes. An SVM classifier was used in [12] to detect dangerous situations by distinguishing sounds. They applied a wide-range of

audio features to recognize four classes and localize them. Likewise, another SVM classifier was applied in [13] with linear discriminate analysis (LDA) to recognize and localize a set of environmental sound events. Kumar *et al.* [14] used the SVM algorithm to detect emotion from audio signals. Huang et al. [15] utilized the SVM for audio events classification. They utilized different features to recognize seven events. Mahana and Singh [16] presented a comparison among audio classification algorithms. Using a similar method, an application of road surveillance was investigated in [17] to recognize dangerous situations, i.e., car accidents and skidding wheels.

### 2.2. Open Set SVM Recognition

A dataset can often be limited in the training stage, so there are many classes received in the testing stage which may not have been seen in the training/modeling stage. However, the posterior distribution involves consideration of all classes, which is insufficient because only the considerationof the known classes is available. Open set problems require the model to identify unknown classes in addition to the correct classification of all known classes. Open set recognition problems using SVM have received more attention when Scheirer *et al.* [9] formulated an open set problem in applications of image recognition and defined an open space as a feature space region that lies outsidethe support of the training trials. Junior *et al.* [18] proposed a Specialized Support Vector Machine (SSVM). The algorithm moves the hyperplanes in directions to stabilize open-space risk and empiricalrisk. For the audio domain, Battaglino *et al.* [19] provided algorithms for audio scene classification inthe open set. They used Support Vector Data Description (SVDD) and the minimum radius around the most positive training points instead of an hypersphere.

### 2.3. Incremental Learning via SVM

The concept of incremental learning refers to the scenario where a classifier can handle an instance with the emergence of new data that may occur at test time. A survey of incremental learning was published in [20] with some algorithms used for incremental learning including SVM, Naïve Bayes,and artificial neural networks. Another survey [21] explained the novel methods using incremental learning but for face/image recognition. Crammer *et al.* [22] published a discriminative large-marginalgorithm based on online incremental SVM algorithms to recognize multi-class categorization conducted on synthetic handwritten digits data. Likewise, Xu *et al.* [23] presented an incremental learning algorithm based on SVM conducted on a benchmark image database. The paper performeda comparative analysis between randomly independent sampling and Markov resampling in incremental learning. In addition to image recognition, incremental SVM research was conducted indifferent domains such as Human action recognition [24] and cyber risk detection [25]. Several otherresearchers reported their findings based on incremental learning, but as far as we know, none of themwas conducted on audio recognition. The challenge of implementing incremental recognition is thatwe do not know prior knowledge about all the classes that may occur. In our paradigm, we train onlythe new class data to update the model. We propose two algorithms for this task. The first algorithmuses the new class' data as positive samples and all the previous classes' data as negative samples tobuild the new model. The second method does not use all the previous classes' data, but it applies thedistillation of the classes using a kNN classifier. Our work is different from clustering used in [26][27], where the authors used a hierarchical clustering algorithm to cluster the novel class samples andtransfer the classification knowledge from supervised learning to unsupervised learning.

## 3. Method

### 3.1. Machine Learning Classifiers

Audio classification aims to discriminate between features representing different groups/classes of interest. This classification initially involves learning sounds with some pre-determined labels and tries to predict similar sounds based on the learned knowledge.

#### 3.1.1. Support Vector Machine

SVM methods have achieved great success in classification applications. The SVM is considered the most up-to-date kernel-based classification that is used for incremental learning applications [28]. In general, the SVM determines the optimal hyperplanes that divide two or more classes. Consider $M$ training data pairs $(x_i, y_i)$ for $i = 1, \dots, M$, with observations $x_i \in \mathbb{R}^L$ and their associated labels $y_i = \{1, \dots, N_k\}$, as an $N_k$ -categories classification problem.

*Binary SVM*

For binary recognition ( $N_k = 2$) , the output labels can be represented by $y_i = \pm 1$. Omitting the subscript $i$ for simplicity, an observation $x$ is classified into a certain class according to:

$$y = sgn[f(x)] \tag{1}$$

where $sgn(.)$ is the signum function, and the decision-making function $f(.)$ of an SVM kernel classifier is as in [29]:

$$f(x) = \sum_i \alpha_i y_i K(x_i, x) + b \tag{2}$$

where b is an offset and $\alpha_i$ are Lagrangian multipliers. According to the Karush–Kuhn–Tucker (KKT) optimality condition [30], the non-zero multipliers $\alpha_i \neq 0$ are known as support vectors (SVs), whose examples lie closest to the optimal hyperplane. $K(.)$ is a kernel function that turns the input into a high-dimensional feature space in a non-linear manner. The most popular kernels are the linear, polynomial, quadratic, and radial basis functions [31]. Here we apply the radial basis function (RBF) kernel:

$$K(x, x_j) = \exp(-\gamma \|x - x_i\|^2), \text{ where } \gamma > 0 \tag{3}$$

*Multi-Class SVM*

Because SVM is a binary classifier, it must be used in conjunction with other SVM classifiers to tackle multi-binary classification problems. We use the one-vs-all (OVA) structure to build these classifiers. This structure needs the number of classifiers to be the same as the number of classes $N_k$. As a new sample $x$ arrives, each classifier gives a continuous classification score $f_j(x) \in \mathbb{R}$ used to perform classification decisions according to a winner-takes-all strategy:

$$\hat{y} = \underset{j=1,\dots,N_k}{\operatorname{argmax}} f_j(x) \tag{4}$$

In the open-set scenario, the classification decision requires the system to not only assign a certain class but also predict whether it belongs to a class type. More explanations on this will be provided later in the paper.

#### 3.1.2. K-Nearest Neighbors

The simplicity of the kNN algorithm makes it a good choice as a classifier, especially when assisting other classifiers. The idea behind the nearest neighbors methods is to find a predetermined number of k neighbor samples that are closest to the new point. This depends on the choice of k and a suitable distance function. For distance functions, a variety of functions are available in the literature, but the most common function is the Euclidean distance function because it prefers a simple shape as the nearest neighbor [32]. The Euclidean distance is:

$$d(u,v)=\sqrt{\sum_{l=1}^{L}(u_l-v_l)^2} \tag{5}$$

where $u_l, v_l$ are components of data vectors $u, v \in \mathbb{R}^L$ . The classification is usually done by voting among the neighbors found in different classes.

### 3.2. Proposed Method

This section describes our methodologies applied for multi-class incremental learning. We present the basics of audio analysis preprocessing in sub-section (3.2.1), input representation and feature extraction in sub-section (3.2.2), classification with rejection in sub-section (3.2.3), and multi-class incremental learning in sub-section (3.2.4).

#### 3.2.1. Pre-processing

The preprocessing operations consist of normalization, pre-emphasis, segmentation, and silence removal. For each file, the sampling frequency is fixed as 44,100 samples/sec and files are stored in a library with the mono format (single channel). Each recording is normalized to unit maximum absolute amplitude, to increase the overall signal's strength to its maximum without clipping.

1) Voice Activity Detection (VAD)

We implemented a VAD algorithm to remove silence, where each recording is divided into small frames of 20 ms with 50% overlapping. Let $ES_{pq}$ be the actual total signal energy and $En_{\mathrm{pq}}$ be some estimated background noise energy, where $p$ denotes segment index, and $q$ denotes frequency bin. When the actual signal energy is low, the estimated noise energy is updated as follows:

$$En_{pq}=\begin{cases} En_{pq} & \text{if } ES_{pq} > 2\times En_{pq} \\ 0.94\times En_{pq}+(1-0.94)\times ES_{pq} & \text{otherwise} \end{cases} \tag{6}$$

A signal to noise ratio (SNR) for the $p^{\mathrm{th}}$ frame is defined here by averaging the SNR in dB of each frequency bin as:

$$SNR_p=\frac{1}{L}\sum_{q=0}^{L-1}10\log_{10}\frac{ES_{pq}}{En_{pq}} \tag{7}$$

where $L$ is the frequency bin number. The VAD algorithm is computed by comparing the local SNR of the $p^{\mathrm{th}}$ frame to a global SNR. The average of the local SNRs is used to calculate the global SNR for the current recording (or, in case of a continuous mode of operation, over a long observation window). If the SNR of a certain frame is less than the global SNR, this frame is measured as silence.

2) Segmentation

After silence removal, the audio clips are segmented into partially overlapped frames, using short-time processing with frames of 2048 samples and 512 samples overlap.

#### 3.2.2. Feature Extraction

Each audio signal is composed of different frequencies and different energy amplitudes, with rapid changes in a short amount of time. There is a need to define and represent audio signals such that a robust recognition system can be built. Feature extraction is the method by which useful information is extracted from the waveform signals and it generates an efficient set of feature vectors. Feature extraction is also often called front-end signal processing. A high-pass filter with a transfer function $H(z) = 1 - \alpha z^{-1}$ pre-emphasizes each frame signal $X_p(n)$ with length $N$ samples $(0 \le n \le N-1)$ for spectral slope compensation, where $\alpha = 0.97$. The segments are multiplied with a Hamming window to reduce the spectral leakage in spectrum estimation. The spectrum is computed by applying a short-time Fourier transform (STFT) analysis [33]:

$$X_p(k) = \sum_{n=0}^{N-1} x_p(n) w(n) e^{\frac{-j2\pi kn}{N}} \quad 0 \le k \le L-1 \tag{8}$$

1) Mel Frequency Cepstral Coefficients (MFCC)

MFCCs are a popular choice of features for audio classification and can be computed in four steps described in [34]. The first step estimates the short-time spectrum. The second step applies Mel-filter banks to transfer the spectrum's powers onto the Mel scale. The third step transforms the results into a logarithmic scale. Finally, A discrete cosine transform (DCT) is used to produce Mel frequency cepstral coefficients on the logarithmic filtered outputs $X_c(p)$:

$$cep_i(p) = \sqrt{\tfrac{2}{F}} \sum_{c=0}^{F-1} X_c(p) \cos\left[\tfrac{i\pi}{2F}(2c+1)\right] \tag{9}$$

where $p$ is the frame index, $cep_i$ is the $i^{th}$ coefficient computed via $F$ triangular filters. We use the first 13 MFCC coefficients.

2) Gammatone Cepstral Coefficients (GTCC)

The GTCCs are also part of the cepstral features family. The procedure for computing GTCCs is similar to MFCCs, but it uses a gammatone decomposition instead of the Mel scale. The computation can be described as [35][36]:

$$Gcep_i(p) = \sqrt{\tfrac{2}{G}} \sum_{c=0}^{G-1} G_c(p) \cos\left[\tfrac{\pi i}{2G}(2c+1)\right] \tag{10}$$

where $G$ is the number of gammatone filters, $G_c(p)$ is the signal energy in the $p^{th}$ frame, and $Gcep$ is the $m^{th}$ gammatone cepstral coefficient.

3) Spectral Flux

Spectral flux calculates the difference in spectral amplitudes between two sequential frames. For each frame $p$:

$$flux(p) = \sum_{k=0}^{N-1} \left( |X_p(k)| - |X_{p-1}(k)| \right)^2 \tag{11}$$

4) Spectral Centroid

This computes the brightness of a sound. The sound gets brighter as the centroid rises:

$$Centroid(p) = \frac{\sum_{k=0}^{N-1}(k+1)\left|X_p(k)\right|}{\sum_{k=0}^{N-1}X_p(k)} \tag{12}$$

### 3.2.3. Classification and Rejection

A model is first trained with examples from a limited number $N_k$ of classes. The training samples contain the features vectors $x_i \in \mathbb{R}^L$ with corresponding labels $y_i \in Y = \{1, \dots, N_k\}$ where $L$ is the dimension of the features. The one-versus-all multi-class SVM classification is used for this recognition, where $N_k$ classifiers are conducted to build the model. Each classifier is trained using samples from a certain class. The samples from that class are considered positive and samples from all the other $N_k$-1 classes are considered negatives. The models for the trained classes $\{M_1, M_2, \dots, M_{N_k}\}$ are stored to be used at different stages. In the testing stage, for a sample signal the Platt probabilities estimates [37] are calculated with the use of a sigmoid decision function:

$$P(Y_j = 1 / f_j(x)) = \frac{1}{1+\exp(A_j f_j(x) + B_j)} \qquad 1 \le j \le N_k \tag{13}$$

where Maximum Likelihood Estimation (MLE) is used to determine the $A_j$ and $B_j$ parameters. As a new sample (frame) $\hat{x}$ arrives in the system, the SVM classifiers compute the posterior probability, $P(Y_j|\hat{x})$, however, the probability estimation of unknown classes $P(Y_{unknown}|\hat{x})$ is not possible.

1) Peak-Side-Ratio Distribution Measure

Our proposed method uses the PSR confidence measurement introduced in [5]. It helps to determine if a new sound sample belongs to one of the predefined classes or not. The posterior probabilities values $P_j$ are arranged in descending order, where $P_1$ is the largest value and $P_{N_k}$ the smallest value. The PSR is then:

$$PSR = \left| P_1 - \frac{P_2 - P_{N_k}}{\sqrt{\frac{\sum_{c=2}^{N_k}(P_c - \bar{P})^2}{N_k - 1}}} \right| \tag{14}$$

where $\bar{P}$ is the mean of the $P_j$ values. When the PSR exceeds a specific threshold or a trigger value, the new sample is identified as a new class. If it is below the threshold, the sample is placed in the class with the highest posterior probability using the following formula:

$$\hat{y} = \begin{cases} \underset{j}{\operatorname{argmax}}\, P(Y_j \mid \hat{x}) & \text{if PSR} \le \delta \\ \text{unknown} & \text{if PSR} > \delta \end{cases} \qquad \forall j \in (1, \dots, N_k) \tag{15}$$

Where $\delta$ is a threshold determined during the validation procedure.

2) Threshold Optimization

Different threshold values in the range $\delta \in [0,4]$ were tuned to find the best possible performance, with a resolution of $\Delta\delta = 0.01$. The examination was done using five-fold stratified cross-validation. The threshold value near $\delta = 2.1$ produced the best performance based on the F1-measure, as shown in Fig.1. Therefore, this threshold was used throughout all our experiments. The decision function to recognize the $N_k$ known classes is set based on the predefined threshold.

### 3.2.4. Multi-class Incremental learning

The general diagram of multi-class incremental learning (MCIL) can be decomposed into three steps as in Fig. 2. The first step is recognizing samples from known classes and detecting the novel ones. The new sample will be saved in a buffer. The buffer's saved samples are then labeled into new categories in the second stage. The classifier then updates its model with expanded categories and samples. The challenge of multi-class incremental learning (MCIL) is how to update the model with new classes. The new class produces a new $M_{N_k+1}$ model, and an updated set of labels $y_j \in Y = \{1, \dots, N_k, N_k + 1\}$, where $j = N_k + 1$ is the new class.

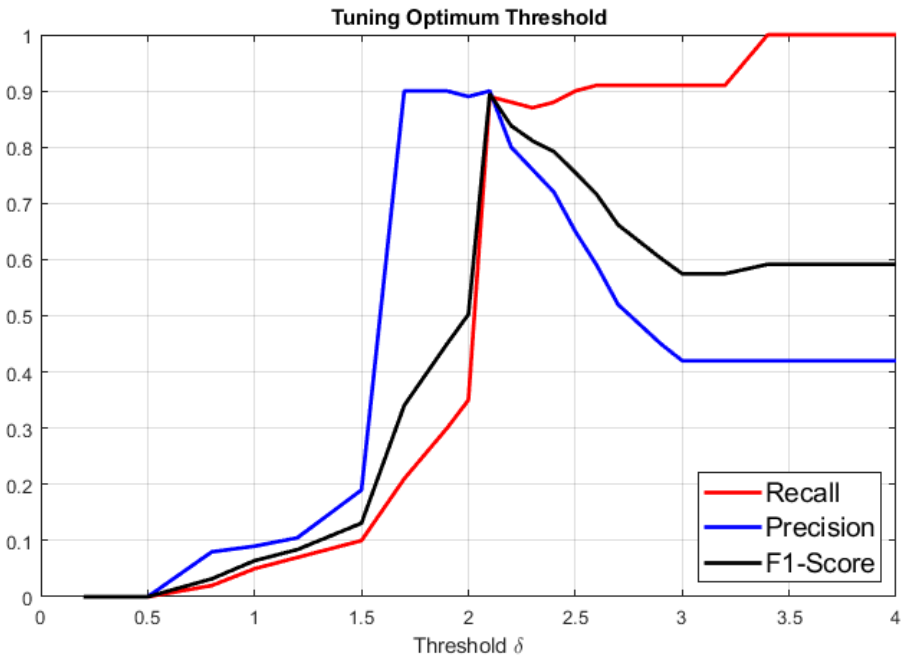


**Fig. 1.** Optimizing the threshold through a validation process.

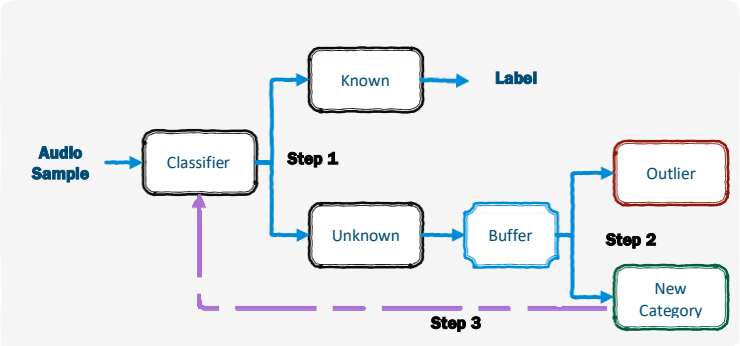


**Fig. 2.** Block-diagram of multi-class incremental learning (MCIL).

We propose two algorithms for the MCIL. The first algorithm (IOmSVM), Fig. 3, utilizes open-set recognition, which detects the unknown classes and recognizes the familiar ones. The unknown data are saved in the buffer in order to be identified. When the users label a group of data with a new class, it feeds back to the classifier for incrementation. The second method utilizes the assistance of a kNN algorithm (Fig. 4), where the system uses only the nearest classes as negative samples to shape the new hyperplane.

Req: $D=\left\langle x^{(i)},y^{(i)}\right\rangle, i=1,..,N$ samples from different classes

$C_{all} \leftarrow \{C_1, C_2, ...\}$, set of all classes

$C_{current}^{train} \leftarrow$ (randomly) pick $N_k$ classes from $C_{all}$

$D_{current}^{train} \leftarrow \left\{\left\langle x^{(i)},y^{(i)}\right\rangle \mid y^{(i)} \in C_{current}^{train}\right\}$, samples from classes

Training SVMs on $D_{current}^{train} \rightarrow \quad Models=(M_1,...,M_{N_k})$

Loop :

$C_{current}^{unknown} \leftarrow$ (randomly) pick classes from $\left\{C_{all} - C_{current}^{train}\right\}$

$D_{current}^{test} \leftarrow D_{current}^{train} \bigcup \left\{\left\langle x^{(i)},y^{(i)}\right\rangle \mid y^{(i)} \in C_{current}^{unknown}\right\}$

$Novel \leftarrow$ outlier detected via PSR-SVM tested on $D_{current}^{test}$

$\Gamma user \rightarrow$ label new class from $(Novel) \Rightarrow C_{\text{dominant}}$

$M_{new} \leftarrow SVM\left[\left\{C_{current}^{train} = negative\right\}\left\{C_{\text{dominant}} = positive\right\}\right]$

$C_{current}^{train} \leftarrow C_{current}^{train} \bigcup C_{\text{dominant}}$, increase "known" classes

$D_{current}^{train} \leftarrow D_{current}^{train} \bigcup \left\{\left\langle x^{(i)},y^{(i)}\right\rangle \mid y^{(i)} \in C_{\text{dominant}}\right\}$

$Models \leftarrow Models \bigcup M_{new}$, update the SVM model

End loop

**Fig. 3.** IOmSVM algorithm.

Req: $D=\left\langle x^{(i)},y^{(i)}\right\rangle, i=1,..,N$ samples from different classes

$C_{all} \leftarrow \{C_1, C_2, ...\}$, set of all classes

$C_{current}^{train} \leftarrow$ (randomly) pick $N_k$ classes from $C_{all}$

$D_{current}^{train} \leftarrow \left\{\left\langle x^{(i)},y^{(i)}\right\rangle \mid y^{(i)} \in C_{current}^{train}\right\}$, samples from classes

Training SVMs on $D_{current}^{train} \rightarrow \quad Models=(M_1,...,M_{N_k})$

Loop :

Neighbor classes: $K_{all}^{nearest} \leftarrow kNN\left\{C_{current}^{train}\right\}$

$C_{current}^{unknown} \leftarrow$ (randomly) pick classes from $\left\{C_{all} - C_{current}^{train}\right\}$

$D_{current}^{test} \leftarrow D_{current}^{train} \bigcup \left\{\left\langle x^{(i)},y^{(i)}\right\rangle \mid y^{(i)} \in C_{current}^{unknown}\right\}$

$Novel \leftarrow$ outlier detected via PSR-SVM tested on $D_{current}^{test}$

$\Gamma user \rightarrow$ label new class from $\Rightarrow C_{\text{dominant}}$

$C_{dk}^{nearest} \leftarrow kNN\left\{K_{all}^{nearest} \Leftrightarrow C_{\text{dominant}}\right\}$, $dk$ =k neighbors

$M_{new} \leftarrow SVM\left[\left\{C_{dk}^{nearest} = \text{negative}\right\}\left\{C_{\text{dominant}} = \text{positive}\right\}\right](Novel)$

$C_{current}^{train} \leftarrow C_{current}^{train} \bigcup C_{\text{dominant}}$, increase "known" classes

$D_{current}^{train} \leftarrow D_{current}^{train} \bigcup \left\{\left\langle x^{(i)},y^{(i)}\right\rangle \mid y^{(i)} \in C_{\text{dominant}}\right\}$

$Models \leftarrow Models \bigcup M_{new}$, update the SVM model

End loop

**Fig. 4.** IOmSVM + kNN alghoritm.

In open set recognition, the classification and rejection decisions are computed based on the probability distributions. As new classes are constantly incremented, the training data increases continually. The system optimizes the margin of the distance from previous hyperplanes and builds a new model. Fig. 3 describes the steps of this method. The second method (IOmSVM +kNN), as described in Fig. 4, does not include all the known classes features to build the new model. Instead, it determines the closest known classes according to a kNN algorithm. The method computes the hyperplane from its tangent points.

### 3.3. Evaluation Metrics

The evaluation metrics allow determining if different methods can provide a reliable classification. Here we follow experimental paradigms that are commonly used in the literature. We approach the performance evaluation with metrics that compute similarities after aligning the recognition outputs with a reference ground truth. Two fundamental assumptions have been used in the DCASE/ AASP challenges [38][7] to evaluate the sound classifications:

1. Segment-based evaluation: For each segment length, the system's output is compared against the ground truth.
2. Event-based evaluation: In this case, the system output is considered the same within the duration of the event. This means that event labels in the recognition output will be compared to the ground truth event.

#### 3.3.1. Dataset

For experimental validation, we use a dataset including audio events captured in an office-like environment. The dataset is retrieved from the DCASE 2013, 2016 [7] and Freesound [8] datasets. The datasets contain various sounds with different background noises at different levels (high, medium, and low). All the recordings used have a WAV format (e.g. uniform 16 bits quantization) to avoid transformation/coding artifacts (e.g., from MP3 or AAC encoding). Table 1 shows the number of recorded frames used for each class.

**Table 1.** Audio dataset: number of frames per class

| | DCASE2016 | DCASE2013 | Freesound | Total |
|---|---|---|---|---|
| alert | - | 1162 | 420 | 1582 |
| clearthroat | 437 | 650 | 74 | 1161 |
| cough | 596 | 670 | 284 | 1550 |
| doorslam | 277 | 1257 | - | 1534 |
| drawer | 612 | 951 | - | 1563 |
| keyboard | 1094 | 2178 | 714 | 3986 |
| keys | - | 1163 | - | 1163 |
| knock | 437 | 743 | 530 | 1710 |
| laughter | 949 | 866 | - | 1815 |
| mouse | - | 840 | 286 | 1126 |
| pageturn | 635 | 1823 | - | 2458 |
| pendrop | - | 472 | - | 472 |
| phone | 1119 | 5307 | 950 | 7376 |
| printer | - | 12099 | 810 | 12909 |
| speech | 1107 | 1709 | - | 2816 |
| switch | - | 285 | 110 | 395 |

### 3.3.2. Performance Measures

Let us consider a binary classification, where the labels consist only of positives or negatives. Based on true labels and predicted labels, we divide the metrics into four intermediate statistics: true-positives (*TP*), false-positives (*FP*), true-negatives (*TN*), and false-negatives (*FN*). A count is made for each category. Applying this to a multi-class problem, every single classifier that produces a "positive" or "negative" prediction can be "true" or "false" depending on the corresponding ground-truth label.

**Recognition Accuracy.** The recognition accuracy (*RA*) can be described as the ratio of the correctly labeled predictions to the whole pool:

$$RA = \frac{TP + TN}{TP + FP + FN + TN} \tag{16}$$

**Precision and Recall.** Precision (*Pr*) is the ratio of predicted positive samples that are computed correctly (true) divided by all predicted positive samples, while recall (*Re*) is the proportion of predicted positive samples correctly detected from all ground truth positive samples (labels). For multi-class classification, there are two ways of computation [39], macro-averaging (17) and micro-averaging (18):

$$Pr_{macro} = \frac{1}{Nk}\sum_{i=1}^{Nk}\frac{TP_i}{TP_i + FP_i} \qquad Re_{macro} = \frac{1}{Nk}\sum_{i=1}^{Nk}\frac{TP_i}{TP_i + FN_i} \tag{17}$$

$$Pr_{micro} = \frac{\sum_{i=1}^{Nk} TP_i}{\sum_{i=1}^{Nk}(TP_i + FP)_i} \qquad Re_{micro} = \frac{\sum_{i=1}^{Nk} TP_i}{\sum_{i=1}^{Nk}(TP_i + FN_i)} \tag{18}$$

**F-measure.** The F-measure introduced in [40] includes both precision and recall merged in a single score, which is computed as the harmonic mean between precision and recall. The general form of the F-measure is computed as:

$$F1\text{-}measure = \frac{(\beta^2 + 1)PrRe}{\beta^2(Pr + Re)} \tag{19}$$

where $\beta$ is a weighted parameter. When $\beta = 1$, this measure is called F1-measure and it is used in this work.

## 4. Results and Discussion

Our experiments are conducted for open set recognition and incremental learning. We conduct several evaluations in which we compare the performance of our proposed algorithms to the performance of previous representative algorithms: W-SVM [9], OSNN [10], and OSmIL [11]. The parameters of all previous algorithms were set according to the corresponding papers.

### 4.1. Audio Recognition and Rejection

The template is designed so that author affiliations are not repeated each time for multiple authors of the same affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization). This template was designed for two affiliations.

This section assesses the effectiveness of the proposed method in an open-set scenario but without incremental updates. We evaluate the performance of the system for both segment-based and event-based metrics, using standard evaluation configurations given in [9]. The systems are assessed for the three types of errors listed below:

1. Misclassification: Test samples were incorrectly labeled as being a member of one of the pre-defined classes.
2. False unknown: Test samples were predicted as unknown, but they did belong to one of the predefined classes.
3. False known: Test samples were actually unknown but labeled as one of the predefined classes.

#### 4.1.1. Protocol Setup

In the training, we need to determine two important parameters, i.e., the margin parameter $c$ and the kernel parameter $\gamma$ that are used in (3). The kernel parameter is related to the span of an RBF kernel, while the margin parameter determines the tradeoff between the complexity of the SVM and the empirical error. Therefore, we tested a wide range of combinations $c \in \{10^{-3}, 10^{-2}, \dots, 10^{4}\}$ and $\gamma \in \{2^{-6}, 2^{-5}, \dots, 2^{9}\}$, and heuristically found the highest generalization ability with parameters $c = 10^{2}$ and $\gamma = 2^{4} = 16$.

#### 4.1.2. Closed-Set Recognition

This task examines the classifier's accuracy, where the first type of error (misclassification) is the only one that can occur. This part is an essential task since it demonstrates how well the training data is learned by an algorithm. In this section, we did not make any comparison with other algorithms, since the goal of this experiment is only to test our proposed algorithm's capacity to distinguish between known classes. Fig. 5 and Fig. 6 use box plots [41] to show the experimental observations for the stratified five-fold cross-validation.

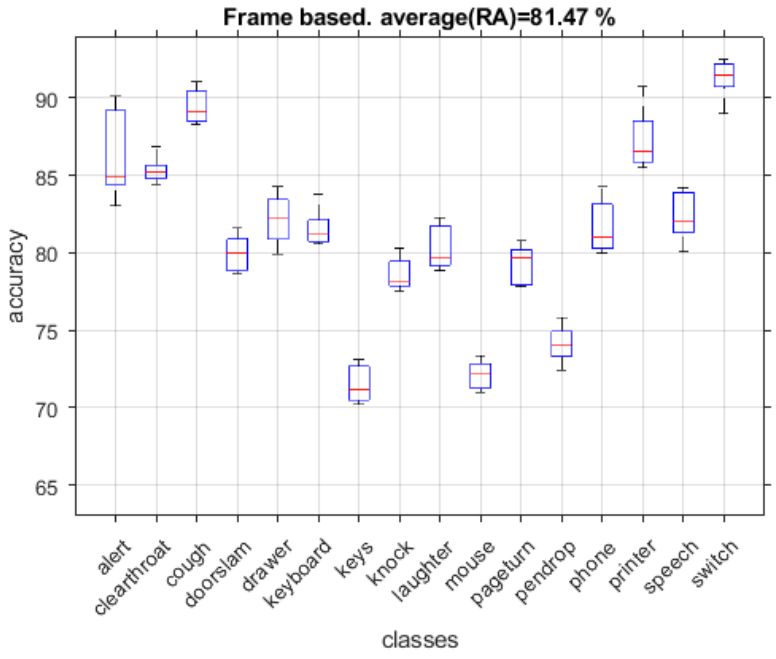


**Fig. 5.** Box-plot distributions of recognition accuracy (RA) of closed-set recognition using frame-based evaluation.

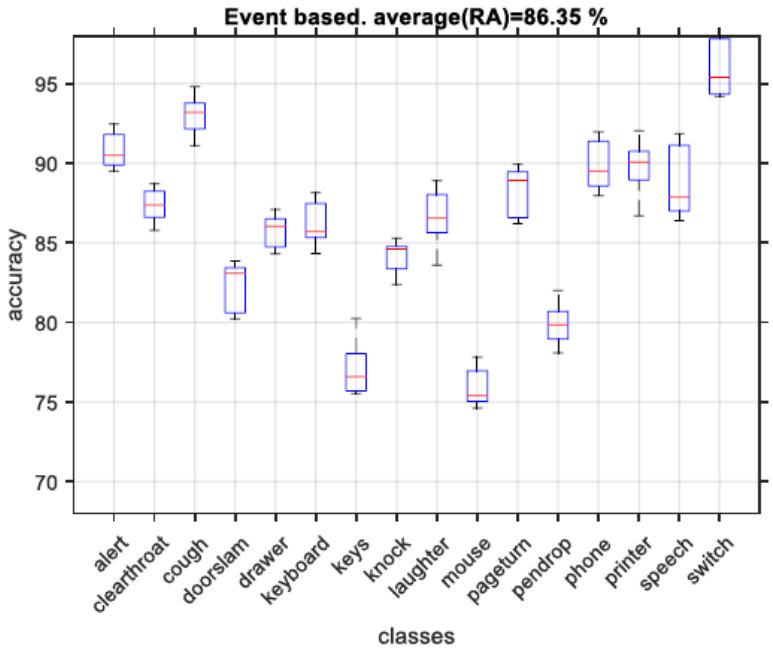


**Fig. 6.** Box-plot distributions of recognition accuracy (RA) of closed-set recognition using event-based evaluation.

The median is indicated by the red center bar in the boxes, and the interior of this box indicates the upper and lower quartiles. Outliers are points that are outside of this range. It is noticeable that the system has good classification accuracy even though some classes are not easily distinguishable because of the strong correlation among them. It can be noticed that among the 16 classes, the 'keys' and 'mouse' sounds were the most difficult classes to identify. The average event-based accuracy was 86.35%, while the average frame-based accuracy was 81.47%, which shows that event-based performance outperforms frame-based performance (as expected).

#### 4.1.3. Open-Set Recognition

The experiments in this section were performed to recognize audio sounds where the testing set does not only include the same categories as in the training dataset. These experiments measure the capability to discriminate novel classes from predefined classes and to discriminate known classes from one another. We let $Y$ represent the dataset that contains all the class labels and use the subscripts and $t$, $k$, and $u$ for the target, known, and unknown label sets, respectively. The openness of the dataset can be defined as:

$$Openness = 1 - \sqrt{\frac{2 \times |Y_t|}{|Y_k| + |Y_u|}} \tag{20}$$

We used all the known classes in the testing stage, which means that all known classes are target classes $Y_t = Y_k$. Thus, the training dataset is $Y_{train} = Y_k$, while all classes are included in the testing dataset: $Y_{test} = Y_k \cup Y_u$. The unknown classes are the complement of $Y_{train}$, i.e., $Y_u = \{y \mid y \in Y_{test} \; and \; y \notin Y_{train}\}$.

The openness has a range of 0 to 1, with 0 being a totally closed-set problem. We used varying degrees of openness and followed 5-fold cross-validation to obtain robust evaluation metrics. We performed the experiments by randomly selecting six classes to be known in the training stage. The

remaining ten classes were used as unknown data. All the algorithms were executed using the same sort of data to sustain a fair comparison. The experimental results in Fig. 7 and Fig. 8 show the performance of open-set recognition for frame-based and event-based metrics evaluations.

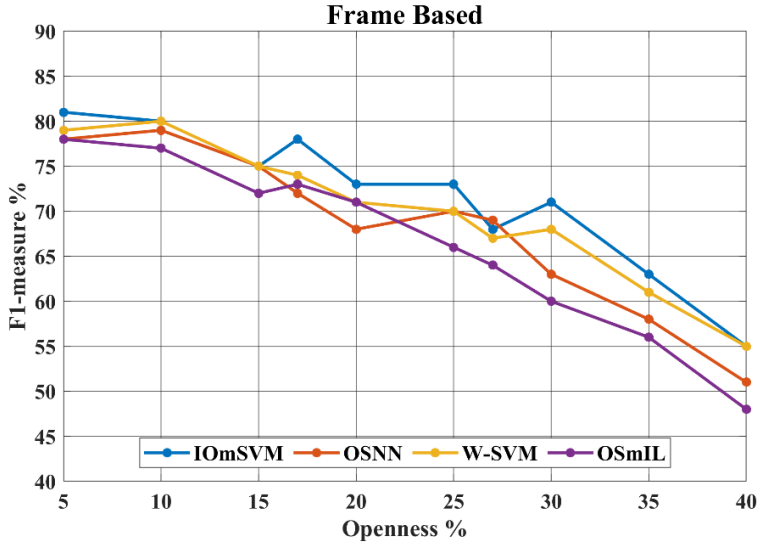


**Fig. 7.** F1-measure as a function of openness using frame-based metrics for open-set recognition.

It can be observed that in comparison to previous methods, the proposed method delivers either the best or near-best results, for a diverse set of openness values from 0% to 40% of openness. We also used 10% and 35% openness as examples to show additional metrics for the considered methods. The results in Fig. 9 were computed based on micro-average metrics, as described in (17)-(18). Because it computes each class independently and then averages them, a macro-average method computes all classes equally. Alternatively, a micro-average method aggregates the contributions of all classes to compute an average. The micro-average is preferable in this case because the audio classes do not have the same number of audio samples. It can be observed again that our proposed algorithm provides overall the best OSR performance for the considered openness values.

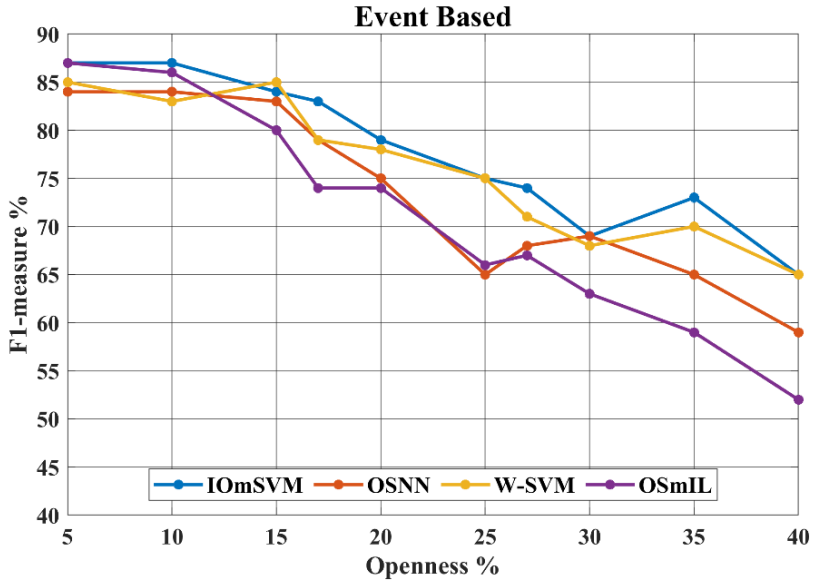


**Fig. 8.** F1-measure as a function of openness using event-based metrics for open-set recognition.

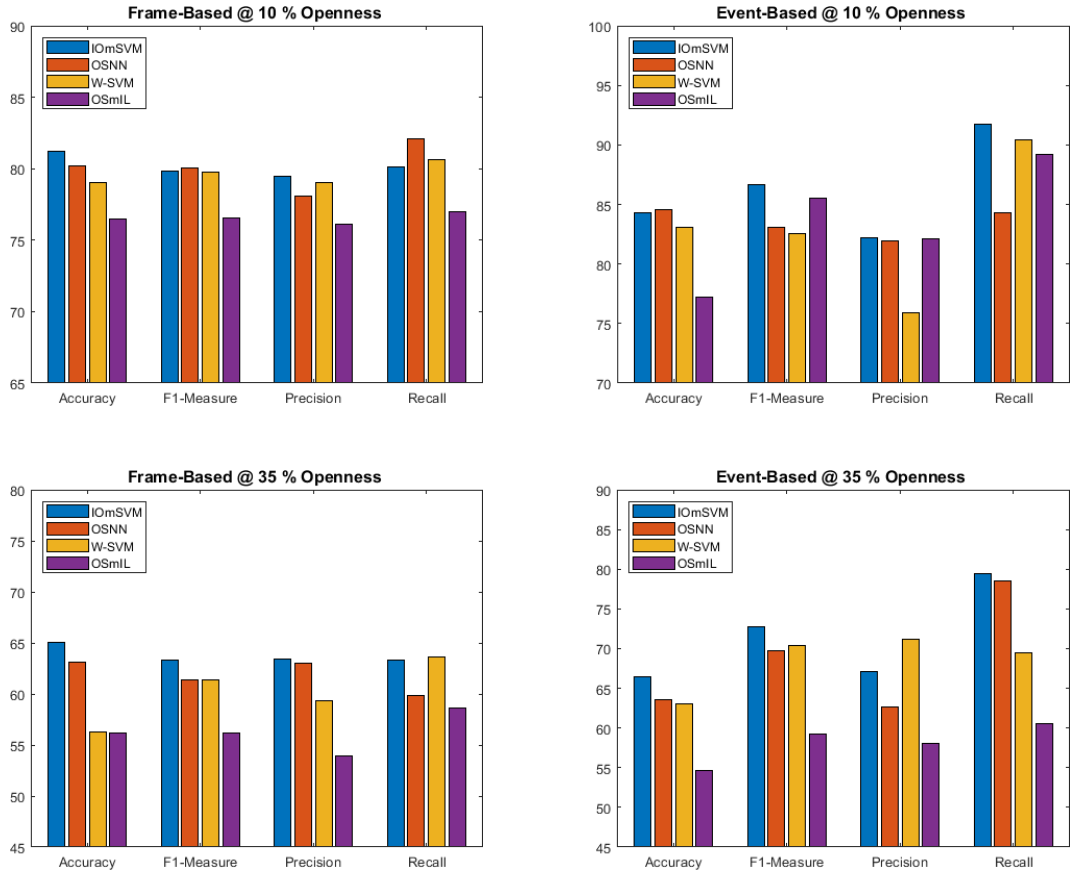


**Fig. 9.** Accuracy, precision, recall, and F1-measure at different openness values for open-set recognition.

### 3.4. Incremental Learning

For incremental learning, we train the system on a pre-defined number $N_k$ of initial classes and then incrementally add the additional classes one by one to evaluate the incremental learning performance. As we start using incremental learning, we use both of our proposed methods for these experiments, i.e., IOmSVM and IOmSVM+ kNN.

1) Training Phase

The training is divided into two parts, an initial learning phase, and an incremental learning phase. A certain number of classes are modeled at the initial phase. In our experiments, we start with four classes. Once the incremental learning phase starts, we add new classes to the system one by one.

2) Testing Phase

The system is evaluated for the incremental learning performance in both closed set and open set scenarios, where the data are split into two sets, the known set and the unknown set, to simulate closed set and open set experiments. These procedures are repeated as more classes are introduced to the system continuously.

3) Closed-set Incremental Learning Performance

We carry out this experiment to show the incremental learning performance in a closed set scenario. The closed set scenario is chosen for this experiment because misclassification may occur in the calibration process of open set rejection. The results in Fig. 10 and Fig. 11 show that the performance

of our proposed methods is promising, in particular, the IOmSVM+ kNN algorithm provides the best performance overall. As expected, the performance of event-based measures outperforms that of frame-based measurements.

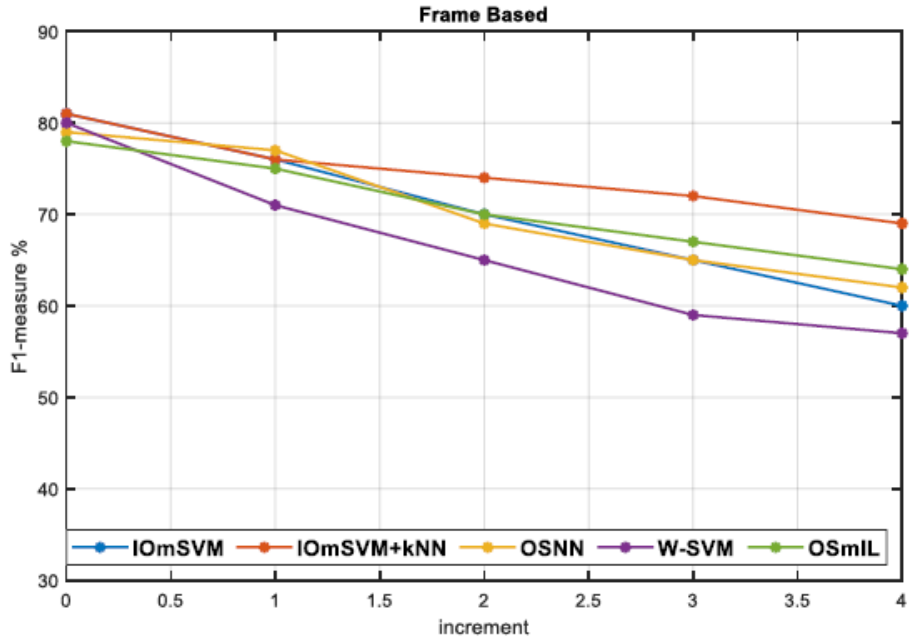


**Fig. 10.** F1-measure for the incremental learning process in closed set scenario using frame-based metrics.

4) Multiclass Open-set Incremental Learning Performance

This part of incremental learning consists of two processes: open set recognition and incremental learning. It is a more realistic incremental scenario. In Fig. 12 and Fig. 13, the axis on the right side illustrates the incremented classes. Considering the F1-measure as the evaluation metric, the more incremented classes are added, the poorer the performance of the classifier is expected to become.

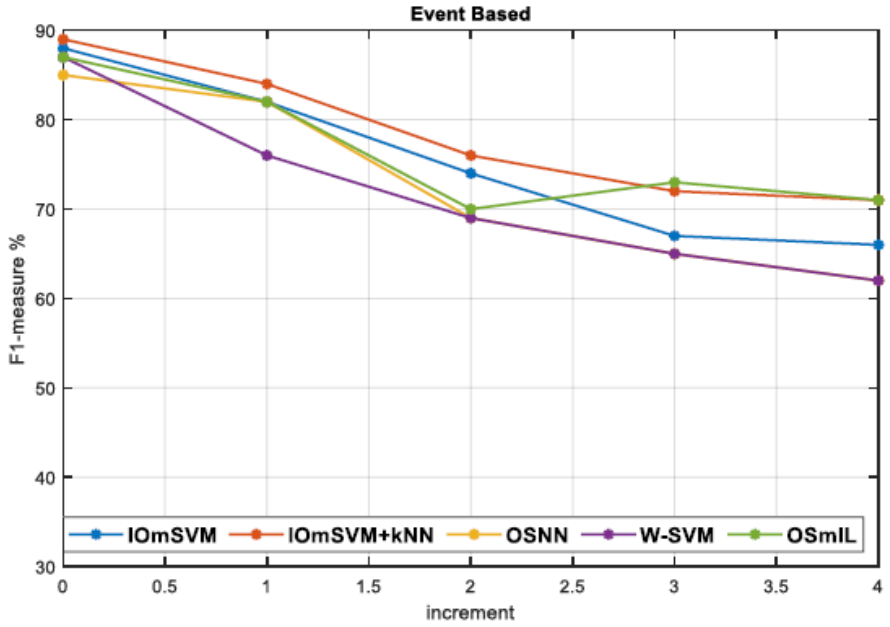


**Fig. 11.** F1-measure for the incremental learning process in a closed set scenario using event-based metrics.

The left side axis depicts the different degrees of openness considered in the simulations. When the openness equals zero, it becomes a closed set incrementing as in a previous sub-section. For zero increment, it becomes open set recognition without incrementing. It is clear from Fig. 12 and Fig. 13 that our proposed methods (especially IOmSVM+ kNN) can maintain a very competitive (typically better) performance in terms of recognizing previously trained classes and identifying new unknown classes, for different openness values and different numbers of incremented classes.

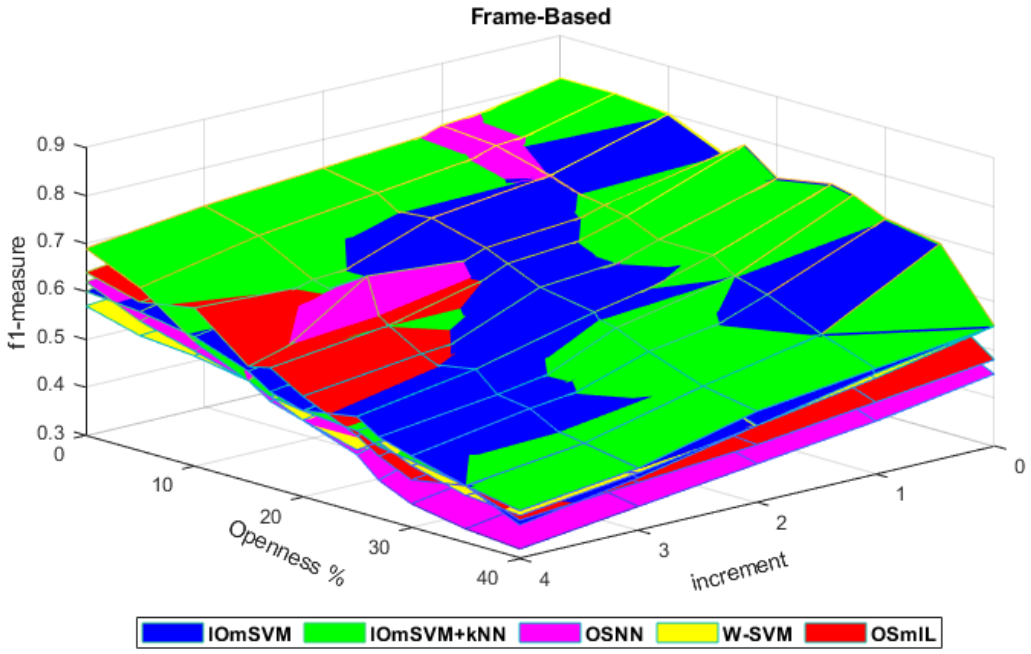


**Fig. 12.** Multi-class open-set recognition with incremental learning, frame-based evaluation.

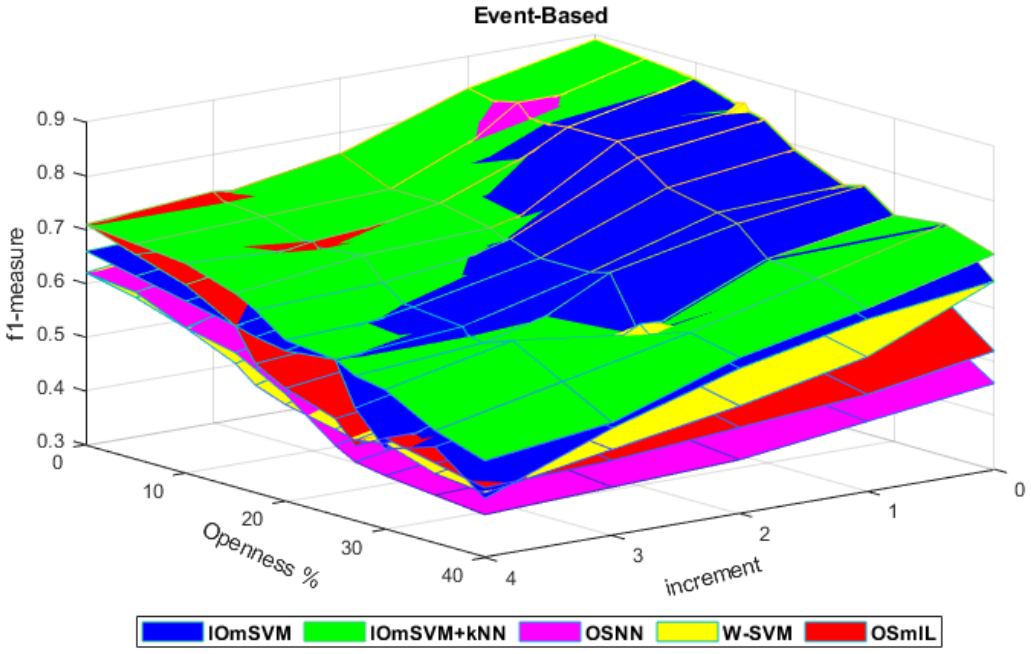


**Fig. 13.** Multi-class open-set recognition with incremental learning, event-based evaluation

## 5. Conclusion

In this work, we proposed solutions for multiclass open-set recognition and incremental learning for audio recognition, which is also known as an open-world scenario. The open-world scenario has two challenging aspects: novel classes may continuously occur, and they should be updated in the system using a multi-class open set recognition algorithm. Our proposed algorithms (IOmSVM and IOmSVM+kNN) can perform incremental open set modeling by updating existing classes' decision-making boundaries and establishing new decision boundaries for new classes. Extensive experiments have been carried out to verify the efficacy of the proposed algorithms for audio recognition. The results reveal that our proposed algorithms are promising approaches for both open set recognition and incremental learning in audio recognition. There are two types of limitations in this work. The first limitation concerns the choice of features. Audio features are still a crucial choice that affects performance and more investigation needs to be performed to evaluate the impact of different sets of features. The second limitation is that the proposed methods cannot handle well the extreme openness scenarios, i.e., when the number of unknown classes is high, the recognition performance decreases significantly. Future work will be required to investigate incremental learning when the unknown classes exist in both the training and the testing database.

## Acknowledgment

This work was supported by the Libyan North America Scholarship Program, and in part by a Discovery Grant from the Natural Sciences and Engineering Research Council (NSERC) of Canada.

## Declarations

**Author contribution.** All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

**Funding statement.** None of the authors have received any funding or grants from any institution or funding body for the research.

**Conflict of interest.** The authors declare no conflict of interest.

**Additional information.** No additional information is available for this paper.

## References

[1] D. R. F. Irvine, "Auditory perceptual learning and changes in the conceptualization of auditory cortex," *Hear. Res.*, vol. 366, pp. 3–16, Sep. 2018, doi: 10.1016/j.heares.2018.03.011.

[2] Y. Yang et al., "Learning Adaptive Embedding Considering Incremental Class," *IEEE Trans. Knowl. Data Eng.,* pp. 1–1, 2021, doi: 10.1109/TKDE.2021.3109131.

[3] C. Geng and S. Chen, "Collective Decision for Open Set Recognition," *IEEE Trans. Knowl. Data Eng.,* vol. 34, no. 1, pp. 192–204, Jan. 2022, doi: 10.1109/TKDE.2020.2978199.

[4] L. P. Jain, W. J. Scheirer, and T. E. Boult, "Multi-class open set recognition using probability of inclusion," in *Computer Vision – ECCV 2014,* vol. 8691, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 393–409. doi: 10.1007/978-3-319- 10578-9_26.

[5] H. Jleed and M. Bouchard, "Open set audio recognition for multi-class classification with rejection," *IEEE Access*, vol. 8, pp. 146523–146534, 2020, doi: 10.1109/ACCESS.2020.3015227.

[6] Y. Guo, Z. Zhang, and F. Tang, "Feature selection with kernelized multi-class support vector machine," *Pattern Recognit.,* vol. 117, p. 107988, Sep. 2021, doi: 10.1016/j.patcog.2021.107988.

[7] A. Diment, T. Heittola, and T. Virtanen, "Sound event detection for office live and office synthetic AASP challenge," *Proc IEEE AASP Chall. Detect. Classif Acoust Scenes Events WASPAA*, 2013, Accessed: Nov. 11, 2016. [Online]. Available: http://c4dm.eecs.qmul.ac.uk/sceneseventschallenge/abstracts/OL/DHV.pdf

[8] F. Font, G. Roma, and X. Serra, "Freesound technical demo," *in Proceedings of the 21st ACM international conference on Multimedia,* 2013, pp. 411–412, doi: 10.1145/2502081.2502245.

[9] W. J. Scheirer, L. P. Jain, and T. E. Boult, "Probability models for open set recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2317–2324, Nov. 2014, doi: 10.1109/TPAMI.2014.2321392.

[10] P. R. Mendes Júnior et al., "Nearest neighbors distance ratio open-set classifier," *Mach. Learn.*, vol. 106, no. 3, pp. 359–386, Mar. 2017, doi: 10.1007/s10994-016-5610-8.

[11] S. Dang, Z. Cao, Z. Cui, Y. Pi, and N. Liu, "Open set incremental learning for automatic target recognition," *IEEE Trans. Geosci. Remote Sens.,* vol. 57, no. 7, pp. 4445–4456, Jul. 2019, doi: 10.1109/TGRS.2019.2891266.

[12] K. Łopatka, J. Kotus, and A. Czyżewski, "Evaluation of sound event detection, classification and localization in the presence of background noise for acoustic surveillance of hazardous situations," in *Multimedia Communications, Services and Security*, A. Dziech and A. Czyżewski, Eds. Springer International Publishing, 2014, pp. 96–110. doi: 10.1007/978-3-319-07569-3_8.

[13] A. R. Hilal, A. Sayedelahl, A. Tabibiazar, M. S. Kamel, and O. A. Basir, "A distributed sensor management for large-scale IoT indoor acoustic surveillance," *Future Gener. Comput. Syst.*, vol. 86, pp. 1170–1184, 2018, doi: 10.1016/j.future.2018.01.020.

[14] R. Kumar and A. Punhani, "Emotion Detection from Audio Using SVM," in *Proceedings of International Conference on Big Data, Machine Learning and their Applications*, Singapore, 2021, pp. 257–265. doi: 10.1007/978-981-15-8377-3_22.

[15] W. Huang, S. Lau, T. Tan, L. Li, and L. Wyse, "Audio events classification using hierarchical structure," in *Proceedings of the 2003 Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing, 2003 and Fourth Pacific Rim Conference on Multimedia*, Dec. 2003, vol. 3, pp. 1299–1303 vol.3. doi: 10.1109/ICICS.2003.1292674.

[16] P. Mahana and G. Singh, "Comparative analysis of machine learning algorithms for audio signals classification," *Int. J. Comput. Sci. Netw. Secur. IJCSNS*, vol. 15, no. 6, p. 49, 2015, available at: Google Scholar.

[17] P. Foggia, N. Petkov, A. Saggese, N. Strisciuglio, and M. Vento, "Audio surveillance of roads: a system for detecting anomalous sounds," *IEEE Trans. Intell. Transp. Syst.,* vol. 17, no. 1, pp. 279–288, Jan. 2016, doi: 10.1109/TITS.2015.2470216.

[18] P. R. M. Júnior, T. E. Boult, J. Wainer, and A. Rocha, "Specialized support vector machines for open- set recognition," *ArXiv160603802 Cs Stat*, Jun. 2016, Accessed: Dec. 04, 2018. [Online], available: http://arxiv.org/abs/1606.03802.

[19] D. Battaglino, L. Lepauloux, and N. Evans, "The open-set problem in acoustic scene classification," in *2016 IEEE International Workshop on Acoustic Signal Enhancement (IWAENC)*, Sep. 2016, pp. 1–5, doi: 10.1109/IWAENC.2016.7602939.

[20] Q. Yang, Y. Gu, and D. Wu, "Survey of incremental learning," in *2019 Chinese Control And Decision Conference (CCDC),* Jun. 2019, pp. 399–404, doi: 10.1109/CCDC.2019.8832774.

[21] S. Madhavan and N. Kumar, "Incremental methods in face recognition: a survey," *Artif. Intell. Rev.,* Aug. 2019, doi: 10.1007/s10462-019-09734-3.

[22] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *J. Mach. Learn. Res.,* vol. 7, no. Mar, pp. 551–585, 2006, available at: Google Scholar.

[23] J. Xu, C. Xu, B. Zou, Y. Y. Tang, J. Peng, and X. You, "New incremental learning algorithm with support vector machines," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 49, no. 11, pp. 2230–2241, Nov. 2019, doi: 10.1109/TSMC.2018.2791511.

[24] M. Gutoski, A. E. Lazzaretti, and H. S. Lopes, "Incremental human action recognition with dual memory," *Image Vis. Comput.,* vol. 116, p. 104313, Dec. 2021, doi: 10.1016/j.imavis.2021.104313.

[25] M. Rahouti, M. Ayyash, S. K. Jagatheesaperumal, and D. Oliveira, "Incremental Learning Implementations and Vision for Cyber Risk Detection in IoT," *IEEE Internet Things Mag.*, vol. 4, no. 3, pp. 114–119, Sep. 2021, doi: 10.1109/IOTM.0011.2100019.

[26] L. Shu, H. Xu, and B. Liu, "Unseen class discovery in open-world classification," *ArXiv E-Prints,* vol. 1801, p. arXiv:1801.05609, Jan. 2018, doi: 10.48550/arXiv.1801.05609.

[27] J. Leo and J. Kalita, "Moving towards open set incremental learning: readily discovering new authors," *ArXiv191012944 Cs Stat*, Oct. 2019, Accessed: Aug. 23, 2020. [Online], Available: http://arxiv.org/abs/1910.12944.

[28] T. Diethe and M. Girolami, "Online Learning with (Multiple) Kernels: A Review," *Neural Comput.*, vol. 25, no. 3, pp. 567–625, Mar. 2013, doi: 10.1162/NECO_a_00406.

[29] S. Abe, Support vector machines for pattern classification, 2nd ed. London ; New York: Springer, 2010, Available at: Google Scholar.

[30] R. Fletcher, Practical methods of optimization. John Wiley & Sons, 2013, Available at: Google Scholar.

[31] K. S. Sahoo et al., "An Evolutionary SVM Model for DDOS Attack Detection in Software Defined Networks," *IEEE Access*, vol. 8, pp. 132502–132513, 2020, doi: 10.1109/ACCESS.2020.3009733.

[32] G. Lin, A. Lin, and J. Cao, "Multidimensional KNN algorithm based on EEMD and complexity measures in financial time series forecasting," *Expert Syst. Appl.,* vol. 168, p. 114443, Apr. 2021, doi: 10.1016/j.eswa.2020.114443.

[33] J. Xin and Y. Qi, Mathematical Modeling and Signal Processing in Speech and Hearing Sciences, vol. 10. Cham: Springer Science & Business Media, 2014, Available at: Google Scholar.

[34] B. McFee et al., "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, 2015, vol. 8, Available at: Google Scholar.

[35] J.-M. Liu et al., "Cough signal recognition with Gammatone Cepstral Coefficients," in *2013 IEEE China Summit and International Conference on Signal and Information Processing, Jul. 2013*, pp. 160-164, doi: 10.1109/ChinaSIP.2013.6625319.

[36] S. Salman, J. Mir, M. T. Farooq, A. N. Malik, and R. Haleemdeen, "Machine Learning Inspired Efficient Audio Drone Detection using Acoustic Features," in 2021 International Bhurban Conference on Applied *Sciences and Technologies (IBCAST)*, Jan. 2021, pp. 335–339, doi: 10.1109/IBCAST51254.2021.9393232.

[37] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Adv. Large Margin Classif.*, vol. 10, no. 3, pp. 61–74, 1999, Available at: Google Scholar.

[38] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for Polyphonic Sound Event Detection," *Appl. Sci.*, vol. 6, no. 6, Art. no. 6, Jun. 2016, doi: 10.3390/app6060162.

[39] K. Zhang, H. Su, and Y. Dou, "Beyond AP: a new evaluation index for multiclass classification task accuracy," *Appl. Intell.*, vol. 51, no. 10, pp. 7166–7176, Oct. 2021, doi: 10.1007/s10489-021-02223-7.

[40] C. J. Van Rijsbergen, *The geometry of information retrieval*. Cambridge, UK ; Cambridge University Press, 2004, Available at: Google Scholar.

[41] G. Wohlfahrt, E. Tomelleri, and A. Hammerle, "The urban imprint on plant phenology," *Nat. Ecol. Evol.*, vol. 3, no. 12, Art. no. 12, Dec. 2019, doi: 10.1038/s41559-019-1017-9.