Neural Computing and Applications (2023) 35:3–20
https://doi.org/10.1007/s00521-021-05749-6

S.I. : INFORMATION, INTELLIGENCE, SYSTEMS AND APPLICATIONS

Classification of acoustical signals by combining active learning strategies with semi-supervised learning schemes

Stamatis Karlos[1] · Christos Aridas[2] · Vasileios G. Kanas[3] · Sotiris Kotsiantis[4]

Received: 28 January 2020 / Accepted: 16 January 2021 / Published online: 5 February 2021

Abstract

In real-world cases, handling both labeled and unlabeled data has raised the interest of several Data Scientists and Machine Learning engineers, leading to several demonstrations that apply data-augmenting approaches in order to obtain a robust and, at the same time, accurate enough learning behavior. The main reason is the existence of much unlabeled data that are ignored by conventional supervised approaches, reducing the chance of enriching the final formatted hypothesis. However, the majority of the proposed methods that operate using both kinds of these data are oriented toward exploiting only one category of these algorithms, without combining their strategies. Since the most popular of them regarding the classification task are Active and Semi-supervised Learning approaches, we aim to design a framework that combines both of them trying to fuse their advantages during the main core of the learning process. Thus, we conduct an empirical evaluation of such a combinatory approach over three problems, which stem from various fields but are all tackled through the use of acoustical signals, operating under the pool-based scenario: gender identification, emotion detection and automatic speaker recognition. Into the proposed combinatory framework, which operates under training sets with small cardinality, our results prove the benefits of adopting such kind of semi-automated approaches regarding both the achieved predictive correctness when reduced consumption of resources takes place, as well as the smoothness of the learning convergence. Several learners have been examined for reaching to more general conclusions, and a variant of self-training scheme has been also examined.

Keywords Combined learning framework · Self-training scheme · Active learning queries · Acoustical signal classification · Data augmentation techniques · Semi-automated approaches

1 Introduction

The generic purpose of Machine Learning (ML) algorithms is to inject intelligence so as to mimic human behavior inside related learning frameworks based on data-driven tools. However, their automated operation may suffer from a series of phenomena that occur at large-scale ecosystems. We distinguish here just two of them: the unstable character of underlying conditions or the evolvement of time-based facts—known as concept drift [1]—as well as the inability to tackle Big Data problems under tight time constraints [2]. This kind of implications has induced a new situation in the field of ML research: instead of trying to collect vast amounts of instances, whose assignment of their target variable—either numeric or categorical—is usually difficult to be mined through an automated process, adoption of techniques that are based on small portions of

✉ Stamatis Karlos
stkarlos@upatras.gr

Christos Aridas
char@upatras.gr

Vasileios G. Kanas
vaskanas@gmail.com

Sotiris Kotsiantis
sotos@math.upatras.gr

[1] Department of Mathematics, University of Patras, 26500 Patras, Greece

[2] Computational Intelligence Laboratory, Department of Mathematics, University of Patras, 26500 Patras, Greece

[3] Department of Electrical and Computer Engineering, University of Patras, 26500 Patras, Greece

[4] Educational Software Development Laboratory, Department of Mathematics, University of Patras, 26500 Patras, Greece

labeled data exploiting collected or provided unlabeled data so as to refine more accurate predictive models has been widely applied the last years in several real-life scenarios. Thus, the effort needed to be spent by human experts or other sources of knowledge (e.g., volunteers or users of some domain applications) is drastically reduced [3], assuring probably a high enough quality of these initially collected training data. At the same time, it offers the chance even to more time-expensive learning models to be applied without inducing extreme time delays, exploiting potentially their discriminative ability.

As a consequence, a large family of approaches has been devised, whose main learning core relies on augmenting the cardinality of the existing instances iteratively with the most appropriate non-annotated instances. The term "appropriate" is usually measured through a suitable ranking metric that expresses the kind of information that need to characterize the newly mined instances. An in-depth review of such works has been published by Schwenker and Trentin [4] categorizing this kind of approaches as Partially Supervised Learning (PSL) techniques, covering all great applications of Machine Learning: (i) Classification, (ii) Regression, (iii) Clustering and (iv) Feature Selection. Several other works summarize recent achievements of these approaches, providing appropriate taxonomies and commenting on the main mechanisms that are applied [5–7].

For the rest of this work, the case of classification task under the pool-based scenario will be considered as the main concept of the described mechanisms. Into this context, our proposed framework deals with the trade-off of achieving accurate classification performance without spending much human effort. Thus, the term learner coincides with the meaning of classifier and the target variable is either in categorical format or in a discretized numerical one. Pool-based scenario is the most popular format of PSL techniques, where all the collected data, annotated or not, are available before the learning process begins. On contrast, during Online or Sequential learning cases, the corresponding instances arrive at specific time slots and a decision has to be drawn instantly about their utility or not. During the former, the assumption about the origin of the data is that they are obtained by an independent identically distributed (i.i.d.) sampling process from an unknown data generation function ($H(x, \omega)$), such that $H : X \rightarrow \Omega$, where $X \in R^f$ is the feature space, f is the cardinality of different features that describe each instance $x_i \in X$, defining also the initial dimensionality of the problem, and $\Omega = \{\omega_1, \omega_2, \ldots, \omega_{cl}\}$ is the class space, where $cl \geq 2$ is the number of the classes, while equality holds for binary problems. Our ambition is to exploit different PSL strategies under the existence of a few labeled data so as to format iteratively a hypothesis h, whose behavior is similar enough with this of H, which actually contains the perfect matching between instances and classes.

Active Learning (AL) category of algorithms consists of approaches that provide a semi-automated solution, since it blends the predictive power of both the human factor and the products of ML. To be more specific, starting with a small labeled subset (L) of the total collected data (D), a selected base learner is trained on L and is then applied on the rest of the data—called as unlabeled data (U)—so as to rank them according to an informativeness criterion. The above process varies based on the structure of the base learner. For example, geometry plays a crucial role in boosting the performance of Support Vector Machines (SVM) learner in [8], while the Expected Loss Optimization (ELO) [9] is more generalizable. A more generic approach is to integrate a number of learners under the concept of Query-By-Committee (QBC), where appropriate decisions are drawn based on the disagreement of the QBC participants over the U pool [10].

After the detection of the highest ranked instances ($x^{\text{usefulness}}$), human factor or human oracle ($O^{\text{human}} : x_i \rightarrow \omega^*$, where ω^* depicts the true class label) is responsible for annotating them based on its knowledge or its expertise. This role could be addressed either by human experts, regarding mainly scientific fields that demand specialized theoretical or technical background, or even by larger amount of human entities, a case that is noted as crowdsourcing [11]. This latter case is usually met in recommendation engines, where the opinion of each individual is requested and is evaluated under favoring metrics (e.g., popularity, co-coverage) [12], while some recent works study the effect of adopting less powerful human oracles, or even combining weak and strong human oracles [13]. Then, the decisions exported by human factor are arguably accepted as correct and the currently available L subset is enriched with the newly labeled data:

$$L' = L \cup \left\{x^{\text{usefulness}}, O^{\text{human}}\left(x^{\text{usefulness}}\right)\right\}, \tag{1}$$

On the other hand, the semi-supervised learning (SSL) category does not exploit at all the human factor but is solely based on the decisions produced by the corresponding selected base learner, trusting the most confident of them (x^{MCP}), where MCP stands for the Most Confident Predictions [14, 15]. Hence, instead of measuring another one quantity that transforms the output of the base learner into a convenient form, as in case of AL, its predictions are manipulated as an adequate indication for mining the U pool so as to augment appropriately the corresponding L subset:

$$L' = L \cup \left\{x^{\text{MCP}}, h\left(x^{\text{MCP}}\right)\right\}, \tag{2}$$

Despite the fact that this variant of PSL approaches leads to highly self-confident algorithms, their learning behaviors have been proved really successful in practice [16, 17]. The two basic strategies usually met in order to reduce this inherent property of SSL algorithms that are either to introduce specific performance thresholds or preprocessing procedures during the mining of confident unlabeled data [18, 19] or to employ robust base learners into the main learning kernel [14].

Consequently, judging by the manner that these two separate PSL categories operate, a hybrid framework could be devised in order to compromise both of them effectively and efficiently. The first term depicts the quality of obtained predictive performance regarding one or more performance metrics, while the second one refers to the consumption of sources, which could be translated into the reduction of the number of queries that facilitate the interaction between AL algorithms and the O^{human}. Of course, the ambition here should be the relaxation of the human factor dependency, since this involvement more usually than not induces additional expenses and time delays, without sacrificing at the same time much of the predictive ability of the finally constructed learners [20]. Prioritizing according to this trade-off, a framework that combines AL process along with the Self-training algorithm—a well-known variant of SSL algorithms [15, 21]—is described in this work, letting them to act interchangeably during the iterative proposed process.

The rest of this work is organized as follows: in Sect. 2, related works are reported briefly, including both pioneering approaches of combining AL + SSL approaches, and some of the most recently demonstrated. Section 3 contains the description of the proposed framework, while the next section gives some information about the formulation of the examined datasets, and the problem that is described by them. Finally, our results along with some statistical comparisons and comprehensive comments are given in Sect. 5, before we sum up in the last section, where potential improvements are mentioned.

2 Related Works

In order to boost the performance of the AL + SSL produced variants, the complementary behavior of AL and SSL approaches should be maximized, capturing as much as possible the underlying distribution of $H(x, \omega)$ without seeking for redundant information. This behavior could be achieved by tuning appropriately each participant of this combination so as to explore as good as possible different underlying structures. Thus, the selected Query Selection Strategy inside AL approaches is the Uncertainty Sampling Strategy (UncS), which tries to detect the most ambiguous instances based on the current hypothesis, while Self-training approach usually annotates unlabeled instances that come from dense regions, according to the well-known cluster assumption. As Settles supports [6], algorithms like Self-training extrapolate their predictions based on a latent structure over which they are more confident about their estimations. To prevent also Self-training from inserting noisy labels, a variant of this scheme has been examined, which considers the performance of the current hypothesis on a validation set before deciding about the augmentation of the current L subset or not. Moreover, the efficacy of the proposed combinatory framework is tested specifically on raw data that concern acoustical signals [22, 23]. Since this kind of data are easily interpretable and conceivable by human factor, no restrictions are posed regarding the comprehension by the latter, facilitating thus the adoption of our implementation into practical cases.

Since both AL and SSL are iterative procedures aiming to reduce the burden of manual labeling, either by finding the most informative sample in each iteration for human labeling (AL), or by exploiting the machine itself to label samples, various approaches have been recorded in the literature. McCallum and Nigam [24] were the first who noticed the complementarities between AL and SSL. In their work, they combined committee-based AL with EM-based SSL for text classification. Later, co-testing was proposed in [25], a variant of QBC method. In this method, two different views of features were used to train two classifiers separately. Then the unlabeled instances in which the classifier disagreed the most were selected for human annotation. Finally, co-testing and co-training were combined using an expectation maximization (co-EM) algorithm to automatically label instances that showed a low disagreement between the two classifiers. Their resulting combination, named as Co-EMT, was highly preferred against its ancestors because of its great robustness.

Later, in 2006, Zhou et al. [26] applied a similar combination in the field of Content-Based Image Retrieval (CBIR), where a disagreement-based approach was operating on the side of SSL and Random Sampling Strategy (RndS) process has been exploited initially for acquiring a small number of images before users were asked about their content (relevance feedback). Then, two different learners are trained on the available training data and the most confident instances are given to the other one, trying to inject diversity into the learning process by mutual-teaching. However, when RndS was replaced by a more sophisticated method—considering the instances for which opposite predictions are exported with similar high confidence by the two different learners or those instances for which the corresponding confidence metrics are too small—offering the chance to human entities to actively select the most informative images, leading to better generalization behavior [27, 28].

In another study [29], authors proposed a unified framework using the global entropy reduction maximization criterion for speech recognition. The authors in [30] studied cross-lingual sentiment classification. They proposed a new model based on the initial training data from the source language and the translated unlabeled data from the target language. The initial *L* subset is used to train a base classifier, which consequently is applied to the translated *U* pool. Then AL selects the most representative examples to be labeled by a human expert. During this labeling process, the human expert evaluates the overall sentiment polarity. Simultaneously, Self-training scheme selects some of the most confident classified examples with the corresponding predicted labels, which are added to the training set for the next learning cycle. In the next cycle, the model is retrained based on the augmented training data and this process is repeated until a termination condition is satisfied.

More recently, sound classification was studied [31]. In this work, the proposed method, applied on pool-based and stream-based processing scenarios, pre-processes the unlabeled instances by calculating their confidence scores based on a classifier performance, and then the candidates with lower scores are delivered to human annotators, while those with high scores are automatically labeled by the machine. A large database of environmental sounds was collected there (about 15 h of raw-data) where numerous features were created so as to capture numerous views of the same sound instance. Social networks have also been a field of interest for this kind of approaches since self-training alongside active learning has been utilized for named entity recognition on Twitter [32]. More specifically, uncertainty-based, and diversity-based sampling methods, used as AL query strategies, were applied to the unlabeled data to select the most informative instances, which consequently were labeled by an expert. In addition, the non-informative instances were fed to a conditional random field model and the high confident classified instances were selected. After this process both the manually labeled and machine-labeled instances were added to the training data to retrain the classification model.

Furthermore, Semi-supervised Active learning has been used for support vector machines, aiming to exploit the underlying structure information given by the spatial pattern of the (un)labeled data in the feature space [33]. Probabilistic models were used to capture the data structure. These models were iteratively improved at run time with newly available labeled data during the AL process. The probabilistic models were considered in a selection strategy based on distance, density, diversity, and distribution information for AL (4DS strategy) and in a particular kernel function for SVM (Responsibility Weighted Mahalanobis kernel) [34].

The main approaches of Semi-supervised techniques could be summarized in categories of Generative models, Single-view methods, Multi-view Learning, Semi-supervised Support Vector Machines (S3VMs) and Graph-based Models [35, 36]. Each one of these families of SSL algorithms has its assets and defects, but without Single-view methods are the less restrictive, since they operate like wrappers, proposing learning schemes that exploit one or more base learners for assigning pseudo-labels to the corresponding *U* pool. Multi-view methods also contain several kinds of algorithms that make different assumptions about the manner to format the different feature space for each view. Although they usually demand bigger amounts of labeled data than other SSL approaches, applying techniques like Canonical Correlation Analysis (CCA) for injecting appropriate correlation among features of different views with latent subspaces that are more reliable, especially in highly dimensional feature spaces. However, recent techniques try to alleviate this need based on sparsity properties [37]. Combination of AL with Multi-view SSL approaches has been reported in case of [38], for statistical parsing, heavily reducing the human effort. A theoretical analysis of combining these two kinds of PSL techniques has been given by Wang and Zhou [39], increasing the reasons of trusting such algorithms. Graph-based SSL algorithms have also been recently combined with AL [40].

3 Proposed framework

The key factor of the proposed combinatory scheme is the proper exploitation of two different PSL approaches obtaining the benefits from both sides and successfully reconciling the emerging trade-off between the achieved predictive quality and the employment of human effort. Hence, our ambition is to incorporate into the learning kernel of the proposed iterative process the human factor much less than the pure AL approaches. Therefore, we design our scheme so as to consume the rest amount of unlabeled instances of the total available *Budget*—this denotes a restriction over the unlabeled instances that have to be mined per experiment—through the part of SSL approach, deploying a combined approach that competes ideally the same individually acting AL approach. Thus, the costly and usually time-consuming manual annotation of human factor would be reduced. Hereinafter, this quantity will be mentioned as *al_ssl_ratio,* representing the fraction of the instances that should be annotated by the two different mechanisms into the combined approach. The aforementioned notation of O^{human} would also be used when we refer to the human factor.

Notwithstanding the small participation of O^{human}, its knowledge could both boost the total performance of the proposed framework, by applying discriminative query strategies (Q_{stg}) that extract meaningful unlabeled instances (u_i) from the corresponding U pool, so as to be provided for the labeling stage, instead of using just the confidence of the corresponding base learner [31]. At the same time, AL could control the amount of instances that would be totally mined, since the utility of the mined $x^{\text{usefulness}}$ should be much effective, especially if the corresponding Q_{stg} is really compatible with the underlying distribution of the tackled problem. In real-life scenarios, one great asset of such strategies is the production of feasible solutions by keeping a small enough *Budget* during the training process. The mathematical expression of the Q_{stg} is depicted in the following equations, where the $f_{\text{usefulness}}$ is the metric that is applied inside the Q_{stg} for measuring the necessary utility that has been selected:

$$Q_{stg} : U \times R \rightarrow x^{\text{usefulness}}, wihx^{\text{usefulness}} \subseteq U, \tag{3}$$

$$x^{\text{usefulness}} = \arg f_{\text{usefulness}}(x_i, h(L), U), \tag{4}$$

Concerning the part of SSL approach, the Self-training algorithm was preferred to be integrated into the proposed framework, as one of the most representative and well-studied products of this category [15]. This wrapper algorithm depends solely on the model that is initially built on the provided labeled pool of data with the prerequisite that the selected base learner belongs to the family of probabilistic learning models. Based upon this assumption, for each u_i a vector of class probabilities is exported whose dimension is $cl \times 1$, where the cl parameter depicts the predefined number of classes that appear into each examined dataset. Then, the class with the largest class probability is assigned to the corresponding unlabeled instance and is transferred into the L subset, whose cardinality is now growing. Although various criteria have been implemented for avoiding mislabeling errors during the phase of accepting or rejecting the decisions of the base learner, such as threshold values, similarity measures or distance metrics [41], it has been preferred here not to insert anyone of these mechanisms, but to integrate a validation stage into this operation. To be more specific, the half of the L subset is kept out of the training process and is used as a validation set. Thus, during the k-th iteration, when SSL part is asked to provide its decisions and integrate them into the current labeled set (L^k), a simple criterion is examined: if the current batch of SSL's prediction, after having appended them to the L^k subset ($L^{k'} = L^k \cup x^{\text{MCP}}$), does not improve the classification accuracy against the same metric when computed based only on L^k, then it is rejected and the operation continues. In this way, not highly additional overhead time expenses are introduced, letting the Self-training algorithm to operate under a simplistic version, reducing additionally its inherent confidence with a simple run of the base learner.

Furthermore, without re-training or applying any exhaustive searches, we neither increase computational complexity of the total framework nor reach to the point of using heuristics methods for compromising all posed restrictions. This fact favors the smooth consumption of the total *Budget* that is inserted as one of the main parameters into our learning framework and enables the proper comparison of any produced variants. The corresponding pseudo-code along with the needed input variables is given in Fig. 1. In Fig. 2 is also placed the pseudo-code of a necessary function during the preprocessing stage of the proposed framework. Moreover, for discriminating the produced variants of this framework, a favorable notation that encompasses all the necessary input quantities could be used as follows: *AL_SelfTrain (base*$_{lea}$*,* Q_{stg}*, Budget, al_ssl_ratio, steps)*. Regarding also the convenience that our framework offers, only small modifications are needed so as to obtain the pure AL and Self-training approaches, which consume the provided *Budget* with exactly the same way that the proposed combined version does. Of course, in the latter case the argument of Q_{stg} is unnecessary.

Therefore, these two families of counterparts could denote the quality of the *AL_SelfTrain* framework for any given base$_{\text{lea}}$ holding the rest of the parameters the same. To be more specific, the combined framework should outreach the similar approaches based on SSL concept, since no human intervention takes place, rendering it as the most inexpensive solution. On the contrary, the ambition is to ensure as much closer performance—measured by appropriate metrics—of the *AL_SelfTrain* with the approaches that stem solely on AL concept, since the scenario of surpassing this counterpart, which consumes all of the *Budget* through interacting with the O^{human} increasing the total expenses, regarding both time and monetization resources, would be an ideal case.

The main assumption that is based on the proposed *AL_SelfTrain* framework is the adoption of the UncS strategy that injects a complementary manner of searching for valuable information inside the U pool. To be more specific, UncS tries to refine the *base*$_{lea}$ by choosing the ui*s* which are close to the decision boundaries among the distinct classes. Therefore, it asks the human oracle to provide their labels, due to the ambiguous performance of *base*$_{lea}$ on these instances. On the other hand, Self-training variants explore the U pool based on their confidence, which is clearly enforced on regions far from the decision boundary ones. Consequently, *base*$_{lea}$ is trained through

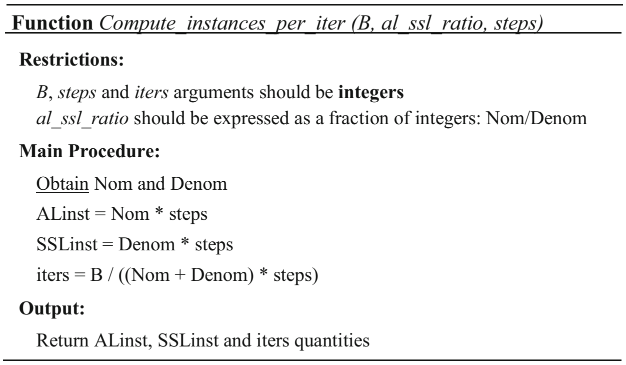
Fig. 1 The combined framework of AL_SelfTrain

```
Framework AL_SelfTrain

Mode:
  Pool-based scenario over a provided dataset ($D_{(f+1) x n}$)
  $\{x_i , y_i\}$ – i-th instance of $D_{(f+1) x n}$ with $1 \leq i \leq n$
  x – vector with f features
  y – scalar variable depicting the categorical class
Input:
  $L^0$ ($U^0$) – initially collected (un)labeled instances, $L^0 \subset D$, $U^0 \subset D$
  $L^k$ ($U^k$) – (un)labeled instances during k-th iteration, $L^k \subset D$
  $base_{lea}$ – selected base classifier
  $Q_{stg}$ – applied query strategy based on $base_{lea}$
  SSL_choice – the kind of Self-training variant
  B – Number of unlabeled instances to get labeled
  al_ssl_ratio – fraction of AL and SSL participation in labeling process
  steps – size of batches from instances to be labeled per iteration
  $H_f$ – employed human factor or crowdsourcing platform
  iters – number of combined executed iterations
Preprocess:
  ALinst, SSLinst, iters = Compute_instances_per_iter (B, al_ssl_ratio, steps)
Main Procedure:
  Set k = 0
  If (SSL_choice == Self-train Modified) do
      Sample half of the instances that belong to $L^0$ without replacement ≡ validation set
      Update $L^0$: $L^0 \leftarrow L^0 \setminus \{x_j, y_j\}$ $\forall j \in$ validation set
  While iters > 0 do
      # Active Learning part
      Train/Update $base_{lea}$ on $L^k$
      Rank through $Q_{stg}$ all $u_i \in U^k$
      Detect from $U^k$ the $x^{usefulness}$ and Provide them to $O^{human}$ for assigning the predicted class value
      B := B – ALinst
      Update $L^k$: $L^{k+1} \leftarrow L^k \cup \{x_j , O^{human}(x_j)\}$ $\forall j \in x^{usefulness}$
      Update $U^k$: $U^{k+1} \leftarrow U^k \setminus \{x_j\}$ $\forall j \in x^{usefulness}$
      k := k +1
      # Self-training part
      Train/Update $base_{lea}$ on $L^k$
      Compute class probabilities through $base_{lea}$ for all $u_i \in U^k$
      Detect from $U^k$ the $x^{MPC}$ and assign the most confident class value based on current $base_{lea}$
      B := B – SSLinst
      If (SSL_choice == Self-train Modified):
      Update $L^k$: $L^{k+1} \leftarrow L^k \cup \{x_j , \mathrm{argmax}_{Cl} P(y_j | x_j)\}$ for each $j \in x^{MPC}$
      Update $U^k$: $U^{k+1} \leftarrow U^k \setminus \{x_j\}$ for each $j \in x^{MPC}$
      If acc($base_{lea}$ ($L^{k+1}$), validation set) > acc($base_{lea}$ (L), validation set) do
          Revert update of $L^{k+1}$: $L^{k+1} \equiv L^k$
      iters := iters + 1
Output:
  Use $base_{lea}$ trained on $L^{iters}$ to predict class labels of test data
```

two separate criteria, and hopefully these iterative refinements could lead to a more trustworthy learning behavior, following a hybrid approach that compromises both strategies.

Thus, achieving similar learning behaviors with pure AL based on small *al_ssl_ratio* values is the most important ambition of this work, because only a quota of the demanded human effort by the latter approach is asked during the operation of the proposed combined one. Moreover, Random Sampling (RndS) process should be also inserted as an alternative Query Selection Strategy, settling the baseline rival from the view of AL algorithms.



```
Function Compute_instances_per_iter (B, al_ssl_ratio, steps)
  Restrictions:
    B, steps and iters arguments should be integers
    al_ssl_ratio should be expressed as a fraction of integers: Nom/Denom
  Main Procedure:
    Obtain Nom and Denom
    ALinst = Nom * steps
    SSLinst = Denom * steps
    iters = B / ((Nom + Denom) * steps)
  Output:
    Return ALinst, SSLinst and iters quantities
```

Fig. 2 The pseudo-code of the Compute_instances_per_iter function

4 Datasets

In this section, a brief description of the most important properties per examined dataset is given. We considered three of them out of the corresponding data repositories, so as to capture several modifications of the main modality that they all share, the acoustical signal. Moreover, among a large variety of related datasets, crucial role played both the publication date of them—trying to choose recently demonstrated works—and the fact of being publicly available. The interest of the related community over the application of PSL techniques over such kind of data is demonstrated in several works [29, 42, 43]

4.1 Gender identification (Voice)

The current dataset refers to the gender's identification of examined speakers using speech samples. Although this problem is easily solved through physical means, its fulfillment with ML approaches demands appropriate digital signal and feature engineering processing so as to reveal patterns that could discriminate between the male and female categories. In our case, 3.168 speech samples were produced and pre-processed by a suitable package that enables the measurement of acoustic quantities (e.g., mean frequency, standard deviation of frequency, spectral entropy and flatness). The duration of each sample has been set equal to 20 s, while peak frequency was omitted from the final constructed dataset. Hence, 20 features remain for fitting any predictive model for the included instances [44].

Moreover, the cardinality of each class is the same, leading to a perfectly balanced binary-class problem. Regarding the difficulty of this task, a simple acoustic model approach of the underlying properties that hold, may lead to really poor performance without tuning frequency thresholds, a process that may be difficult for the following two scenarios: (i) when many more examples are given, tuning would be computationally expensive, (ii) when just a small portion of data is provided, since the variance of the examined variable might not capture efficiently the new instances whose behavior would be unknown.

4.2 Identification of speakers (CHAINS)

The problem that was tackled here regards the identification of speakers among a closed set of candidate speakers through speech signals that are recorded under different recording styles [45]. This dataset is publicly available and widely known as Characterizing Individual Speakers (CHAINS). Cummins et al. chose speakers from Ireland, UK and USA, whose pronunciations vary analog to specific attributes that characterize these regions injecting dialectal homogeneity into the recordings. Three different scenarios were formatted during the initial split of the corpus: 8, 16 and 36 speakers, holding the number of male and female speakers equal in each case. The proposed mining of the original speech signals is implemented by the help of Mel-frequency Cepstral Coefficients (MFCCs) [46].

However, some modifications have been applied as it concerns the window size over which the corresponding signal transformations take place, reducing the cardinality of the total problem. Moreover, some CBIR filters have been exploited for obtaining a new view of the same problem, through Spectrogram's visualization per recorded signal. More information could be found in [47], while only one of these filters has been selected here: fuzzy Color and Texture Histogram (FCTH) filter [48], which offers good results even on images that have been exposed on smoothing of deformations.

Finally, the scenario of 8 speakers has been selected whose recording style is the 'solo,' which means that all the included speakers read the corresponding phrases at their own manner, without any noisy source. The formulation of the final dataset is (1298, 43) without counting the class variable, where the 25 features correspond to the acoustical transformation and the rest to the CBIR filter. It has to be mentioned that the remaining FCTH features were kept after having been preprocessed by a feature selection method which removes all the attributes whose values along the instances vary too less.

4.3 Detection of emotion (ANAD)

This kind of dataset is related to the emotion expressed through speech signals that are extracted from videos of Arabic talk shows. Although similar works have been accomplished for various languages, only recently this dataset came up concerning Arabic corpus [49]. Instead of using just a text-based solution that does not reveal any clue about the emotional situation of any speaking entity, causing possible misunderstandings when the meaning of a sentence is implicit. Apart from applications where deaf people could be favored to communicate accurately with their co-speakers, emerging tasks such as the adoption of anchors in media could be enhanced regarding their quality of service.

As it concerns the creation and annotation of this dataset, a small amount of video signals was initially recorded and afterward was provided to 18 listeners. Their task was to decide about the prevailing emotional situation of participants among these of angry, happy, and surprised. After removing some specific segments from the raw data, all the rest were divided into chunks with duration equal to 1 s.

Eventually, 1384 instances were created, where the quota of each class is 53.58%, 36.5% and 9.9%, respectively. The features that were used are mainly based on 25 low-level acoustical features (e.g., MFCC, ZCR) and a number of variables that are produced applying some well-known statistical functions over these. The final amount of the remaining features sums up to 844.

5 Experiments and results

5.1 Experimentation methodology

This section describes the experimental procedure that was executed so as to implement proper comparisons among the algorithms produced by the proposed framework, its two main variants—the individually acting algorithms of AL and SSL—the baseline method of AL concept considering RndS Strategy, as well as one similar approach embedded into the aforementioned framework. Additionally to our original work, demonstrated in [50], a variant of Self-training Scheme has also been applied, as well as the use of some Dense Deep Neural Networks (Dense DNNs) [51] and VFI model [52], enriching the total experimental procedure so as to obtain even more safe conclusions about the applicability of using combined algorithms of AL + SSL in practical problems.

Before reporting the base learners, we have to define the selected Query Strategies. Actually, Uncertainty Sampling Q_{stg} (UncS) has been preferred in the context of this work, as it has been mentioned previously, as one of the most widely used and easily applicable in the literature [6]. This choice enables the creation of several versions of the same Strategy, trying to define with alternative manner the most uncertain instances according to the predictions of the base$_{\text{lea}}$ and the selected measure of uncertainty. The three preferred versions of UncS strategy are the following:

Entropy (EntS), a popular formula, which measures the average information revealed by any examined variable. Its general form sums up the—*zlog(z)* quantity for each class and selects this that induces the maximum information, where z is replaced by a *posteriori* probability *P(y|x)* as follows:

$$f_{\text{Entropy}}(\boldsymbol{x}_i) = \arg\max_{\boldsymbol{x}_i \in U} - \sum_{\omega} P(\omega|\boldsymbol{x}_i) \log P(\omega|\boldsymbol{x}_i), \quad (5)$$

Smallest Margin (MrgS), a metric that translates the sense of uncertainty into the closeness of the two largest likelihoods between the contained classes. Thus, the smaller is this value, the most ambiguous is the behavior of the base$_{\text{lea}}$ according to this instance and has then to be extracted so as to be annotated by O^{human}:

$$f_{\text{SmallestMargin}}(\boldsymbol{x}_i) = \arg\min_{\boldsymbol{x}_i \in U} \left[P\left(\omega^1|\boldsymbol{x}_i\right) - P\left(\omega^2|\boldsymbol{x}_i\right) \right], \quad (6)$$

Minimum Standard Deviation (StdS) is the well-known mathematical function that takes into consideration a posteriori probability values for all classes per instance. The smaller this value is, the more uncertain is the base$_{\text{lea}}$ about this instance.

Hence, the 14 separate PSL approaches that would be composed here, independently of the parameters apart from Q_{stg}, could be summarized as follows:

- three combined approaches exploiting the simple Self-training scheme: AL_SelfTrain(EntS), AL_SelfTrain(MrgS) and AL_SelfTrain(StdS),
- three combined approaches exploiting the modified Self-training scheme: AL_SelfTrainMod(EntS), AL_SelfTrainMod(MrgS) and AL_SelfTrainMod (StdS),
- three pure AL approaches: AL(EntS), AL(MrgS) and AL(StdS),
- two pure SSL approaches: the default Self-training (SelfTrain) and its modified variant (SelfTrainMod),
- the baseline of AL concept: AL(RndS) which provides randomly selected instances to O$^{\text{human}}$, and
- two hybrid approaches of the proposed framework: AL_SelfTrain(RndS) and AL_SelfTrainMod(RndS), where the AL(RndS) and Self-training algorithms act interchangeably under the same scheme.

In order to provide more comprehensible notations of the already mentioned algorithms, we just recorded the metric under the UncS strategy, while in case of Random Sampling we used a suitable abbreviation of this Query Strategy (RndS).

Regarding the rest of the involved parameters, and taking into consideration the restriction that is posed by the function of Fig. 2 about the integer format of the input arguments, the next set of values has been selected: *steps* ∈ {2, 5, 10, 25}, while the pair of (*B, al_ssl_ratio*) ∈ {(200, 1/1), (200, 1/3)}. Hence, the number of combined iterations for the two different pairs of (*B, al_ssl_ratio*) is 50, 20, 10, 4 and 25, 10, 5, 2 analog to the value of *steps* parameter, where each combined iteration consists of one exactly iteration of AL and SSL mechanisms. It is evident that per each actively labeled batch of instances by O^{human}, the batches that are assessed by the upcoming SSL algorithm are either equal or three times larger, reducing the spent human effort compared with the pure AL approach by 50% and 75%, respectively.

Although the selected values of *Budget* parameter seem quite small, they indeed keep pace with the similarly small initial cardinality of labeled subsets (L^0). More specifically, the three examined datasets were split to train and test subsets, covering the 90% and 10% of the total dataset, respectively. Then, the train dataset ($\mathrm{D} \equiv \mathrm{L} \cup \mathrm{U}$) is divided into L and U subsets according to Labeled Ratio parameter—here mentioned as R and measured in percentage values—whose value is usually small enough for simulating the scarce of labeled data. Its formula is shown in next equation:

$$R(\%) = \frac{\text{cardinality}(L)}{\text{cardinality}(D)}, \tag{7}$$

The values of R during our experimental procedure were equal to 2% and 10%, while only the half of them were used for initializing the Self-training Modified versions, because of the creation of the validation set. The cardinalities of the corresponding L, U, and test subsets for all our evaluated datasets are presented here (Table 1):

Summing up all the constructed scenarios, there exist three datasets, examined under two different R values, two separate combinations of applying the synergy of AL and SSL mechanisms consuming the provided *Budget*. This leads to 12 (3 × 2 × 2) separate classification problems, where each one operates under four distinct step-based approaches. The last parameter that has to be selected is this of base$_{\text{lea}}$. Eight different classifiers have been contained for evaluating their learning behavior under the proposed framework:

- Extremely Randomized Trees (ExT): an ensemble learner that fits several unpruned trees over various subsamples of the provided data, aggregating their decisions for achieving accurate predictions [53],
- Random Forest (Rf): an ensemble learner which is differentiated mainly by the ExT because of the resampling process during the formatting process of the decision trees, since each subsample is chosen through replacement [54],
- Multi-Layer Perceptron (MLP): a typical neural network with one layer of 100 neurons that uses stochastic gradient decent method for weight optimization [55]. Additionally, other two variants of this Neural Network were used: MLP_2layers and MLP_3layers with two layers of 100 and 50 neurons, as well as three layers with 200, 100 and 50 neurons, respectively. All of these Neural Network classifiers share the same activation function (≡ReLU) and solver (≡Adam),
- k-Nearest Neighbors (kNN): a well-known lazy classifier that applies a voting stage of the decision of the of k closer instances to any test example [56],
- Naive Bayes (NB): the popular learner that is based on Bayes' Theorem and is exporting the class that maximizes the maximum a posteriori hypothesis [57],
- Voting Feature Intervals (VFI): a learner that constructs intervals for each feature and class counts are recorded for each interval for each feature. The classification of an unseen instance is performed by using a voting scheme among features' interval confidence [58].

Table 1 Representative quantities of examined datasets

Datasets	Properties			
	Features	Train instances		Test instances
		R = 2%	R = 10%	
Voice	20	L = 56	L = 284	317
		U = 2795	U = 2567	
CHAINS	43	L = 23	L = 116	129
		U = 1144	U = 1051	
ANAD	844	L = 24	L = 124	139
		U = 1220	U = 1120	

Moving to more technical details, all the included learners are adopted with their default values from sklearn Python package [59]. Therefore, kNN will be symbolized as 5NN, hereinafter. Moreover, all the L^0 subsets were formatted through a stratified sampling process and all the experiments were repeated three times. The main performance metrics for our experiments have been selected to be the classification accuracy (acc), precision (prec), recall and the f1-score. This last metric constitutes a weighted average of precision and recall, which depicts the exactness and completeness of any tested classifier. However, f1-score is a great solution for leveraging the importance of recorded results over imbalanced datasets [60].

As it regards the produced results, appropriate comparisons have been made so as to understand the predictive ability of the composed AL + SSL approaches per base learner, as well as to notice the impact of the batch size over the performance of all included classifiers.

Due to lack of space, for facilitating the presentation of these results, only a small portion of them are demonstrated here, while the rest have been placed along with our code implementation in the following link: https://github.com/terry07/AL_SelfTrain_NCAA.

5.2 Quantitative description

In the quantitative part of our experiments, the first stage includes the consideration of all the 4-performance metrics for each one of the 14 included algorithms, recorded per each iteration over all the 4 distinct step-based scenarios. Then, application of Friedman statistical test takes place in order to obtain the related ranking per base$_{\text{lea}}$ [61]. Trying to reduce the volume of the results, an average ranking per same *al_ssl_ratio* has been adopted, ignoring thus the different $R(\%)$ value, rendering the separate problems to 6. Secondly, a post hoc test of Nemenyi [62] is applied so as to ascertain the statistical importance of the obtained behaviors.

From this second stage, a Critical Difference (CD) value is computed, which denotes the minimum difference between the corresponding rankings of two different algorithms so as to be considered as statistically different. The significance level during the selected post hoc test is equal to 0.05. Although slight changes occur between the different performance metrics, the average rankings over all of them depict the underlying relationships about the predictive performance of each PSL algorithm per separate classifier.

After having depicted this kind of results for two out of the eight included classifiers (Tables 2, 3), we provide Table 4, which summarizes the most important comparisons for which we are interested in, as it has been already mentioned. Therefore, for each Q_{stg} and per distinct classifier, we count the frequency of the cases that the ranking of the examined Query strategy under the proposed framework is higher than:

1. the same AL approach with the same Query Strategy: AL(Q_{stg}),
2. the baseline of AL concept: AL(RndS),
3. the hybrid approach which uses RndS: AL_SelfTrain(RndS), and
4. the pure SSL variant: SelfTrain.

These comparisons take place for both variants of Self-training algorithm.

We observed that all the examined classifiers managed to outperform the pure Self-training variant almost in all cases—small deterioration is presented in case of EntS combined especially with tree-based learners—as well as the hybrid approach of AL_SelfTrain(RndS), where 5NN algorithm was recorded the smaller number of successes, probably because the limited initially provided labeled data affected its predictive ability. This situation has also affected the behavior of the MLP-based learners compared with the AL(RndS) approach, since their behavior was more often than not inferior to the baseline strategy of AL. This performance is of course not accepted, but we should mention the fact that these strategies consume at least two times more human resources than the proposed, setting a quite strict baseline. EntS Strategy also did not perform well in the most cases of this comparison, even when tree-based learners were exploited, whose behavior was robust enough combined with the other two strategies. Another main reason why this happen is the dependence of the

Table 2 Friedman ranking for all performance metrics in case of MLP_3layers Classifier

Algorithm	ANAD		CHAINS		Voice	
	al_ssl_ratio:1/1	*al_ssl_ratio:1/3*	*al_ssl_ratio:1/1*	*al_ssl_ratio:1/3*	*al_ssl_ratio:1/1*	*al_ssl_ratio:1/3*
AL(MrgS)	4.416	5.799	4.772	4.295	4.267	7.029
AL(StdS)	5.364	4.815	5.875	5.080	3.263	4.764
AL_SelfTrainMod(MrgS)	6.930	6.830	6.582	8.419	8.248	7.780
AL(RndS)	7.295	6.769	5.173	7.883	6.106	7.321
AL(EntS)	6.381	6.185	5.202	4.199	4.839	4.957
AL_SelfTrainMod(StdS)	6.387	6.830	7.034	7.931	7.972	5.677
AL_SelfTrain(EntS)	7.913	8.916	7.513	7.659	7.055	7.588
AL_SelfTrain(MrgS)	7.824	8.680	6.993	7.321	6.114	8.337
AL_SelfTrainMod(EntS)	7.847	7.399	6.633	7.930	6.947	7.113
AL_SelfTrain(StdS)	8.546	8.627	9.055	7.293	9.026	7.573
AL_SelfTrainMod(RndS)	8.973	8.110	9.036	7.652	11.113	7.542
AL_SelfTrain(RndS)	9.166	9.043	9.247	8.865	6.951	9.628
SelfTrainMod	7.471	6.831	11.135	10.140	11.588	9.686
SelfTrain	10.487	10.166	10.750	10.333	11.511	10.005
CD value	*1.52*					

Table 3 Frequency count of proposed framework victories concerning statistical ranking

Classifier/self-training variant	Victories			
ExT/simple	*AL(Qstg)*	*AL(RndS)*	*AL_SelfTrain (RndS)*	*SelfTrain*
Ent	3	0	0	6
Mrg	0	4	5	6
Std	0	4	6	6
ExT/Modified				
Ent	2	0	0	2
Mrg	0	3	6	6
Std	0	5	6	6
Rf/simple				
Ent	1	0	0	3
Mrg	0	3	6	6
Std	0	3	6	6
Rf/Modified				
Ent	3	0	1	6
Mrg	0	5	6	6
Std	0	3	6	6
NB/simple				
Ent	2	2	3	4
Mrg	0	4	4	6
Std	1	5	5	6
NB/Modified				
Ent	4	3	3	4
Mrg	0	5	5	5
Std	2	5	5	6
VFI/simple				
Ent	0	0	6	6
Mrg	0	1	5	6
Std	0	0	6	6
VFI/Modified				
Ent	1	3	4	6
Mrg	0	5	4	6
Std	0	4	4	6
5NN/simple				
Ent	2	0	0	6
Mrg	1	2	3	6
Std	1	3	4	6
5NN/Modified				
Ent	1	0	3	4
Mrg	1	0	6	6
Std	1	1	6	6
MLPModel/simple				
Ent	0	0	4	5
Mrg	0	0	6	5
Std	0	1	5	5
MLPModel/Modified				
Ent	0	3	4	5
Mrg	0	3	6	6
Std	0	3	5	6

Table 3 (continued)

Classifier/self-training variant	Victories			
ExT/simple	*AL(Qstg)*	*AL(RndS)*	*AL_SelfTrain (RndS)*	*SelfTrain*
MLPModel_2layers/simple				
Ent	1	0	4	6
Mrg	0	0	4	6
Std	0	0	3	6
MLPModel_2layers/Modified				
Ent	1	0	5	4
Mrg	0	0	5	5
Std	1	2	6	5
MLPModel_3layers/simple				
Ent	0	1	5	6
Mrg	0	1	6	6
Std	0	1	5	6
MLPModel_3layers/Modified				
Ent	0	1	6	5
Mrg	0	1	4	6
Std	0	1	5	6

Table 4 Conclusions identification of the most favorable step value

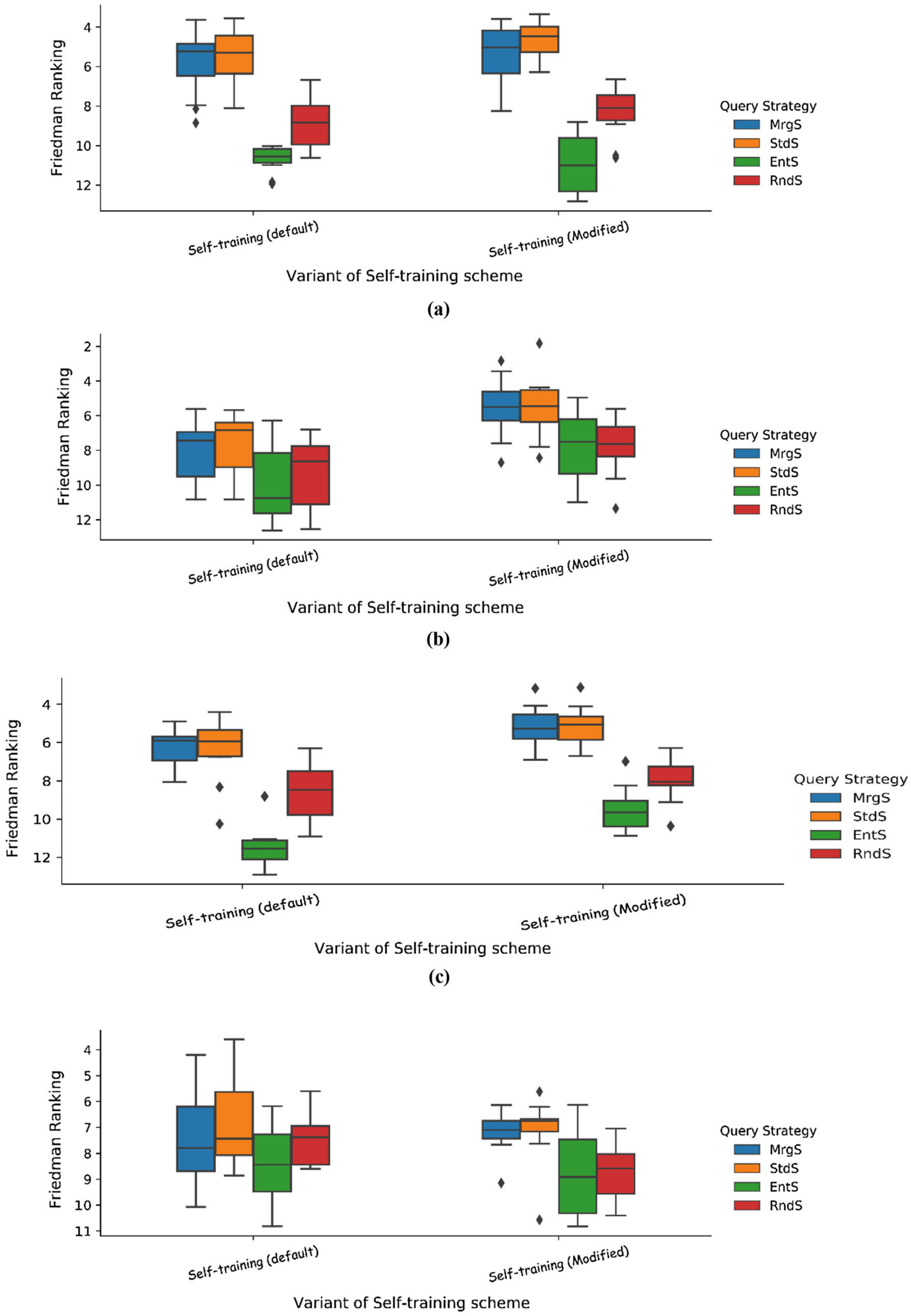
Dataset {R(%), al_ssl_ratio}	Final value/improvement/stability							
	ExT	*Rf*	*NB*	*VFI*	*5NN*	*MLP*	*MLP_2 layers*	*MLP_3 layers*
ANAD								
{2%, 1/1}	25/25/2	5/5/2	5/5/5	2/2/10	25/25/5	5/5/2	25/25/2	25/25/2
{2%, 1/3}	5/5/2	2/2/2	10/10/5	10/10/25	5/5/10	2/2/2	25/25/2	10/10/2
{10%, 1/1}	10/10/5	10/10/2	2/2/5	5/5/10	2/2/10	10/2/5	10/10/2	25/25/2
{10%, 1/3}	2/2/2	10/10/2	2/2/25	2/2/25	5/5/5	2/2/2	2/2/2	2/2/2
CHAINS								
{2%, 1/1}	10/10/2	25/25/2	25/25/5	25/25/5	2/2/5	2/2/2	25/25/2	25/25/2
{2%, 1/3}	25/25/2	25/25/2	5/5/2	25/25/2	25/25/2	25/25/2	10/10/2	10/10/2
{10%, 1/1}	25/25/2	10/10/2	10/10/10	25/25/5	5/5/10	2/2/2	25/25/2	5/5/2
{10%, 1/3}	25/25/2	10/10/2	10/10/5	25/25/2	5/5/5	2/2/2	5/5/2	5/5/2
Voice								
{2%, 1/1}	5/5/2	2/2/10	2/2/5	5/5/5	2/2/10	2/2/5	2/2/2	2/2/2
{2%, 1/3}	5/5/2	2/2/5	2/2/5	10/10/5	5/5/10	5/5/10	5/5/2	25/25/2
{10%, 1/1}	2/2/2	5/5/10	25/25/5	10/10/10	25/25/25	10/10/2	10/10/2	10/10/2
{10%, 1/3}	25/25/2	2/2/5	5/5/10	25/25/5	10/10/10	5/5/2	2/2/2	5/5/2

majority of the included learners by their inherent parameters. In our work, all the corresponding parameters have been set to their default values, while only VFI and NB learners are free of parameters.

5.3 Qualitative description

To mitigate the aforementioned phenomenon, a tuning stage could boost the total learning performance, so as to be much more competitive against the pure AL approaches with the same Q_{stg}, since only in a few situations the produced by the proposed framework approaches outreached them. However, the statistical difference among them is not important in the most cases, as it is recorded from our results and the corresponding post hoc test. In particular, the modified variants scored better results,

Fig. 3 Boxplots based on the Friedman Ranking results comparing ▶ classification accuracy of the next learners: **a** ExT, **b** Rf, **c** NB, **d** VFI, **e** 5NN, **f** MLP, **g** MLP_2layers, **h** MLP_3layers


Friedman Ranking
4
6
8
10
12
Self-training (default)
Self-training (Modified)
Variant of Self-training scheme
Query Strategy
MrgS
StdS
EntS
RndS
Friedman Ranking
2
4
6
8
10
12
Self-training (default)
Self-training (Modified)
Variant of Self-training scheme
Query Strategy
MrgS
StdS
EntS
RndS
Friedman Ranking
4
6
8
10
12
Self-training (default)
Self-training (Modified)
Variant of Self-training scheme
Query Strategy
MrgS
StdS
EntS
RndS
Friedman Ranking
4
5
6
7
8
9
10
11
Self-training (default)
Self-training (Modified)
Variant of Self-training scheme
Query Strategy
MrgS
StdS
EntS
RndS

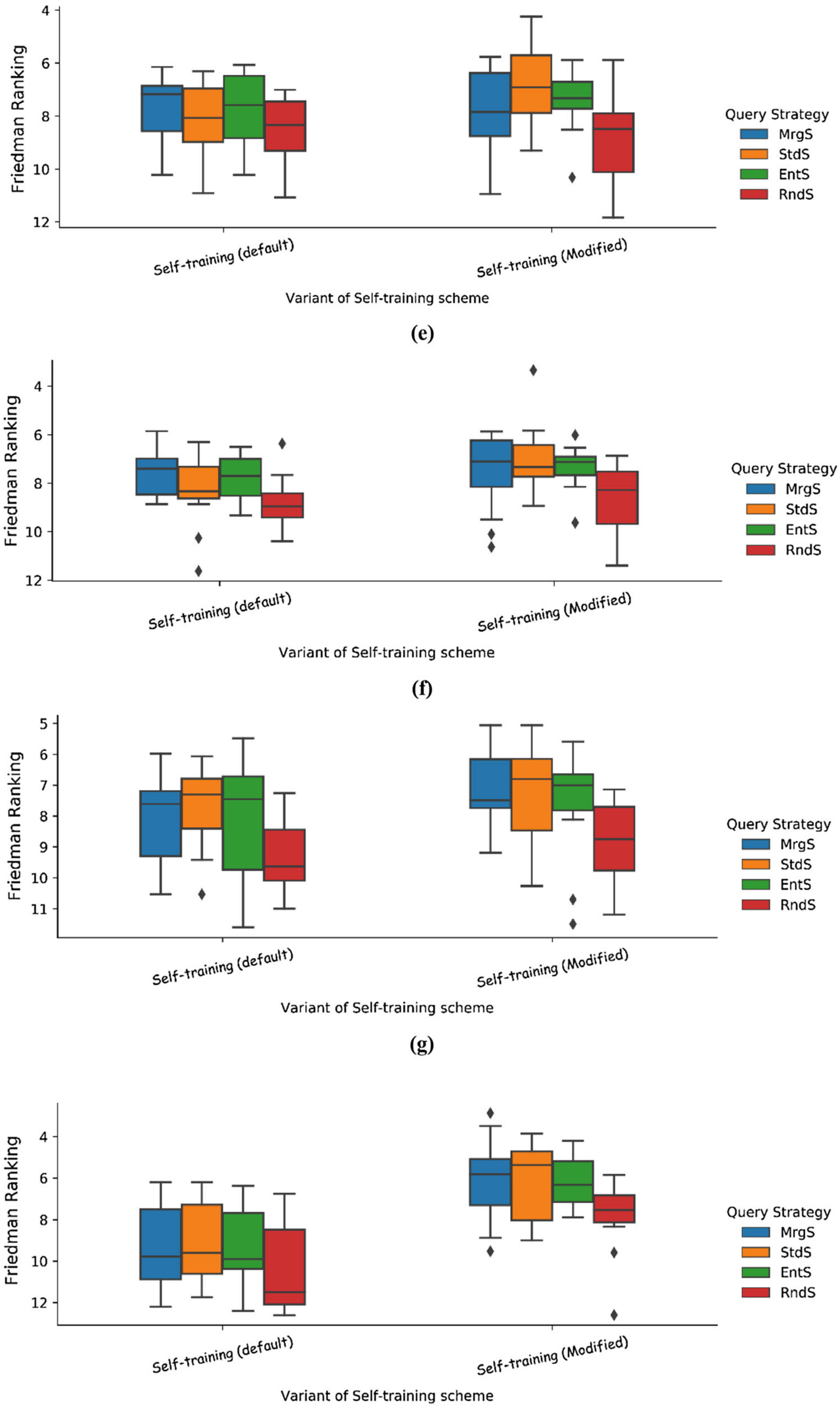
(a)

(b)

(c)

(d)

Fig. 3 continued



(e)

(f)

(g)

(h)

protecting the validity of the training data that were augmented by the human factor, leaving the simple Self-training scheme to suffer from its inherent confidence. For this reason, we provide some boxplot visualizations which depict the distribution of the rankings per classifier for any selected performance metric. The approaches that are based on Rf and MLP_3layers have been highly favored, while in general, all the approaches which adopted Self-training Modified managed to outperform their main rival. Due to lack of space, only these that are related to classification accuracy metric are placed in the next figure (Fig. 3):

For evaluation purposes, we have taken into consideration three different measurements to verify under which *step* value the different learning concepts are better favored. We highlight first the *final value* per metric, after having executed all the necessary iterations. Secondly, we measured the difference of the selected 4-performance metrics between the final iteration and the initial stage (*improvement*). Then, we counted the times that the corresponding performance metric was reduced among the executed iterations and normalized it using the total number of iterations, since this value depends on the *step* value (*stability*). All these measurements were applied over the 3 different executions that were made, computing their average value, before we rank the performance per learner for each *step* value. The ideal cases are the maximum of the *final value* and the *improvement* criteria, as well as the minimum of *stability* criterion. The best step per criterion is recorded in Table 4.

Via this qualitative investigation, it is evident that the value of *step* parameter equal to 2 is the most favorable, as it concerns the average of all the examined performance metrics. Actually, this scenario was the most favorable in 107 out of the 288 cases, while the second best—*step* equal to 5—follows with 71 successes. The rest scenarios based on step equal to 25 and 10 managed to achieve the best performance for 56 and 54 examined cases. These results are quite reasonable, especially in cases that the human oracle interacts with the algorithms, injecting its decisions on the total procedure, since when larger amount of iterations are conducted, each newly refined model is provided with labels that do not suffer from noise, since we have employed an ideal oracle. It has to be mentioned that in all cases, the step value regarding the first two criteria coincides, while the third one has been highly favored by the smallest *step* value.

To sum up, the proposed strategy of formatting an AL + SSL framework that tries to increase the complementary behavior of these distinct PSL techniques could be used in practice for tackling real-life pool-based problems without spending much human effort, since the obtained learning behaviors tend to outperform their main rivals. Different behaviors are recorded per base learner. The most important insights are the fact that MLP-based models did not perform well, mainly because of the small initially provided L subsets, while, on the other hand, tree-based ensemble learners proved more compatible with this property. Furthermore, the use of validation set inside the Self-training scheme eliminates noisy batches that could mislead the total algorithm, and at the same time, favored the improvement of less accurate models, like VFI and NB, since only safer decisions were included in the current labeled subset. Similar model-based mechanisms should be examined further for increasing the predictive quality of self-confident algorithms, or even employ more adaptive versions, trying to find the best size of inserting batch. However, such modifications would increase further the complexity of the framework, something that is left for future work.

6 Conclusions

In this paper, we proposed a framework of combining Active Learning strategy along with Semi-supervised Learning methods, oriented toward reducing the human burden over real-life scenarios. In practice, abundant unlabeled data are usually easily collected, in contrast with labeled examples whose ground truth demands either expert's knowledge or contribution by larger groups of human entities, which have to be motivated by related rewards [63]. In both cases, monetization expenses and time delays occur, while these factors are not easily constrained into practical applications. Hence, relaxation of this necessity is the main ambition here, trying to achieve at the same time better learning behavior, compared at least with the baseline of AL—Random Sampling Query Strategy—and the corresponding individually acting SSL approaches. Additionally, through augmenting the cardinality of the initially collected L subset through two different aspects, the obtained learning behavior could be boosted toward more accurate predictions, because of the complementary behavior that characterizes AL and SSL strategies.

This combination competes the pure AL scenario that demands more human annotations than the proposed, since it asks human entities for all of its decisions, outperforming at the same time almost always the pure SSL scenario. Furthermore, attempting to hinder possible noisy decisions over the unlabeled examples because of the confidence that governs the simple Self-training algorithm, a variant of this SSL scheme was also implemented which tries to verify its positive effect by obtaining proper indications from a validation subset. This subset is formatted through a portion of the initially available training data.

The adoption of this strategy was proved to be much more beneficial against the simple Self-train scheme, regarding prediction metrics, stability of learning performance and time efficiency. Simultaneously, it overcomes the inherent myopic predictions that could appear on several learning models, whose confidence may surpass a predefined accuracy threshold, even a strict one, but finally reduce the total performance when augmentation of L subset takes place [64].

The constructed framework constitutes a straightforward implementation of this combination—which is highly appreciated the past few years as a really effective solution by the ML community [31, 65–67]—depending on a small amount of parameters, so as to tune the consumption of the provided *Budget* properly according to user's choices about the participation of both human factor and automated learner into the labeling stage. During its operation, any probabilistic classifier—exporting probability distributions over the classes with either natural or elaborate manner—is supported, since the described operation of the mechanisms that mine the unlabeled pool of instances need class probabilities for formulating the appropriate decisions. Three different datasets that are based on Data Mining from Acoustical Signals and Digital Signal Processing were evaluated in this context, since this field has already highly met the circumstances under which vast amounts of collected data demand much effort for being employed into predictive tools [29, 43]. The produced results clarified several aspects, such as the quality of learning behavior regarding the size of the batches that are extracted per iteration and the compatibility of several functions that measure the usefulness of predictive Decision Profiles into Uncertainty Sampling Query Strategy that stem from various Machine Learning models.

Future works that can be considered to potentially extend this work are mentioned here. First, the employment of different types of Deep Neural Networks into the learning kernel of the proposed framework should be examined, such as Convolutional Neural Networks (CNNs) or Long Short Term Memory (LSTMs) [68], especially during the concept of SSL part, where the selection of unlabeled instances is usually relied purely on the confidence of the base learner. The factor of interpretability should also be explored [69], since many of these algorithms sacrifice this property against boosting their predictive accuracy, but practical applications require a better balance between these two factors [70]. Different SSL approaches could also be integrated into the proposed framework, such as multi-view schemes that seem to be compatible enough with the nature of raw-data or S3VMS [71], while weakly supervised methods that increment their knowledge from weak annotations—which may suffer from noisy labels—have been actually proven beneficial for acoustical signals [72].

Another one aspect could be the utilization of different kinds of Query Sampling strategies, since UncS, although it favors the time feasibility, often is rendered as a myopic approach [6]. The Query-By-Committee solution, which applies a voting scheme over the decisions of the participating learners, seems a promising solution, also enabling the use of non-probabilistic learners [73]. Furthermore, different approaches should be adopted for applying AL to massive high-dimensional data, as it is proposed in [74], since Big Data is a hot-topic nowadays, or for selecting among different Query Sampling Strategies the most profitable per iteration adaptively [75], as in case of frameworks like the proposed one. Use of multi-armed bandit methods is required in the latter scenario for gaining confident insights [76]. Finally, the design of query strategies that try to optimize more than one criterion so as to tackle with the efficient ranking of unlabeled examples is an active field for research [77].

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this paper.

References

1. Khamassi I, Sayed-Mouchaweh M, Hammami M, Ghédira K (2018) Discussion and review on evolving data streams and concept drift adapting. Evol Syst 9:1–23. https://doi.org/10.1007/s12530-016-9168-2
2. Shayaa S, Jaafar NI, Bahri S, Sulaiman A, Seuk Wai P, Wai Chung Y, Piprani AZ, Al-Garadi MA (2018) Sentiment analysis of big data: methods, applications, and open challenges. IEEE Access 6:37807–37827. https://doi.org/10.1109/ACCESS.2018.2851311
3. Nguyen AT, Wallace BC, Lease M (2015) Combining crowd and expert labels using decision theoretic active learning. In: HCOMP. pp 120–129
4. Schwenker F, Trentin E (2014) Pattern classification and clustering: a review of partially supervised learning approaches. Pattern Recognit Lett 37:4–14. https://doi.org/10.1016/j.patrec.2013.10.017
5. Kostopoulos G, Karlos S, Kotsiantis S, Ragos O (2018) Semi-supervised regression: a recent review. J Intell Fuzzy Syst 35:1483–1500. https://doi.org/10.3233/JIFS-169689
6. Settles B (2012) Active learning. Morgan & Claypool Publishers, San Rafael
7. Akyürek HA, Koçer B (2019) Semi-supervised fuzzy neighborhood preserving analysis for feature extraction in hyperspectral remote sensing images. Neural Comput Appl 31:3385–3415. https://doi.org/10.1007/s00521-017-3279-y
8. Liu W, Zhang L, Tao D, Cheng J (2017) Support vector machine active learning by Hessian regularization. J Vis Commun Image Represent 49:47–56. https://doi.org/10.1016/j.jvcir.2017.08.001

9. Long B, Bian J, Chapelle O, Zhang Y, Inagaki Y, Chang Y (2015) Active learning for ranking through expected loss optimization. IEEE Trans Knowl Data Eng 27:1180–1191. https://doi.org/10.1109/TKDE.2014.2365785
10. Freund Y, Seung HS, Shamir E, Tishby N (1997) Selective sampling using the query by committee algorithm. Mach Learn 28:133–168. https://doi.org/10.1023/A:1007330508534
11. Granell E, Romero V, Martínez-Hinarejos CD (2018) Multimodality, interactivity, and crowdsourcing for document transcription. Comput Intell 34:398–419. https://doi.org/10.1111/coin.12169
12. Elahi M, Ricci F, Rubens N (2016) A survey of active learning in collaborative filtering recommender systems. Comput Sci Rev 20:29–50. https://doi.org/10.1016/j.cosrev.2016.05.002
13. Zhang C (2015) Active learning from weak and strong labelers. In: NIPS. pp 703–711
14. Karlos S, Fazakis N, Kotsiantis S, Sgarbas K (2016) A semisupervised cascade classification algorithm. Appl Comput Intell Soft Comput 2016:14. https://doi.org/10.1155/2016/5919717
15. Triguero I, García S, Herrera F (2015) Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. Knowl Inf Syst 42:245–284. https://doi.org/10.1007/s10115-013-0706-y
16. Kang P, Kim D, Cho S (2016) Semi-supervised support vector regression based on self-training with label uncertainty: an application to virtual metrology in semiconductor manufacturing. Expert Syst Appl 51:85–106. https://doi.org/10.1016/j.eswa.2015.12.027
17. Dalal MK, Zaveri MA (2013) Semisupervised learning based opinion summarization and classification for online product reviews. Appl Comput Intell Soft Comput 2013:1–8. https://doi.org/10.1155/2013/910706
18. Wu D, Luo X, Wang G, Shang M, Yuan Y, Yan H (2018) A highly accurate framework for self-labeled semisupervised classification in industrial applications. IEEE Trans Ind Inform 14:909–920. https://doi.org/10.1109/TII.2017.2737827
19. Wang Y, Xu X, Zhao H, Hua Z (2010) Semi-supervised learning based on nearest neighbor rule and cut edges. Knowl Based Syst 23:547–554. https://doi.org/10.1016/j.knosys.2010.03.012
20. Sabata T, Pulc P, Holena M (2018) Semi-supervised and active learning in video scene classification from statistical features. In: Krempl G, Lemaire V, Kottke D, Calma A, Holzinger A, Polikar R, Sick B (eds.), IAL@PKDD/ECML. CEUR-WS.org, pp 24–35
21. Yarowsky D, David (1995) Unsupervised word sense disambiguation rivaling supervised methods. In: Proceedings of the 33rd annual meeting on association for computational linguistics. Association for Computational Linguistics, Morristown, NJ, USA, pp 189–196
22. Potapova R, Potapov V (2016) On Individual Polyinformativity of Speech and Voice Regarding Speakers Auditive Attribution (Forensic Phonetic Aspect). Speech and Computer. SPECOM. Lecture Notes in Computer Science, vol 9811. Springer, Cham, pp 507–514
23. Kunešová M, Radová V (2015) Ideas for clustering of similar models of a speaker in an online speaker diarization system. TSD. Springer, Cham, pp 225–233
24. McCallumzy Andrew Kachites;Nigamy K (1998) Employing EM and pool-based active learning for text classification. In: ICML. pp 350–358
25. Muslea I, Minton S, Knoblock CA (2002) Active+ semi-supervised learning = robust multi-view learning. In: ICML. pp 435–442
26. Zhou Z-H, Chen K-J, Dai H-B (2006) Enhancing relevance feedback in image retrieval using unlabeled data. ACM Trans Inf Syst 24:219–244. https://doi.org/10.1145/1148020.1148023
27. Hanneke S (2014) Theory of disagreement-based active learning. Found Trends® Mach Learn 7:131–309. https://doi.org/10.1561/2200000037
28. Zhou ZH, Li M (2010) Semi-supervised learning by disagreement. Knowl Inf Syst 24:415–439. https://doi.org/10.1007/s10115-009-0209-z
29. Yu D, Varadarajan B, Deng L, Acero A (2010) Active learning and semi-supervised learning for speech recognition: a unified framework using the global entropy reduction maximization criterion. Comput Speech Lang 24:433–444. https://doi.org/10.1016/j.csl.2009.03.004
30. Hajmohammadi MS, Ibrahim R, Selamat A, Fujita H (2015) Combination of active learning and self-training for cross-lingual sentiment classification with density analysis of unlabelled samples. Inf Sci (Ny) 317:67–77
31. Han W, Coutinho E, Ruan H, Li H, Schuller B, Yu X, Zhu X (2016) Semi-supervised active learning for sound classification in hybrid learning environments. PLoS ONE 11:1–23. https://doi.org/10.1371/journal.pone.0162075
32. Tran VC, Nguyen NT, Fujita H, Hoang DT, Hwang D (2017) A combination of active learning and self-learning for named entity recognition on Twitter using conditional random fields. Knowl Based Syst 132:179–187. https://doi.org/10.1016/J.KNOSYS.2017.06.023
33. Calma A, Reitmaier T, Sick B (2018) Semi-supervised active learning for support vector machines: a novel approach that exploits structure information in data. Inf Sci (Ny) 456:13–33. https://doi.org/10.1016/J.INS.2018.04.063
34. Reitmaier T, Sick B (2013) Let us know your decision: Pool-based active training of a generative classifier with the selection strategy 4DS. Inf Sci (Ny) 230:106–131. https://doi.org/10.1016/J.INS.2012.11.015
35. Ding S, Zhu Z, Zhang X (2017) An overview on semi-supervised support vector machine. Neural Comput Appl 28:969–978. https://doi.org/10.1007/s00521-015-2113-7
36. van Engelen JE, Hoos HH (2020) A survey on semi-supervised learning. Mach Learn 109:373–440. https://doi.org/10.1007/s10994-019-05855-6
37. Hou S, Liu H, Sun Q (2019) Sparse regularized discriminative canonical correlation analysis for multi-view semi-supervised learning. Neural Comput Appl 31:7351–7359. https://doi.org/10.1007/s00521-018-3582-2
38. Hwa R, Osborne M, Sarkar A, Steedman M (2003) Corrected Co-training for Statistical Parsers. In: ICML 2003
39. Wang W, Zhou Z-H (2008) On multi-view active learning and the combination with semi-supervised learning. In: Proceedings of the 25th international conference on machine learning. association for computing machinery, New York, NY, USA, pp 1152–1159
40. Huang L, Liu Y, Liu X, Wang X, Lang B (2014) Graph-based active semi-supervised learning: a new perspective for relieving multi-class annotation labor. In: 2014 IEEE international conference on multimedia and expo (ICME). IEEE, pp 1–6
41. Li M, Zhou Z-H (2005) {SETRED:} Self-training with Editing. In: Ho TB, Cheung DW-L, Liu H (eds.), Advances in Knowledge Discovery and Data Mining, 9th Pacific-Asia Conf. {PAKDD}, Hanoi, Vietnam, Proceedings, Springer, pp 611–621. https://doi.org/10.1007/11430919_71
42. Tur G, Hakkani-Tür D, Schapire RE (2005) Combining active and semi-supervised learning for spoken language understanding. Speech Commun 45:171–186. https://doi.org/10.1016/J.SPECOM.2004.08.002
43. Yu C, Hansen JHL (2017) Active learning based constrained clustering for speaker diarization. IEEE/ACM Trans Audio Speech Lang Process 25:2188–2198

44. Gender Recognition by Voice | Kaggle. https://www.kaggle.com/primaryobjects/voicegender
45. Cummins F, Grimaldi M, Leonard T, Simko J (2006) The CHAINS Speech Corpus: CHAracterizing INdividual Speakers. In: Proc SPECOM, pp 1–6
46. Wang J-C, Wang C-Y, Chin Y-H, Liu Y-T, Chen E-T, Chang P-C (2017) Spectral-temporal receptive fields and MFCC balanced feature extraction for robust speaker recognition. Multimed Tools Appl 76:4055–4068. https://doi.org/10.1007/s11042-016-3335-0
47. Karlos S, Fazakis N, Karanikola K, Kotsiantis S, Sgarbas K (2016) Speech recognition combining MFCCs and image features. In: Speech and Computer. SPECOM 2016, LNCS (LNAI). Springer, Cham, pp 651–658
48. Chatzichristofis SA, Boutalis YS (2008) FCTH: Fuzzy color and texture histogram—a low level feature for accurate image retrieval. In: 2008 ninth international workshop on image analysis for multimedia interactive services. IEEE, pp 191–196
49. Klaylat S, Osman Z, Zantout R, Hamandi L (2018) Arabic Natural Audio Dataset, v1. In: Mendeley Data. https://data.mendeley.com/datasets/xm232yxf7t/1
50. Karlos S, Kanas VG, Aridas C, Fazakis N, Kotsiantis S (2019) Combining active learning with self-train algorithm for classification of multimodal problems. In: 10th international conference on information, intelligence, systems and applications (IISA). IEEE, pp 1–8
51. Qin Y, Langari R, Wang Z, Xiang C, Dong M (2017) Road excitation classification for semi-active suspension system with deep neural networks. J Intell Fuzzy Syst 33:1907–1918. https://doi.org/10.3233/JIFS-161860
52. Demiröz G, Güvenir HA (1997) Classification by voting feature intervals. Springer, Berlin, Heidelberg, pp 85–92
53. Geurts P, Ernst D, Wehenkel L (2006) Extremely randomized trees. Mach Learn 63:3–42. https://doi.org/10.1007/s10994-006-6226-1
54. Breiman L (2001) Random forests. Mach Learn 45:5–32. https://doi.org/10.1023/A:1010933404324
55. He K, Zhang X, Ren S, Sun J (2015) Delving deep into rectifiers: surpassing human-level performance on ImageNet classification
56. Cai Y, Ji D, Cai D (2010) A KNN research paper classification method based on shared nearest neighbor. In: Proceedings of the 8th NTCIR Work Meet Eval Inf Access Technol Inf Retrieval, Quest Answering Cross-Lingual Inf Access, pp 336–340
57. Chen H, Liu W, Wang L (2016) Naive Bayesian classification of uncertain objects based on the theory of interval probability. Int J Artif Intell Tools 25:1–31. https://doi.org/10.1142/S0218213016500123
58. Aridas CK (2020) vfi: Classification by voting feature intervals in Python
59. Buitinck L, Louppe G, Blondel M, Pedregosa F, Müller AC, Grisel O, Niculae V, Prettenhofer P, Gramfort A, Grobler J, Layton R, Vanderplas J, Joly A, Holt B, Varoquaux G (2013) API design for machine learning software: experiences from the scikit-learn project
60. Saito T, Rehmsmeier M (2015) The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLoS ONE 10:1–21. https://doi.org/10.1371/journal.pone.0118432
61. Rodríguez-Fdez I, Canosa A, Mucientes M, Bugarín A (2015) STAC: a web platform for the comparison of algorithms using statistical tests. In: FUZZ-IEEE. pp 1–8
62. Hollander M, Wolfe DA, Chicken E (2013) Nonparametric statistical methods, 3rd edn. Wiley, Hoboken
63. Holzinger A (2016) Interactive machine learning for health informatics: when do we need the human-in-the-loop? Brain Inform 3:119–131. https://doi.org/10.1007/s40708-016-0042-6
64. Singh A, Nowak R, Zhu J (2008) Unlabeled data: now it helps, now it doesn't. In: Koller D, Schuurmans D, Bengio Y, Bottou L (eds.), NIPS. Curran Associates, Inc., pp 1513–1520
65. Leng Y, Xu X, Qi G (2013) Combining active learning and semi-supervised learning to construct SVM classifier. Knowl Based Syst 44:121–131. https://doi.org/10.1016/J.KNOSYS.2013.01.032
66. Reitmaier T, Calma A, Sick B (2015) Transductive active learning—a new semi-supervised learning approach based on iteratively refined generative models to capture structure in data. Inf Sci (Ny) 293:275–298. https://doi.org/10.1016/J.INS.2014.09.009
67. Batista AJL, Campello RJGB, Sander J (2016) Active semi-supervised classification based on multiple clustering hierarchies. In: DSAA. pp 11–20
68. Wang Q, Downey C, Wan L, Mansfield PA, Moreno IL (2017) Speaker Diarization with LSTM
69. I. Del Carmen Grau Garcia D. Sengupta MMGL, Nowé A (2018) Interpretable self-labeling semi-supervised classifier. In: Proceedings of the 2nd workshop on explainable artificial intelligence
70. Ioannis M, Nick B, Ioannis V, Grigorios T (2020) LionForests: local interpretation of random forests. In: Alessandro S, Luciano S, Paul L (eds.), First international workshop on new foundations for human-centered AI (NeHuAI 2020), Aachen, pp 17–24
71. Wang X, Wen J, Alam S, Jiang Z, Wu Y (2016) Semi-supervised learning combining transductive support vector machine with active learning. Neurocomputing 173:1288–1298. https://doi.org/10.1016/j.neucom.2015.08.087
72. Yan J, Song Y, Dai LR, McLoughlin I (2020) Task-Aware Mean Teacher Method for Large Scale Weakly Labeled Semi-Supervised Sound Event Detection. In: Proceedings of the ICASSP, IEEE international conference on acoustics, speech and signal processing. Institute of Electrical and Electronics Engineers Inc., pp 326–330
73. Kee S, del Castillo E, Runger G (2018) Query-by-committee improvement with diversity and density in batch active learning. Inf Sci (Ny) 454–455:401–418. https://doi.org/10.1016/j.ins.2018.05.014
74. Huang E, Pao H, Lee Y (2017) Big active learning. In: BigData. pp 94–101
75. Hsu W-N, Lin H-T (2015) Active learning by learning. In: AAAI conference on artificial intelligence, pp 2659–2665
76. Yue Y, Broder J, Kleinberg R, Joachims T (2012) The K-armed dueling bandits problem. J Comput Syst Sci 78:1538–1556. https://doi.org/10.1016/J.JCSS.2011.12.028
77. Huang S-J, Jin R, Zhou Z-H (2014) Active learning by querying informative and representative examples. IEEE Trans Pattern Anal Mach Intell 36:1936–1949

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.