Using Methods From Dimensionality Reduction for Active Learning With Low Query Budget

Alaa Tharwat and Wolfram Schenck

***Abstract*—Recently, it has been challenging to generate enough labeled data for supervised learning models from a large amount of free unlabeled data due to the high cost of the labeling process. Here, the active learning technique provides a solution by annotating a small but highly informative set of unlabeled data. This ensures high generalizability in space and improves classification performance with test data. The task is more challenging when the query budget is small, the data is imbalanced, multiple classes are present, and no predefined knowledge is available. To address these challenges, we present a novel active learner geometrically based on principal component analysis (PCA) and linear discriminant analysis (LDA). The proposed active learner consists of two phases: The PCA-inspired exploration phase, in which regions with high variances are explored, and the LDA-inspired exploitation phase, in which boundary points between classes are selected. The proposed geometric strategy improves the search capabilities of the active learner, allowing it to explore the space of minority classes even with multiple minority classes and a small query budget. Experiments on synthetic and real binary and multi-class imbalanced data show that the proposed algorithm has significant advantages over multiple known active learners.**

***Index Terms*—Active learning, dimensionality reduction, imbalanced data, LDA, PCA.**

I. Introduction

RECENTLY, the huge increase in the number of IoT devices and Internet data has increased the amount of free unlabeled data. Due to the cost and time involved in labeling, annotating all data or even a portion of it to create sufficient training data for developing machine learning applications has become a new challenge for data scientists. The active learning (AL) technique offers a solution by annotating only the most representative and informative points to obtain small but high-quality training data [1].

In the active learning technique, given a set of unlabeled data, the query strategy task is to query one or more points in order to label them and add them to the labeled/training data. Increasing the number of query points (i.e., the query budget) increases the cost and time of labeling, but may improve the overall classification performance [2]. However, due to the high labeling cost in many applications, one of the main goals of active learners, including ours, is to cover the most informative parts of the instance space with only a few labeled points (i.e., with a small query budget) [3], [4].

The main difference between active learners is the query strategy (i.e., the way a new point is queried). Some active learners search only for the most informative points; these points are expected to be around the decision boundaries between classes and they are informative but not necessary representative [5]. On the other hand, some active learners attempt to explore the instance space by covering large portions of it; these points are representative but may not adequately cover critical regions [6]. Few studies have combined both strategies to find the most informative and representative points [7].

Anyhow, most active learners need initial knowledge, such as some labeled points from each class for initial training data of the learning algorithms. In addition, many of the active learners handle data in limited scenarios, such as only binary classes or balanced data. Besides, some active learners only search for informative or representative points. Further, several active learners have not addressed the problem of imbalanced data. Another major problem is that most active learners mainly rely on machine learning (ML) algorithms, which i) need some initial knowledge, ii) may extrapolate incorrectly if the initial training data is not sufficient for the training process, leading to inaccurate or unreliable results, and iii) have some parameters that should be tuned. To fill some of these gaps, and inspired by the geometry of two of the most well-known dimensionality reduction methods: principal component analysis (PCA) and linear discriminant analysis (LDA), the proposed *dimensionality reduction-based active learner* (DimAL) geometrically searches for the most informative and representative points with a small query budget, without predefined knowledge, without relying on the use of ML models, using a parameter-free model to select high-quality training data even in the presence of imbalanced data to cover all/most minority classes.

Our model consists of two main phases (*exploration* and *exploitation*) and a short transition phase in between. The goal of the exploration phase is to explore the search space. For this purpose, many active learners query points from regions with high density. For example, in one of the clustering techniques, the nearest neighbours of cluster centers are selected [8]. To avoid selecting the same points over and over again, some active

Manuscript received 27 June 2022; revised 25 October 2023; accepted 30 January 2024. Date of publication 13 February 2024; date of current version 12 July 2024. This work was supported by Programme "Netzwerke 2021," an initiative of the Ministry of Culture and Science of the State of North Rhine-Westphalia through Project "SAIL: SustAInable Lifecycle of Intelligent SocioTechnical Systems" under Grant NW21-059B. Recommended for acceptance by X. Zhu. *(Corresponding author: Alaa Tharwat.)*

The authors are with the Center for Applied Data Science (CfADS), Hochschule Bielefeld-University of Applied Sciences and Arts, 33619 Bielefeld, Germany (e-mail: alaa.othman@hsbi.de; wolfram.schenck@hsbi.de).

Digital Object Identifier 10.1109/TKDE.2024.3365189

learners try to increase the diversity between the selected points as in [9]. This might result of outliers being queried. However, inspired by the dimensionality reduction method of PCA, the exploration phase in our model identifies the directions with the largest variances by searching for the maximum variance in multiple directions. To avoid selecting outliers, our model then tries to query only representative points in these directions. This increases the probability of covering all classes including the minority ones when the data are imbalanced.

In the exploitation phase, the goal is to explore critical/disagreement/uncertain regions between classes. For example, in state-of-the-art active learners with probabilistic learning models and binary classes, the *least confident* method queries the point where the posterior probability of being positive/negative is close to 0.5 [10]. Moreover, the *margin sampling* method queries the point that has a small margin between the first and the second most likely class labels [10]. Additionally, the *entropy* method was employed for measuring the uncertainty, and the points with high entropy will be queried [10]. In the query-by-committee (QBC) approach [11], the uncertain region was defined as the region where the maximum disagreement exists between a committee of learning models [11]. However, all these methods and many others depend mainly on ML learning algorithms that have some parameters that need to be tuned. Moreover, these learning algorithms require a sufficient amount of initial training data, which is practically not always available, and the performance of these ML algorithms is strongly influenced by the initial training data and its size. Further, some of the above methods may select similar (or identical) points, leading to redundancy in the selected labeled set. Furthermore, focusing on selecting borderline points might lead the active learner to select some outliers near the decision boundaries. Instead of this, inspired by the idea of the LDA dimensionality reduction method, the proposed active learner searches only geometrically (i.e., without ML models) for the borderline points between classes. This is an advantage of our model because it does not require any predefined knowledge or parameter tuning. This increases the adaptability of our model with new data, even with real-world challenges such as the imbalanced data problem with multi-class scenarios, which is one of the goals of our proposed active learner.

The transition phase in our model tries to combine the advantages of the exploration and the exploitation phases by iteratively decreasing the exploration power slightly while increasing the exploitation power. This strategy helps our model to select representative borderline points that are relatively far from the currently labeled points, which increases the covered area between classes.

Our pure geometrical strategy increases the adaptability of our model when dealing with different variations of the received data, such as balanced or imbalanced data, binary classes, or multi-class datasets without predefined knowledge. Moreover, our search strategy, which geometrically tracks the variations in the data, helps to find the minority classes when there are many of them. This is one of the goals of our model, since the number of studies that used multiple classes is small compared to those that used binary classes.

To evaluate the proposed model, we conducted a series of experiments using imbalanced datasets. In the first experiment, different synthetic datasets were used. Further, the performance of the proposed active learner was also tested on real imbalanced datasets with binary and multiple classes with different imbalance ratios to test how our model covers minority classes in multi-class scenarios.

The rest of the article is organized as follows: Section II explains the theoretical background of active learning techniques, including some studies related to the active learning technique. The detailed steps of the proposed model are explained in Section III. Section IV presents a set of experiments comparing the proposed active learner with different state-of-the-art active learners using different datasets and different experimental scenarios. Finally, concluding remarks and future work are presented in Section V.

II. Theoretical Background

A. Supervised, Unsupervised, and Semi-Supervised Learning

In machine learning, there are two main types of algorithms: supervised and unsupervised. In supervised learning algorithms, training data is used for learning algorithms, and this training data consists of a set of labeled data points as follows, $D_L = \{(\mathbf{x}_i, y_i)|\mathbf{x}_i \in X, y_i \in Y\}_{i=1}^{n_l}$, where D_L is the labeled/training data, n_l represents the number of labeled data points, $\mathbf{x}_i \in \mathcal{R}^m$ is one point, m is the number of attributes/features, the label for each instance is denoted by $y_i \in Y$, where $Y = \{\omega_i, \omega_2, \ldots, \omega_C\}$, ω_i represents the i^{th} class, and C indicates the number of classes. The data in unsupervised algorithms have no labels and are structured as follows, $D_U = \{\mathbf{x}_i|\mathbf{x}_i \in X\}_{i=1}^{n_u}$, where n_u is the number of unlabeled instances and D_U represents the unlabeled data.

As mentioned earlier, the labeling process is expensive and time-consuming; therefore, learning supervised learning models with fully-labeled data is challenging. A good alternative is the partially-supervised machine learning approach, which uses both the labeled and unlabeled data. In this approach, there are two main techniques. The first is the *semi-supervised* technique, which uses the unlabeled data for improving supervised learning models that are trained on the labeled data [7]. The second is the active learning technique, which includes an additional query strategy component that selects the most informative data points from the current unlabeled data for annotating them [10].

Fig. 1 illustrates the steps of the active learning technique. As shown, given a set of unlabeled data points (D_U) and based on a specific query strategy, an unlabeled point ($\langle \mathbf{x}^*, ? \rangle$) is selected to be queried by an expert; the selected unlabeled point is surrounded by a red circle. This newly selected and labeled data point is then added to the labeled data ($D_L = D_L \bigcup \langle \mathbf{x}^*, y^* \rangle$). Iteratively, the current labeled points are used to re-train the learning model to improve it.

B. AL With Imbalanced Data: State-of-The-Art

One of the biggest challenges in real-world environments is the presence of imbalanced data. This deteriorates the

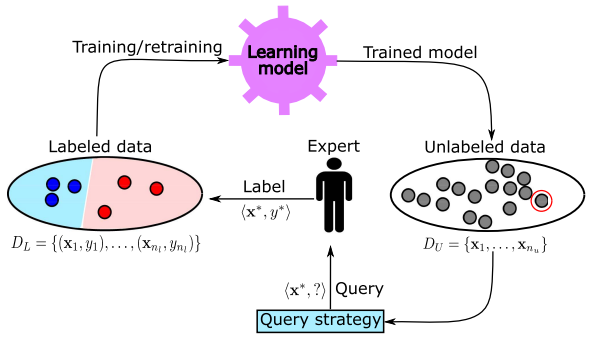

Fig. 1. Visualization of how active learners work.

performance of active and even passive learners because the model cannot learn sufficiently from the minority class, especially when the imbalance ratio is high (the imbalance ratio (IR) is the ratio between the number of majority class instances and the number of minority class instances). Therefore, with imbalanced data, the chance of querying a minority point with active learners is small compared to selecting a majority class instance. However, many active learners did not consider this problem (e.g., [12]). Some studies used classical resampling techniques to obtain balanced data. For example, in [13], only the minority data points from earlier batches were propagated, which increases the number of minority points. In [14], only the minority data points that are similar to the current batch are oversampled. The synthetic minority over-sampling technique (SMOTE) algorithm was also employed in the Learn++.CDS algorithm for balancing the data [15]. In another study, the active learning model prioritized the labeling of minority class observations for improving the balance of the learning process [16]. In addition, in [7], a new strategy was introduced to find the most informative and representative points with the aim of increasing the chance of finding and covering the minority class. Recently, new active learners were presented for handling imbalanced data with binary and multi-classes datasets [17].

In general, the number of studies that used multiple classes is small compared to those using binary classes. Further, most active learners designed for the imbalanced data problem should i) first be informed which class is the minority class, and ii) be initialized with some labeled points from the minority class. For example, in [18], in the case of binary classes, the active learner should first be initialized with a pair of opposites: one point from each class. This initial knowledge is challenging and may be impractical to collect in many environments. This is because, for example, in some real industrial scenarios, new classes of faults may appear unexpectedly. This has led us to present a novel geometry-based active learner that can handle imbalanced data with multiple classes and adapt to the newly received data without any predefined knowledge.

III. The Proposed Model

In our model, we first assume that all the available data are unlabeled, and this unlabeled data is denoted by $D_U = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n_u}\}$, where $\mathbf{x}_i \in D_U$ is one of the unlabeled points, n_u is the number of unlabeled points, and the labeled data set D_L is empty. Hence, there are neither initial labeled points nor information about the number of classes or the presence of imbalanced data. As with many active learners, there are two distinct phases in our model. The first is the exploration phase (ϕ_R), which attempts to cover the entire distribution by efficiently exploring the search space and finding the representative points by tracking the dimensions with the highest variances that represent the distribution of the data. While the aim of the second phase (exploitation phase (ϕ_L)) is to explore the regions between classes; these regions are called *critical* or *uncertain* regions. Table I lists the notations and their descriptions used in this article.

Algorithm 1: Annotate a Set of Unlabeled Points. D_L=**DimAL** $(D_U, Q.)$

Input: Unlabeled data $(D_U = \{\mathbf{x}_1, \ldots, \mathbf{x}_{n_u}\})$ and Q
Output: Labeled points $(D_L = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{n_l}, y_{n_l})\})$
Set $D_L \leftarrow [\,]$, $n_l = 0$, $n_S = 0.05 n_u$
$\mu \leftarrow \text{mean}(D_U)$; $D \leftarrow D_U - \mu$; $\Sigma \leftarrow DD^T$
Calculate $\lambda = \{\lambda_1, \ldots, \lambda_m\}$ and $\mathcal{V} = \{\mathbf{V}_1, \ldots, \mathbf{V}_m\}$ of Σ ($\mathcal{V}$ is sorted according to λ)
$MaxVar = \lambda_1$ and $Id = 1$ (index of the component or interval that has the maximum variance)
for $i = 1$ to m **do**
  $D_X \leftarrow D_U \mathbf{V}_i$
  $Lb(i) \leftarrow \min(D_X)$; $Ub(i) \leftarrow \max(D_X)$
$\Upsilon = MaxVar$ ▷ original variance
for $i = 1$ to Q **do**
  $D_X = D_U \mathbf{V}_{Id}$
  $D_{X_2} = Ub(Id) \geq D_X \geq Lb(Id)$
  $D_{X_3} = D_{X_2} \setminus Ol(D_{X_2})$, ▷ $Ol(D_{X_2})$: outliers of D_{X_2}
  $\bar{\mathbb{D}} \leftarrow \text{median}(D_{X_3})$
  if $D_L = []$ **then**, ▷ Exploration phase ($\phi_R(\mathbf{A})$)
    $\mathbf{x}^* = \text{nnSearch}(\bar{\mathbb{D}}, D_{X_3}, 1)$
  else
    if $C = 1$ **then** , ▷ Exploration phase ($\phi_R(\mathbf{B})$)
      $D_S = \text{nnSearch}(\bar{\mathbb{D}}, D_{X_3}, n_S)$
      $MD = \text{nnSearch}(D_L, D_S, 1)$
      $\mathbf{x}^* = D_S(\max(MD))$
    else ▷ Exploitation or transition phase
      if $n_l \leq \lceil 0.8Q \rceil$ & $\max(\lambda) > 0.1\Upsilon$ **then** ▷ ϕ_T
        $\eta = \frac{n_l}{Q}$
      else ▷ ϕ_L
        $\eta = 1$
      $D_S = \text{nnSearch}(\bar{\mathbb{D}}, D_{X_3}, n_S)$
      $L \leftarrow$ **CalculateADSumsEigValues**(D_L, D_S)
      $\mathcal{L} = \frac{L - \min(L)}{\max(L) - \min(L)}$, ▷ scaled $\mathcal{L}$
      $MD = \text{nnSearch}(D_L, D_S, 1)$
      $\mathcal{D} = \frac{MD - \min(MD)}{\max(MD) - \min(MD)}$ ▷ scaled MD
      $Idx = \min\,(\eta\mathcal{L} + (1 - \eta)\mathcal{D})$
      $\mathbf{x}^* \leftarrow D_S(Idx)$
  $y^* \leftarrow \text{Query}(\mathbf{x}^*)$; $D_L = D_L \bigcup (\mathbf{x}^*, y^*)$; $D_U \leftarrow D_U \setminus \mathbf{x}^*$
  Split the interval of the projected data (D_X) into two smaller intervals ($[Lb(Id), Ub(Id)]$ into $[Lb(Id), \bar{\mathbb{D}}]and(\bar{\mathbb{D}}, Ub(Id)]$)
  Replace the current interval of the projected data with the new ones
  Calculate the weighted variance of the new intervals
  Update $MaxVar$ and Id with the maximum variance

TABLE I
NOTATIONS AND DESCRIPTIONS USED IN THIS ARTICLE

Notation	Meaning	Notation	Meaning
D_L	Labeled data	D_U	Unlabeled data
n_l	No. of labeled points	n_u	No. of unlabeled points
$\mathbf{x}_i$	i^{th} data point	y_i	Label of $\mathbf{x}_i$
Q	Query budget	m	No. of dimensions of the instance space
D_S	The set of selected points	C	No. of classes
ω_i	The i^{th} class	W	Transformation matrix
n_S	No. selected points in D_S	S_B	Between-class variance
α	Regularization parameter	$\lambda = \{\lambda_1, \ldots, \lambda_m\}$	Eigenvalues of Σ
$S_W = \{S_{W_i}\}_{i=1}^{C}$	Within-class variance	X	Input space
Y	Set of outcomes	$\mathcal{V} = \{\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_m\}$	Eigenvectors of Σ
D_X	Projected data	Σ	Covariance matrix
$D = \{d_1, \ldots, d_{n_u}\}$	Centered data	$d_i = \mathbf{x}_i - \mu$	Centred data of $\mathbf{x}_i$
μ	Total mean of D_U	μ_i	The mean of the i^{th} class
$\mathbf{x}^*$	The new selected/queried point	Υ	Original variance
$L = \{l_1, l_2, \ldots, \}$	The absolute difference between sums of eigenvalues	y^*	The label for the new selected point ($\mathbf{x}^*$)
η	The weight of $\mathcal{L}$ (see (3))	$\mathcal{L}$	Scaled values of L
$MD = \{md_1, md_2, \ldots, \}$	Minimum distance between a point in D_S and all points in D_L	$\mathcal{D}$	Scaled values of MD
S_W^{-1}	Inverse of S_W	n_i	The number of points of the i^{th} class

A. PCA-Inspired Exploration Phase (ϕ_R)

1) Tracing the Directions of the Greatest Variances: The proposed model explores the instance space by continuously exploring the data in the directions of maximum/largest variances in the given unlabeled data. Inspired by PCA [19], the most well-known unsupervised dimensionality reduction method, these directions could be found by calculating the eigenvalues and eigenvectors of the covariance matrix of the unlabeled data. The direction of the eigenvector with the maximum variance (i.e., maximum eigenvalue) represents the direction of the largest variance.

However, our PCA-inspired exploration phase consists of two different strategies $\phi_R(\mathbf{A})$ and $\phi_R(\mathbf{B})$ as shown in Fig. 3(a) and (b). More details are in the next sections.

2) Theoretical Background of PCA: Mathematically, the covariance matrix ($\Sigma \in \mathcal{R}^{m \times m}$) of the current unlabeled data ($D_U \in \mathcal{R}^{n_u \times m}$) is calculated as follows, $\Sigma = DD^T$, where $D = \{d_1, d_2, \ldots, d_{n_u}\}$ is the data that is centred around the zero point by subtracting the mean value from each data point as follows, $d_i = \mathbf{x}_i - \mu$, $\mathbf{x}_i \in \mathcal{R}^{1 \times m}$ is the i^{th} unlabeled point, and $\mu \in R^{1 \times m}$ is the mean of D_U. The eigenvalues and eigenvectors of Σ are calculated as follows, $\mathcal{V}\Sigma = \lambda\mathcal{V}$, where $\lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_m\}$ represents the eigenvalues (magnitude) of the eigenvectors ($\mathcal{V} = \{\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_m\}$), the eigenvectors are sorted according to the corresponding eigenvalues, and the first eigenvector ($\mathbf{V}_1$) has the maximum eigenvalue (i.e., it has the maximum variance). Since λ_i is the amount of variance of $\mathbf{V}_i$, we also denote it as σ_i^2 (see Fig. 2).

3) Searching for the First Point ($\phi_R(\boldsymbol{A})$): Instead of randomly selecting the first point(s) as in [7], after finding the dimension that has the maximum variance (the first principal component) using PCA, the first exploration strategy of our model begins by exploring this dimension by first projecting the data (unlabeled data) onto this component/dimension (i.e., $\mathbf{V}_1$) as follows, $D_X = D_U\mathbf{V}_1$, where $D_X \in \mathcal{R}^{n_u \times 1}$ is the projected data (see Fig. 2). Since the data points in many datasets are mostly concentrated near the mean or median of the whole data, after projecting the data onto $\mathbf{V}_1$, our model will annotate the closest point to the median of the projected data after ignoring (but not removing) the outliers (see Fig. 3 ($\phi_R(\mathbf{A})$)). In our model, the outlier points should have more than 1.5 interquartile ranges above/below the upper/lower quartile. The maximum eigenvalue, which represents the variance of the first principal component is called the *original variance* and it is denoted by Υ. Depending on the value of Υ, the model may use one of the two phases (this point will be discussed later).

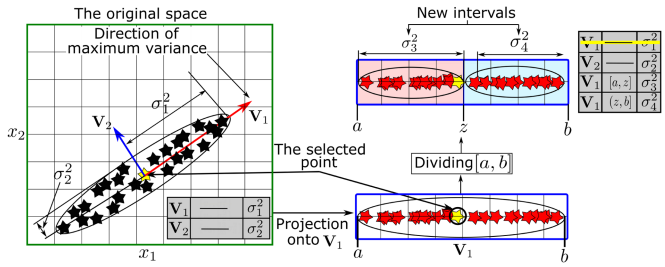

Fig. 2. Illustrative example of how our model explores the instance space by tracking variations within the data. Left: The original data is in two-dimensional space, and the PCA method is then used to find the principal components $\mathbf{V}_1$ and $\mathbf{V}_2$, where the variance (eigenvalue) of $\mathbf{V}_1$ is higher than $\mathbf{V}_2$; therefore, $\mathbf{V}_1$ is the selected dimension to be explored. Right: the original data is projected onto the selected dimension $\mathbf{V}_1$, after ignoring outliers, and the point closest to the median of the projected data is annotated. The full interval of the original dimension ($\mathbf{V}_1$) is denoted by $[a, b]$ and it will be divided into two new non-overlapping intervals ($[a, z]$ and $(z, b]$), both of which lie in the same direction as the original dimension ($\mathbf{V}_1$), but each has a different range.

In the exploration phase of our model, since the annotation of a point means that the part of the space containing that point has already been scanned or explored, and to avoid exploring the same region repeatedly, this part of the space will be divided into smaller parts/regions. This could be done as follows: i) after projecting the data onto the vector with the maximum variance, ii) finding the median of the projected data; this median represents the splitting point, iii) split the interval (or full interval) or the entire range of projected data into two smaller intervals, and iv) replace the old interval with the new smaller ones. As shown in Fig. 2, there are two principal components ($\mathbf{V}_1$ and $\mathbf{V}_2$), and surely, $\mathbf{V}_1$ (first principal component) has more variance than $\mathbf{V}_2$ ($\sigma_1^2 > \sigma_2^2$); therefore, $\mathbf{V}_1$ will be selected. The full interval of the projected data onto $\mathbf{V}_1$ is $[a, b]$, and the median of the projected data (as shown in Fig. 2) will be used for splitting the full-interval into two non-overlapping intervals: the first is $[a, z]$

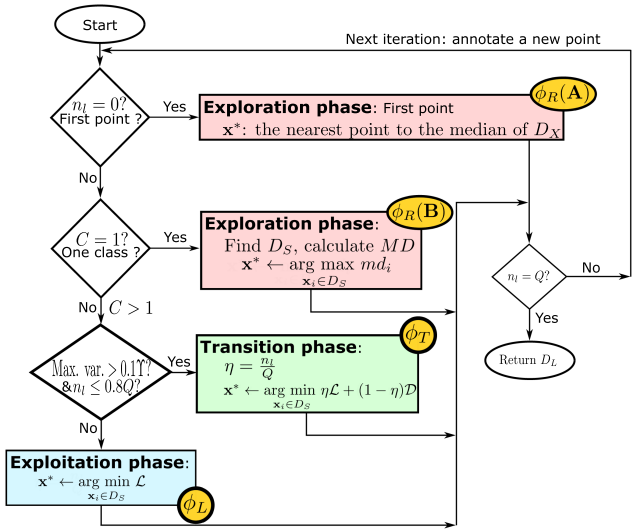

Fig. 3. Subsequent phases of the proposed active learner which differ in the way point $\mathbf{x}^*$ is determined to be labeled next.

Algorithm 2: Calculate the Absolute Differences Between the Sums of the Eigenvalues. L=**CalculateADSumsEigValues** (D_L, D_S).

Input: $D_L = \{(\mathbf{x}_1, y_1) \ldots, (\mathbf{x}_{n_l}, y_{n_l})\}$ and $D_S = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n_S}\}$
Output: L absolute differences between the sums of eigenvalues of all points in D_S

$C \leftarrow$ Unique classes in D_L
for $i = 1 : n_S$ **do**
  $NewData \leftarrow D_L \bigcup D_S(i)$
  for $j = 1 : C$ **do**
    $NewLabels \leftarrow Y \bigcup j$
    Calculate μ_j, S_{W_j} and S_{B_j} for each class
    $W = (S_W)^{-1} S_B$
    Calculate λ and V of W
    $S(j) = \sum_{o=1}^{m} \lambda_o$ ▷ This only for two classes
  $l_i \leftarrow |S(1) - S(2)|$

and the second is $(z, b]$. Hence, we still have the same number of vectors (or principal components), but instead of having the full interval of the projected data onto $\mathbf{V}_1$, this full interval is divided into two half intervals (see Fig. 2). Since the splitting point is the median, it is expected that the two new half-intervals will have approximately similar amounts of data.

Splitting the dimension into smaller intervals gives our model the opportunity to improve its exploration strategy by visiting new regions instead of exploring the same region multiple times. This strategy is approximately similar to the one in [7], where the model divides the space into cells and searches for the most uncertain regions/cells (with a high density of unlabeled data) that are less explored (with a small number of labeled points), without taking the variations in the data into consideration. For example, the model in [7] might recommend exploring a cell with more points and less variations than a cell with less data but with high variance, which reduces the representativeness of the selected points. To avoid this, our active learner evaluates data variance and point density. It does this by first calculating the variance (or weighted variance) of the projected data onto each dimension (or interval) within the space. The weighted variance is only calculated for the intervals, not for the principal components. This is because the principal components have full intervals, so all the data could be projected onto the principal components, while each interval has a range; therefore, it could have some of the projected data, but not always all of it. Mathematically, the projected unlabeled data onto $\mathbf{V}_1$ that lie within the interval $[a, z]$ is defined as follows, $D_X = \{\mathbf{x} \in D_U | a \leq \mathbf{x}\frac{\mathbf{V}_1}{|\mathbf{V}_1|} \leq z\}$. The weighted variance of this interval ($[a, z]$) is calculated by multiplying the variance D_X by the ratio of the number of points in that interval to the total number of points. Fig. 2 shows that the new intervals $[a, z]$ and $(z, b]$ have the variance σ_3^2 and σ_4^2, respectively.

In summary, during the iteration of our exploration phase, in all available directions, the full intervals and the half (or smaller) intervals compete with each other in terms of variance (or weighted variance), and the interval with the maximum variance is explored by annotating a new point there, and then that interval will be further subdivided into smaller intervals. It is worth mentioning that our active learner iteratively splits only the dimension or interval that has the maximum variance not the whole space as in [20]. This reduces the search space, which reduces the required computational time of our active learner.

4) Annotating Next Points ($\phi_R(\mathbf{B})$): After annotating the first point using $\phi_R(\boldsymbol{A})$, the second exploration strategy ($\phi_R(\mathbf{B})$) will be used to annotate new points (see Fig. 3). This strategy aims to track the variability in the data and increase the representativeness of the labeled data, if the labeled data has only one class. In this strategy, our model selects the n_S points closest to the median of the projected data onto the interval that has the maximum weighted variance (we call it simply the projected data), where n_S is the number of selected points (in our model, $n_S = 0.05n_u$), and the selected points are denoted by D_S. These selected points are expected to be highly representative because they are close to the median of the projected data. For each point in D_S, the minimum distance to the current labeled data will be calculated and denoted by $MD = \{md_1, md_2, \ldots, md_{n_S}\}$, where md_i is the minimum distance between the point $\mathbf{x}_i \in D_S$ and all points in D_L. The point with the maximum of these minimum distances will be annotated as follows:

$$\mathbf{x}^* \leftarrow \arg\max_{\mathbf{x}_i \in D_S} md_i = \underbrace{\arg\max}_{\mathbf{x}_i \in D_S} \left\{ \underbrace{\min}_{\mathbf{x}_j \in D_L} (\|\mathbf{x}_i - \mathbf{x}_j\|) \right\} \quad (1)$$

Fig. 4 shows an example to explain how a new point is selected among the selected points based on the distance to the labeled data. As shown, for each selected point ($\mathbf{x}_i \in D_S$), the minimum distance to the labeled data is calculated ($md_i = \underbrace{\min}_{\mathbf{x}_j \in D_L} (\|\mathbf{x}_i - \mathbf{x}_j\|)$). For all points in D_S, the point with the largest of these minimum distances will be annotated (i.e., $\mathbf{x}^* \leftarrow \arg\max_{\mathbf{x}_i \in D_S} md_i$).

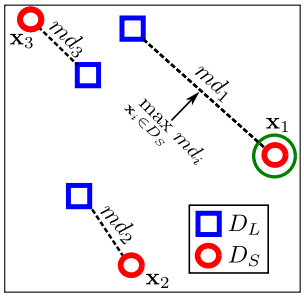

Fig. 4. Illustrative example to explain how our active learner selects, among the selected points, the point that has the maximum of the minimum distances to all labeled data. The selected point is surrounded by a green circle.

This exploration strategy ($\phi_R(\mathbf{B})$) enables our active learner to i) visit new regions that are far from D_L, thus covering a large area, and ii) explore and track variations in the data. Further, this strategy increases the chance of discovering parts of the minority classes (if any). However, this process continues until points from other classes are annotated, at which our active learner switches to one of the transition or exploitation phases. If i) the maximum variance of the current intervals is greater than 10% of the original variance[1] (10% of Υ) and ii) the number of labeled data is less than 80% of the query budget, our model switches to the transition phase; otherwise, it switches to the exploitation phase (more details are in the next section).

Finally, it is worth mentioning that in both strategies of our exploration phase in Fig. 3, i) the selected point should be one of the closest points to the median of the data after ignoring the outliers, which increases the robustness of our model against selecting an outlier, and ii) after exploring the dimension or interval with the largest variance by labeling a point in that dimension or interval, this dimension or interval is then divided into smaller intervals as mentioned before. More details about our exploration phase are in Algorithms 1 and 2.

B. LDA-Inspired Exploitation Phase (ϕ_L)

After finding points from different classes during the exploration phase, our model starts to focus on exploring the area between different classes by searching for points within uncertain regions (i.e., borderline points). For state-of-the-art active learners, this uncertain region was determined by finding the disagreement region(s) between different trained models [7], [20]. In our model, instead of using black-box ML models, our idea of finding the borderline points is inspired by the geometry underlying *Linear discriminant analysis* (LDA), one of the best-known supervised dimensionality reduction methods [21].

1) Theoretical Background of LDA: Mathematically, the goal of LDA is to reduce the dimensions of the data by projecting the original data onto a lower dimensional space so that the different classes in the data are well separated in the lower-dimensional space. To do this, first the separability between different classes (this is called also the between-class variance or the spread between the classes) should be calculated as follows, $S_B = \sum_{i=1}^{C} n_i S_{B_i}$, where n_i is the number of instances in the i^{th} class and S_{B_i} is the between-class variance of the i^{th} class and it is calculated as follows, $S_{B_i} = n_i(\mu_i - \mu)^T(\mu_i - \mu)$, where μ_i is the mean of the i^{th} class and μ is the total mean of all data. The second step is to calculate the within-class variance (this represents the distance between the mean and the samples of each class or the spread of data within each class), and it is calculated as follows, $S_W = \sum_{i=1}^{C} S_{W_i}$, where $S_{W_i} = \sum_{x_j \in \omega_i}(x_j - \mu_i)(x_j - \mu_i)^T$. For increasing the separation between different classes, LDA simply searches for a low-dimensional subspace that i) increases the between-class variance (S_B) that keeps the data points from different classes far apart, and ii) reduces the within-class variance (S_W) to keep the data points from the same class as close as possible. This could be done by projecting the data onto the line (or the subspace) having direction $\mathcal{V}$ which maximizes the ratio of S_B to S_W; hence, the objective function of LDA will be $J(\mathcal{V}) = \frac{\mathcal{V}^T S_B \mathcal{V}}{\mathcal{V}^T S_W \mathcal{V}}$. This objective function is optimized by solving the generalized eigenvalue problem as follows, $((S_W)^{-1} S_B)\mathcal{V} = \lambda \mathcal{V}$, where the eigenvalues represent the discriminative power of each linear discriminant and $\mathcal{V}$ is the eigenvectors that have the top k eigenvalues, where k is the number of desired discriminants (usually, k is less than or equal to the number of classes minus one, as a rule of thumb) [21]. In PCA, eigenvectors or principal components are new axes in the feature space that capture the directions of maximum variance in the data, while in LDA, eigenvectors are used to compute linear discriminants (i.e., linear combinations of the original features) that maximize the separation between different classes in the data and minimize the distances between the instances of the same class. The geometry behind the LDA technique will be used for finding the disagreement regions between different classes by calculating the uncertainty of unlabeled data points.

2) Exploitation Phase Vs. Transition Phase: In our active learner, there is only a small portion of labeled data that was annotated in the exploration phase, and the rest of the data is still unlabeled. Using this small amount of labeled data, LDA is applied to each unlabeled point to check the differences in the assignment of that point to the currently explored classes. These calculated differences are intended to represent the uncertainty of these unlabeled points. In this step, our strategy does not check all unlabeled data, which is very computationally expensive, but follows the same exploration strategy that tracks the variance of the data. Therefore, in our exploitation phase, we will check each unlabeled point in D_S, where D_S, as mentioned before, represents the closest n_S data points to the median of the projected data onto the interval with the maximum variance. After each annotation in the exploitation phase, the selected interval will be divided as in the exploration phase.

Assuming that we have two classes from the exploration phase (ω_1 and ω_2), for each unlabeled data point $\mathbf{x} \in D_S$, the model assumes that the point belongs to

[1] After some initial experimentation, we found that 10% of Υ is a limit after which our active learner can proceed to the next step (i.e., the transition phase). This limit provides some guarantee that our active learner has acquired the most representative points within the space before starting the transition phase (ϕ_T) and then the exploitation phase.

1) the first class (i.e. $\mathbf{x} \in \omega_1$), and then calculates $((S_W)^{-1} S_B)\mathcal{V}^{(1)} = \lambda^{(1)}\mathcal{V}^{(1)}$, and
2) the second class (i.e., $\mathbf{x} \in \omega_2$), and similarly calculates $\lambda^{(2)}$ and $\mathcal{V}^{(2)}$,

where $\lambda^{(i)}$ and $\mathcal{V}^{(i)}$ are the eigenvalues and eigenvectors when $\mathbf{x} \in \omega_i$.

The borderline points are expected to have a small absolute difference between the sums of the eigenvalues in the two cases (i.e., whether the point belongs to i) the first class, or ii) the second class), and for the point $\mathbf{x}_i$, this is defined as follows:

$$l_i = \left| \sum_{j=1}^{m} \lambda_j^{(1)} - \sum_{j=1}^{m} \lambda_j^{(2)} \right| \tag{2}$$

This pure exploitation phase may select points that are close to the currently labeled points. This reduces the area covered by our model, which reduces the representativeness of the selected points. To address this issue, our model introduces a smooth transition phase (ϕ_T) between the two phases (i.e., the exploration phase and the exploitation phase). In this transition phase, our model combines the scaled MD (i.e., scaled minimum distances to D_L) and the scaled $L = \{l_1, l_2, \ldots, l_{n_S}\}$ into a single objective function as follows:

$$\begin{aligned} \mathbf{x}^* \leftarrow \arg\min_{\mathbf{x}_i \in D_S} \eta\mathcal{L} + (1-\eta)\mathcal{D} \\ \text{subject to: } 0 \leq \eta \leq 1 \end{aligned} \tag{3}$$

where

- $\mathcal{D}$ is the normalized/scaled MD, $\mathcal{D} = \frac{MD - \min(MD)}{\max(MD) - \min(MD)}$, $MD = \{md_1, md_2, \ldots\}$, where md_i is the minimum distance between the point $\mathbf{x}_i \in D_S$, and all points in D_L.
- $\mathcal{L}$ is the scaled L and it is calculated as follows, $\mathcal{L} = \frac{L - \min(L)}{\max(L) - \min(L)}$, where $L = \{l_1, l_2, \ldots\}$ and l_i is calculated as in (2).
- η is the weight of $\mathcal{L}$, and it is calculated as follows:

$$\eta = \begin{cases} \frac{n_l}{Q} & \text{if } n_l \leq \lceil 0.8Q \rceil \\ 1 & \text{otherwise} \end{cases} \tag{4}$$

where Q is the query budget and $\lceil x \rceil$ is the ceiling function of x.

The objective function in (3) forms a smooth transition between the two phases of our active learner. As illustrated in (4), when $n_l \leq \lceil 0.8Q \rceil$, the value of η will be $\frac{n_l}{Q}$. This means that when the number of annotated points is less than or equal to 80% of the query budget, the model will consider the distances between the selected unlabeled points and the current labeled points, and the selected point should be relatively far from the current labeled points (see Fig. 3). While when $n_l > \lceil 0.8Q \rceil$, the model uses only the exploitation phase for selecting a new point to be queried without considering the distance to the current labeled points (see Fig. 3). In our initial experiments, especially with imbalanced datasets, we tested our active learner with different datasets to determine the limit of the query budget above which we can only use the exploitation phase, and we found that this limit is approximately $\lceil 0.8Q \rceil$, as illustrated in (4). This means that 80% of the query budget is used for the exploration of the space and the transition phase, which gives our active learner the opportunity to collect enough data, especially from minority classes, before moving to the exploitation phase. It is expected that this will improve the efficiency of the exploitation phase, since with enough and representative labeled data, the uncertain regions to be explored can be accurately determined with a reasonable level of certainty.

TABLE II
ABSOLUTE DIFFERENCE BETWEEN THE TWO SUMS OF THE EIGENVALUES OF THE POINTS IN FIG. 5

The point ($\mathbf{x}_i$)	$\sum_{i=1}^{m} \lambda_i$		L
	$\mathbf{x}_i \in \omega_1$	$\mathbf{x}_i \in \omega_2$	
A	20.0	0.313	19.687
B	12.8	2.45	10.35
C	5.0	5.0	0
D	2.45	12.8	10.35
E	1.78	68.6	66.82
F	0.313	20.0	19.687

The first column contains the points, the second and third columns contain the sums of the eigenvalues if the points in the first column belong to the first and second class, respectively, and the last column contains the absolute difference between the two sums.

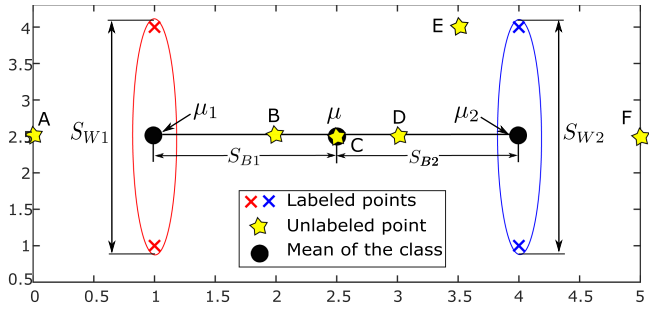

Fig. 5. Illustrative example of how our model searches for highly informative points by finding the points that have a small absolute difference between the sums of the eigenvalues (i.e., a small l_i) (see Table II). The point **C** has the minimum l_i; hence, it is the most uncertain and it will be selected by our model to be queried.

Fig. 4 shows an example to explain how the borderline points are detected. As shown, there are six unlabeled points (**A, B, C, D, E**, and **F**), two classes with two labeled points each. For each unlabeled point, we calculate the sum of the eigenvalues of $(S_W)^{-1}S_B)$ when the points belong to the first or second class, and the absolute differences between them; these values are illustrated in Table II. For example, as illustrated, the sum of the eigenvalues ($\sum_{i=1}^{m} \lambda_i$) when the point **A** belongs to i) the first class ($\mathbf{A} \in \omega_1$) is 20.0 or ii) the second class ($\mathbf{A} \in \omega_2$) is 0.313. Hence, we will have two different sums. Mathematically, and as illustrated in Fig. 5 and Table II, the absolute difference between the two sums for the borderline points between the two classes (the points **B**, **C**, and **D**) is small, and the smallest difference is for the point **C**, which is exactly in the middle between the two classes. For the points that are significantly closer to one class than the other(s), such as the points **A**, **F**, and **E**, the absolute difference between the two sums is large, and as shown, the class with the lowest sum is the farthest one from the point. For example, for the point **A**, the value of $\sum_{i=1}^{m} \lambda_i$ is

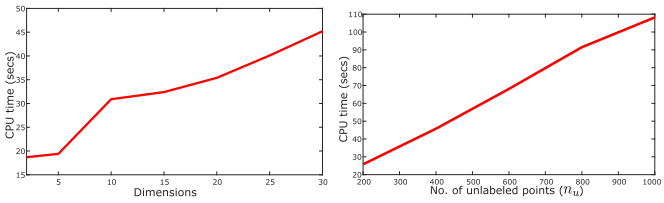

Fig. 6. Required computational time of the proposed active learner in terms of CPU time (secs) using (left) different dimensions and (right) different sizes of unlabeled data.

higher when assigning the point to the first class (closest class) than if it is assigned to the second class (farthest class). Another clear example is the point **E**, the first assumption is that the point **E** belongs to ω_1. This assumption decreases S_B because the two classes will be very close to each other and increases S_{W_1} (because this increases the covered area of ω_1). As a result, this assumption decreases the ratio between S_B and S_W and consequently decreases the sum of the eigenvalues (as shown in the table). The other assumption is that the point **E** belongs to ω_2. This keeps S_B with a large value and slightly increases S_{W_2}; which makes the ratio between S_B and S_W higher; consequently, increases the sum of the eigenvalues as illustrated in Table II.

C. Model Complexity

Our model depends mainly on the idea of the PCA and LDA methods, therefore the complexity of our model is highly affected by the steps of these methods. As illustrated in Algorithm 1, the proposed model starts by calculating the covariance matrix and the eigenvalue calculations. The complexity of each of these two steps are $O(m^2 n_u)$ and $O(m^3)$, respectively. The complexity of the next steps is very simple (e.g., sorting the eigenvalues, data projection, and calculating the median of the data); therefore, these steps won't considerably affect the complexity of our active learner. However, for calculating the absolute difference between the sums of the eigenvalues, our model should calculate S_W^{-1}, and the eigenvalues and eigenvectors of $S_W^{-1} S_B$; and the complexity of each step will be $O(m^3)$ (more details are in [21]). Therefore, our model is highly affected by the number of the dimensions of the unlabeled data and the number of unlabeled points. Using synthetic data, Fig. 6 shows the required computational time of our active learner using different numbers of dimensions and different sizes of unlabeled data. As shown, increasing the dimensions of the data or the number of unlabeled points increases the required computational time, which agrees with our analysis.

D. Some Practical considerations

- One of the main problems of LDA is that it fails to find a suitable subspace when the classes are not linearly separable. In other words, this method fails when the discriminatory information is not in the mean values of the classes, but in the variance of the data. Let us assume mathematically that in our model the mean values of two classes are approximately equal, so that the value of S_B will be zero; therefore, we could not calculate L that is used to find the most informative points in the exploitation phase. This problem could be solved by the transformation concept using one of the kernel methods that transforms the original data into a higher-dimensional space, and the data can be easily and linearly separated [22].
- The small sample size (SSS) problem may occur in LDA when the number of data points is small compared to the number of dimensions of the data. This leads to a singular S_W. As reported in [23], there are many solutions to solve this problem. In our model, we solved this problem using the regularization method by adding the identity matrix (I) that is scaled by a regularization parameter (α) to S_W as follows, $S_W = S_W + \alpha I$ [21].
- In our active learner, if the new annotated point has some identical points in D_U, we remove all these identical points from D_U to avoid wasting the query budget for annotating the same position in the space many times. This is also important because it reduces the number of unlabeled points in the explored positions in space.
- In some real-world scenarios, the low query budget may not be sufficient to find all minority classes, especially if there are many minority classes with small sizes. This was clear in our experiments where some minority classes in the multi-class datasets were not covered (i.e., some classes were missed). Consequently, the trained model with the annotated data using the active learner will assign some test data to wrong (but near) classes. A clear example here is with the streaming data, where new classes may appear (e.g., a new type of faulty data) and the training data does not contain any data from these new classes. Thus, in practice, the training data may not cover all classes [24]. Therefore, in our experiments, we tested all active learners with the worst-case scenario: when the query budget is low. However, increasing the query budget (if possible) increases the guarantee of better coverage of the space and finding most classes, including the minority classes. This was clear in our experiment when we increased the query budget from 5% to 10% of the total unlabeled data; as a result, more minority points were annotated and the number of minority classes detected increased.
- Although the median of the data is less affected by outliers than other measures of central tendency, our model ignores the outliers when calculating the median to avoid any influence of the outliers on the calculated median that might deviate our active learner from selecting representative points.
- Based on the unlabeled data, our model may use all or some of the previously mentioned phases. For example, in some cases, our model may only find data points from one class; therefore, it may only use the exploration phase and repeatedly iterating in the exploration phase may reduce the likelihood of using the transition phase; instead, our model jumps directly from the exploration phase to the exploitation phase.
- Since no machine learning is involved in the active learner's query strategy, we do not need mini-batches at all. Instead, our active learner queries one point at a time; this increases the required computation time since the learning models need to be re-trained after each newly annotated point.

However, if there is a need for mini-batches, we can simply run the active learner for a couple of iterations (each with single data points) until the next mini-batch is full.

IV. Experimental Results

In this section, we conducted a set of experiments to demonstrate the performance of the proposed active learner on synthetic/artificial and real datasets with different sizes, different numbers of classes, and different imbalance ratios. The scenario of all experiments is as follows:

1) For each dataset, we assume that all instances are unlabeled data (D_U) and D_L is empty.
2) To evaluate any active learner, iteratively this active learner will annotate a point from D_U and add it to D_L.
3) The annotated/labeled points (D_L) represent the training data used to train AdaBoost learning algorithm [25], while the remaining data (i.e., $D_U \setminus D_L$) represents the test data used for evaluating the trained model.
4) In most of our experiments, a query budget of 5% of the total number of unlabeled points was annotated.
5) For evaluating the performance of different active learners, we used the accuracy (Acc) metric [26]. Since the imbalanced datasets are dominated by either negative or positive instances, the measurement of sensitivity (Sen) and specificity ($Spec$) is of great importance. Therefore, a part of our results will be in the form of $Acc(rnk)/Sen(rnk)/Spec(rnk)$, where rnk is the rank of the model (i.e., the active learner) among all the other models. Further, in our datasets, the minority class is the positive one; therefore, the sensitivity results are expected to be lower than the specificity results. In some experiments, especially, with multi-class datasets and high IR, not all minority classes are covered, so the classification performance metrics in these experiments may not be very representative and fair. Therefore, in these experiments we will count the number of runs in which the model was unable to annotate points from all classes; we call this the number of failures (NoF) and used it as an evaluation method. In addition, because finding minority points is especially challenging when the query budget is small, we used the number of annotated points from the minority class (N^{min}) as an evaluation metric; this measure reflects how the active learner covers the minority class. For imbalanced data with multiple classes, we also counted the number of annotated points in each class to show how many classes are covered by each active learner. Further, we will calculate i) the total number of minority points (TMPs), which is the total number of annotated points from all minority classes by each active learner, and ii) the number of missed minority classes (NMMCs), which is the number of minority classes not covered by the active learner. In our experiments, we assume that the class covered by less than one point is not covered.
6) To reduce the effect of randomness in some algorithms, each experiment was repeated 51 times.

In all experiments, we compared

- the random sampling method, which iteratively selects and annotates an instance from D_U at random. This simple point selection approach is suitable for most scenarios (e.g., with or without initial data, with balanced or imbalanced data) and is therefore most commonly used to assess the performance of new active learners,
- the LLR algorithm, which tries to select the most representative instances from D_U without using initial knowledge [27]. Therefore, it fits our experimental scenarios and is expected to be suitable for problems with imbalanced data as well as for binary and multi-class scenarios,
- the A-optimal design (AOD) algorithm, which was described in [28]. AOD is one of the design of experiments (DOE) methods. Both active learning and DOE involve strategic data selection; however, DOE is broader in scope and is used in a variety of scientific and engineering disciplines. DOE methods, however, are designed to systematically explore the input space, optimize the amount of information obtained from each experiment, provide a structured and well-defined approach to data point selection, and reduce the number of experiments required without losing essential information. In addition, AOD and many other DOE methods are also suitable for different experimental scenarios that we will use in our experiments. All these advantages have led us to use one of the DOE methods in our experiments,
- the cluster-based (CB) algorithm, which was introduced in [29], where the cluster structure in the data will be exploited. By capturing the cluster structure, the active learner is guided to find highly representative points, which improves the coverage of the entire space. This makes CB ideal for dealing with imbalanced data in multi-class scenarios,
- the LHCE algorithm (it is the LHCE-III in [7]), which achieved good results with the imbalanced data. In our experiments, we used the default parameters given in [7]: 100 multilayer perceptron classifiers in each iteration that uses the exploitation phase, and for the particle swarm optimization (PSO) optimization algorithm, we used only five particles per dimension,
- The first variant of the LQBAL algorithm in [20] (LQBAL-I), which obtained promising results with imbalanced data with binary and multi-class scenarios,
- The proposed DimAL algorithm.

It is worth noting that we chose these active learners because they all i) do not require predefined knowledge, such as initial labeled points, ii) can handle problems with binary or multiple classes, iii) can obtain good results with low query budget, and iv) achieve promising covering for the whole space as mentioned before; as a result, they can handle the imbalanced data to some extent.

A. Synthetic Dataset

In this experiment, we used a set of synthetic datasets randomly distributed in two-dimensional space, where each dataset

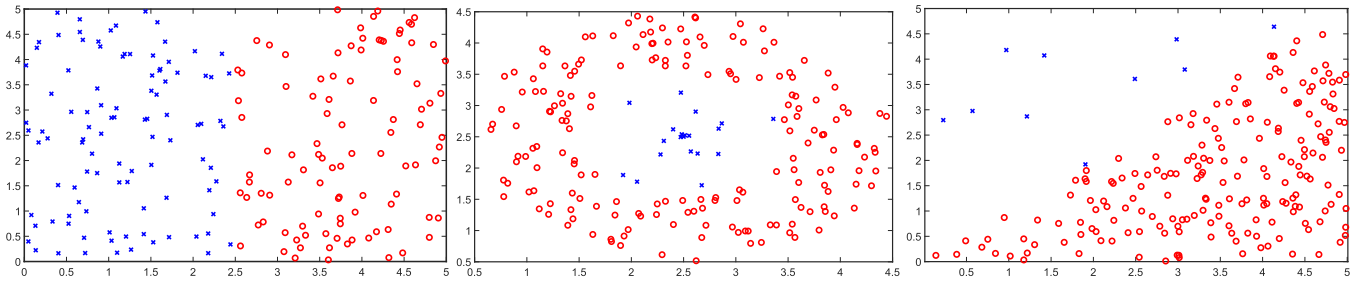

Fig. 7. Visualization of a sample of the synthetic datasets that we used in our experiments. (Left) F_1 and IR $=$ 1 : 1, (Center) F_2 and IR $=$ 9 : 1, and (Right) F_3 and IR $=$ 19 : 1.

TABLE III
COMPARISON BETWEEN THE PROPOSED MODEL (DIMAL) AND THE RANDOM, LLR, AOD, CB, LHCE, AND LQBAL MODELS USING SYNTHETIC DATA IN TERMS OF ACCURACY, SENSITIVITY, AND SPECIFICITY (IN THE FORM OF $Acc(rnk)/Sen(rnk)/Spec(rnk)$)

IR	Fn.	Random	LLR	AOD	CB	LHCE	LQBAL	DimAL
1:1	F1	90.1(6)/91.7(4)/88.5(7)	90.7(5)/85.7(6)/96.1(3)	85.7(7)/77.5(7)/94.0(6)	92.2(4)/89.6(5)/94.9(5)	95.2(3)/94.9(3)/95.6(4)	97.6(1)/97.4(1)/97.9(1)	96.3(2)/96.4(2)/96.4(2)
	F2	64.6(2)/72.1(4)/57.6(4)	62.9(5)/63.1(6)/63.3(2)	50.0(7)/0.0(7)/100.0(1)	63.9(4)/78.7(3)/49.7(5)	62.6(6)/67.1(5)/58.4(3)	65.1(1)/89.5(1)/41.4(7)	64.5(3)/83.0(2)/46.4(6)
	F3	89.7(5)/90.6(4)/88.9(6)	90.5(4)/91.0(3)/90.1(5)	85.5(7)/83.2(7)/87.8(7)	86.7(6)/83.3(6)/90.5(4)	90.8(3)/89.6(5)/92.2(2)	91.3(2)/91.7(2)/91.0(3)	93.3(1)/94.1(1)/92.6(1)
2.33:1	F1	91.6(4)/83.3(4)/95.3(5)	90.8(5)/76.4(7)/98.3(3)	89.5(6)/81.8(5)/92.7(6)	87.9(7)/81.7(6)/90.7(7)	93.7(3)/84.4(3)/97.7(4)	97.8(1)/94.3(2)/99.3(1)	97.3(2)/94.7(1)/98.4(2)
	F2	62.2(4)/50.3(2)/67.7(5)	63.5(2)/25.2(6)/80.3(2)	30.0(7)/0.0(7)/100.0(1)	58.2(6)/38.7(4)/67.0(6)	62.9(3)/40.4(3)/76.4(4)	66.0(1)/35.0(5)/79.3(3)	60.0(5)/50.9(1)/64.2(7)
	F3	86.3(6)/81.0(6)/88.7(6)	90.3(3)/87.3(1)/91.7(5)	86.3(7)/85.2(3)/86.7(7)	88.8(5)/81.9(5)/92.0(4)	90.2(4)/77.5(7)/95.8(3)	94.2(1)/83.3(4)/98.8(1)	94.0(2)/86.4(2)/97.3(2)
9:1	F1	73.5(4)/71.1(5)/73.8(6)	41.4(7)/23.9(7)/99.1(2)	94.7(2)/81.5(3)/96.0(4)	43.5(6)/84.8(1)/38.9(7)	66.2(5)/52.6(6)/99.0(3)	81.3(3)/73.9(4)/82.1(5)	98.3(1)/82.8(2)/99.9(1)
	F2	55.2(5)/42.4(2)/56.5(6)	80.2(3)/3.2(4)/93.9(4)	10.0(7)/0.0(6.5)/100.0(1.5)	44.3(6)/56.9(1)/42.8(7)	59.7(4)/0.7(5)/97.5(3)	90.1(1)/0.0(6.5)/100.0(1.5)	81.4(2)/11.6(3)/88.9(5)
	F3	68.1(5)/64.5(5)/68.4(5)	88.1(2)/56.0(6)/95.2(3)	77.1(3)/89.9(1)/75.9(4)	51.3(6)/75.8(3)/48.5(6)	71.5(4)/40.2(7)/95.8(2)	26.6(7)/89.4(2)/19.6(7)	96.0(1)/73.7(4)/98.4(1)
19:1	F1	37.6(4)/84.0(3)/35.1(5)	31.9(5)/21.4(7)/99.3(3)	90.4(2)/58.4(5)/99.3(4)	24.9(6)/87.7(2)/21.5(6)	49.2(3)/38.9(6)/99.9(2)	21.2(7)/93.5(1)/17.4(7)	98.5(1)/68.0(4)/100.0(1)
	F2	49.7(5)/50.2(2)/49.6(6)	70.2(3)/1.3(5)/97.8(4)	5.0(7)/0.0(6.5)/100.0(1.5)	19.8(6)/83.8(1)/16.4(7)	56.3(4)/1.5(4)/98.0(3)	95.3(1)/0.0(6.5)/100.0(1.5)	76.5(2)/19.6(3)/79.1(5)
	F3	38.8(5)/76.1(4)/36.7(5)	56.6(3)/27.9(6)/98.9(1)	68.7(2)/95.7(2)/67.8(4)	33.7(6)/78.3(3)/31.3(6)	41.3(4)/15.4(7)/98.4(3)	5.0(7)/100.0(1)/0.0(7)	97.0(1)/58.3(5)/98.8(2)
Avg. Rks.		4.58/3.75/5.50	3.92/5.33/3.08	5.33/5.00/3.92	5.67/3.33/5.83	3.83/5.08/3.00	2.75/3.00/3.75	1.92/2.50/2.92

Avg. Rks. (average ranks).

TABLE IV
COMPARISON BETWEEN THE PROPOSED MODEL (DIMAL) AND THE RANDOM, LLR, AOD, CB, LHCE, AND LQBAL MODELS USING SYNTHETIC DATA IN TERMS OF NoF AND NUMBER OF MINORITY POINTS (N^{min}) (IN THE FORM OF NOF/N^{min})

IR	Fn.	Random	LLR	AOD	CB	LHCE	LQBAL	DimAL
1:1	F1	0(4)/5.4(1)	0(4)/3.2(7)	0(4)/4.5(6)	0(4)/ 4.6(5)	0(4)/5.3(2)	0(4)/4.9(3)	0(4)/4.8(4)
	F2	0(3.5)/4.9(4)	0(3.5)/4.2(6)	51(7)/0.0(7)	0(3.5)/5.4(2)	0(3.5)/4.4(5)	0(3.5)/6.1(1)	0(3.5)/5.1(3)
	F3	0(3.5)/4.9(4)	0(3.5)/4.2(6)	51(7)/0.0(7)	0(3.5)/5.4(2)	0(3.5)/4.4(5)	0(3.5)/6.1(1)	0(3.5)/5.1(3)
2.33:1	F1	1(5.5)/2.8(5)	1(5.5)/1.8(7)	0(2.5)/4.3(1)	3(7)/2.7(6)	0(2.5)/3.2(4)	0(2.5)/3.6(3)	0(2.5)/3.9(2)
	F2	1(4.5)/3.3(3)	0(2)/2.6(5)	51(7)/0.0(7)	7(6)/2.2(6)	1(4.5)/2.8(4)	0(2)/3.4(2)	0(2)/3.4(1)
	F3	1(4.5)/3.3(3)	0(2)/2.6(5)	51(7)/0.0(7)	7(6)/2.2(6)	1(4.5)/2.8(4)	0(2)/3.4(2)	0(2)/3.4(1)
9:1	F1	13(5)/1.1(4)	32(7)/0.4(7)	0(1.5)/3.0(1)	31(6)/0.6(6)	6(3)/1.4(3)	9(4)/1.1(5)	0(1.5)/2.3(2)
	F2	20(5)/0.9(4)	3(3)/1.3(2)	51(7)/0.0(7)	27(6)/0.8(6)	6(4)/0.8(5)	0(1)/1.1(3)	1(2)/1.7(1)
	F3	20(5)/0.9(4)	3(3)/1.3(2)	51(7)/0.0(7)	27(6)/0.8(6)	6(4)/0.8(5)	0(1)/1.1(3)	1(2)/1.7(1)
19:1	F1	33(4)/0.5(4)	36(5)/0.3(5)	4(2)/1.9(2)	40(6)/0.3(6)	9(3)/0.9(3)	42(7)/0.2(7)	0(1)/1.9(1)
	F2	25(5)/0.6(5)	13(4)/0.9(3)	51(7)/0.0(7)	42(6)/0.2(6)	7(2)/0.6(4)	0(1)/1.0(2)	10(3)/1.2(1)
	F3	25(5)/0.6(5)	13(4)/0.9(3)	51(7)/0.0(7)	42(6)/0.2(6)	7(2)/0.6(4)	0(1)/1.0(2)	10(3)/1.2(1)
Avg. Rks.		4.54/3.83	3.88/4.83	5.50/5.50	5.50/5.25	3.38/4.00	2.71/2.83	2.50/1.75

consists of two classes of 100 data points each. In our experiments, to compare the robustness of different active learners against imbalanced data, we used four imbalance ratios for each dataset, IR$=$ 1:1, 2.33:1, 9:1, and 19:1; thus, the numbers of points for each class are 100:100, 140:60, 180:20, and 190:10, respectively. Fig. 7 shows examples of the generated data for each dataset. As shown, the data have different shapes to test the behavior of each active learner with respect to these shapes. For example, in Fig. 7 (left and right), the data are linearly separable, while the data in Fig. 7 (middle) are non-linearly separable.

Tables III and IV show the results of this experiment. From these tables, it is clear that

- For the balanced data (the first three rows in both tables), there is a strong competition between all active learners, with all active learners except AOD finding points from both classes (i.e., $NoF = 0$). In addition, the number of annotated points from both classes is balanced; therefore, among all active learners, there is not much difference between the results for accuracy, sensitivity, and specificity, except for AOD, which scored the worst. However, in terms of accuracy results, our active learner and the LQBAL model obtained the best results.
- With the imbalanced data (rows 4:12), the proposed algorithm achieved the best results in terms of NoF; this is also evident from the average ranks, where our algorithm outperformed all the other active learners and achieved the best NoF results with LQBAL. In terms of the number of minority points (N^{min}), as shown in the table, the proposed active learner clearly obtained the best results in most cases. For example, with IR 9:1 and 19:1 (i.e., high imbalance ratios) and for most of the datasets, the proposed algorithm annotated more minority points than the other algorithms. On the other hand, some active learners such as the random, CB and AOD models failed in many runs to find minority points; this is clear from the NoF and N^{min} results. For example, with IR 19:1, CB failed in most runs (approximately more than 40 out of 51 runs) to find minority points in all datasets. In terms of accuracy, sensitivity, and specificity, the average ranks show that the proposed active learner achieved the best results, while LQBAL achieved the second-best results in terms of sensitivity and accuracy.

In terms of required computational time, as shown in Table V, the proposed active learner is much faster than the other two active learners LHCE and LQBAL. Nevertheless, our active

TABLE V
COMPARISON BETWEEN THE PROPOSED MODEL AND THE RANDOM, LLR, AOD, CB, LHCE, AND LQBAL MODELS USING SYNTHETIC DATA IN TERMS OF COMPUTATION TIME IN SECS

Fn.	Random	LLR	AOD	CB	LHCE	LQBAL	DimAL
F_1	1.4	1.9	1.4	3.4	70.2	50.6	9.3
F_2	1.2	2.0	1.5	2.5	85.0	39.8	4.7
F_3	1.3	1.8	1.4	2.9	88.9	65.3	9.8

TABLE VI
DESCRIPTION OF THE REAL DATASETS

Dataset	n_u	C	m	IR (min./maj.)
Liver (BD1)	345	2	6	1.38 (145/200)
Glass0 (BD2)	214	2	5	2.06 (70/144)
Ecoli1 (BD3)	336	2	7	3.36 (77/259)
Ecoli2 (BD4)	336	2	7	5.46 (52/284)
Ecoli034vs5 (BD5)	200	2	7	9 (20/180)
Ecoli0146vs5 (BD6)	280	2	6	13 (20/260)
Ecoli4 (BD7)	336	2	7	15.8 (20/316)
Glass5 (BD8)	336	2	9	22.78 (9/205)
Ecoli0137vs26 (BD9)	281	2	7	39.14 (7/274)
Wine (MD1)	178	3	13	1.5 (59/71/48)
New-thyroid (MD2)	215	3	5	5 (30/35/150)
Balance (MD3)	625	3	4	5.88 (49/288/288)
Glass (MD4)	214	6	9	8.44 (70/76/17/13/9/29)
Ecoli (MD5)	336	8	7	71.5 (143/77/20/52/5/35/2/2)

(min./maj.) (number of minority class instances/number of majority class instances); BD. Binary class dataset; MD. multiclass dataset.

learner still requires more time than LLR, AOD, and CB, but it could be applicable in many real-world scenarios.

From the results, our model can annotate points from both classes in most cases when the data is balanced or imbalanced, even at high imbalance ratios, whereas some active learners such as AOD and CB are severely affected when the data is imbalanced and could not find minority points in many runs. Further, our model annotated significantly more minority points than the other active learners in most cases, especially at high imbalance ratios. This was challenging because the query budget is small (we used only $0.05n_u$). This good coverage of the minority class by the proposed active learner increases the accuracy and sensitivity results without affecting the specificity results. These results reflect the high exploration capabilities of the proposed active learner, even with imbalanced data.

B. Real Imbalanced Datasets

In this experiment, we used real imbalanced datasets consisting of two-class (or binary-class) and multi-class datasets [30]. As shown in Table VI, the datasets have different sizes, different numbers of classes, different dimensions, and different imbalance ratios. The binary-class datasets (BD) consist of nine datasets, each containing only two classes, while the *multi-class datasets* (MD) have different numbers of classes and some datasets have many minority classes.[2]

In our initial experiments, after some investigations, we found that some features are approximately constant in some datasets, which increases the required computational time and may deviate the active learning models from querying high-quality labeled data. Therefore, we used the PCA dimensionality reduction method [19] to reduce the dimensions and keep only the features that have 95% of the total variance. Due to the poor results of the AOD algorithm with synthetic data, we excluded it from our next experiments.

[2]All real datasets that we used in our experiments are found here: https://sci2s.ugr.es/keel/imbalanced.php.

TABLE VII
COMPARISON BETWEEN THE PROPOSED MODEL (DIMAL) AND THE RANDOM, LLR, CB, LHCE, AND LQBAL MODELS IN TERMS OF NoF AND N^{min} (IN THE FORM OF NOF/N^{min}) WITH REAL BINARY CLASS DATASETS

Ds.	Random	LLR	CB	LHCE	LQBAL	DimAL
BD1	0/7.4(3)	0/7.0(4.5)	0/7.8(2)	20/4.5(6)	0/7.0(4.5)	0/9.0(1)
BD2	0/3.7(4)	0/5.0(1.5)	0/3.4(5)	14/2.7(6)	0/5.0(1.5)	0/4.0(3)
BD3	1/2.8(3)	0/3.0(2)	4/1.9(6)	42/2.5(4)	0/2.0(5)	0/6.0(1)
BD4	1/4.0(4)	0/6.0(2)	0/2.8(6)	12/2.9(5)	0/4.0(3)	0/7.4(1)
BD5	25/0.7(6)	0/1.0(3)	18/1.0(3)	22/0.9(5)	0/1.0(3)	0/2.0(1)
BD6	16/0.9(5)	0/1.0(3)	22/0.8(6)	17/1.1(1)	0/1.0(3)	0/1.0(3)
BD7	18/1.1(2)	51/0.0(6)	35/0.4(5)	22/0.9(4)	0/1.0(3)	0/1.3(1)
BD8	21/1.0(2.5)	51/0.0(6)	43/0.2(5)	17/1.0(2.5)	0/1.0(2.5)	0/1.0(2.5)
BD9	16/1.1(1)	0/1.0(3)	39/0.3(6)	18/0.9(5)	0/1.0(3)	0/1.0(3)
Avg. Rks.	10.89/3.39	11.33/3.44	17.89/4.89	20.44/4.28	0.00/3.17	0.00/1.83
TMPs	22.7	24	18.6	17.4	23	32.7

TMPs (total number of minority points in all datasets).

1) Binary Classes Datasets: In this experiment, we used only the datasets that have two classes (the first nine datasets in Table VI). As shown, the IRs ranged from 1.38:1 to 39.14:1, and in some datasets, the minority classes have a small number of minority instances (e.g., HD5 and HD6 have only nine and seven minority points, respectively, while the majority class has 205 and 274 data points, respectively). This decreases the probability of finding a minority point with a query budget of only 5%. Therefore, we set the query budget to $\lceil \max(0.05 \times n_u, IR) \rceil$. This means that the query budget will be increased with high IR. For example, with BD9, 5% of the total number of unlabeled points is $0.05 \times 281 \approx 14.05$ and IR$=39.14$; thus, the query budget will be $\lceil \max(0.05 \times 281, IR) \rceil = \lceil \max(14.05, 39.14) \rceil \approx 40$. The results of this experiment are reported in Tables VII and VIII. From these tables, we can conclude that:

- In terms of NoFs, the proposed DimAL algorithm and the LQBAL algorithm obtained the best results, as both algorithms could find minority points in all datasets. On the other hand, out of 51 runs, the LHCE algorithm failed to find minority points in 42 and 22 runs with BD3 and BD5, respectively. Similarly, all active learners failed to find minority classes in some runs. This reflects the good search strategies of the DimAL and LQBAL algorithms, which help them to always find minority points with all datasets.
- In terms of N^{min}, from the average ranks, the proposed active learner significantly obtained the best results. As shown in Table VII, in terms of the total number of minority points in all datasets (TMPs), DimAL annotated about 32.7 minority points, while the second-best algorithm (LLR) annotated only 24 minority points and the worst algorithm (LHCE) annotated only 17.4 points. This large gap between the results of our active learner and the others shows the

TABLE VIII
COMPARISON BETWEEN THE DIMAL MODEL AND THE RANDOM, LLR, CB, LHCE, AND LQBAL MODELS IN TERMS OF ACCURACY, SENSITIVITY, AND SPECIFICITY RESULTS (IN THE FORM OF $Acc(rk)/Sen(rk)/Spec(rk)$) WITH REAL BINARY-CLASS DATASETS

Ds.	Random	LLR	CB	LHCE	LQBAL	DimAL
BD1	56.6(3)/49.3(4)/62.1(3)	49.2(6)/49.3(3)/49.2(6)	56.2(4)/53.4(2)/58.4(4)	51.1(5)/30.5(6)/76.8(1)	58.4(1)/43.5(5)/69.3(2)	58.0(2)/61.0(1)/55.8(5)
BD2	63.0(1)/57.4(2)/66.0(4)	62.6(2)/40.0(6)/73.2(2)	62.4(3)/49.0(4)/69.3(3)	54.2(6)/40.6(5)/75.1(1)	60.4(4)/53.1(3)/63.8(5)	60.1(5)/100.0(1)/40.9(6)
BD3	81.0(2)/65.2(4)/87.4(4)	76.5(6)/78.9(2)/75.7(6)	79.2(3)/52.3(5)/87.7(3)	77.1(5)/16.0(6)/91.4(1)	88.6(1)/89.8(1)/88.2(2)	78.3(4)/73.3(3)/79.7(5)
BD4	77.5(4)/56.2(3)/85.3(4)	79.3(1)/77.5(1)/79.8(5)	79.3(2)/48.9(4)/88.6(1)	64.8(6)/43.0(6)/87.9(2)	78.7(3)/47.9(5)/87.8(3)	75.5(5)/77.0(2)/75.3(6)
BD5	47.2(6)/38.8(5)/91.6(3)	92.1(1)/42.1(3)/97.7(1)	59.5(4)/40.3(4)/92.9(2)	51.6(5)/42.4(2)/90.9(4)	81.6(2.5)/26.3(6)/87.7(5)	81.6(2.5)/95.3(1)/64.8(6)
BD6	65.8(4)/26.1(4)/97.7(2)	94.4(1)/52.6(3)/97.6(3)	54.7(6)/24.9(5)/96.8(4)	63.8(5)/18.0(6)/98.0(1)	93.6(2.5)/57.9(2)/96.4(5)	93.6(2.5)/66.1(1)/47.4(6)
BD7	62.7(3)/29.9(2)/97.8(4)	6.0(6)/0.0(6)/100.0(1)	33.8(5)/10.6(5)/99.5(2)	55.8(4)/21.2(3)/98.3(3)	85.0(1.5)/21.1(4)/89.0(5)	85.0(1.5)/99.7(1)/52.5(6)
BD8	55.0(4)/23.0(2)/95.7(6)	4.2(6)/0.0(6)/100.0(1.5)	17.2(5)/5.8(4)/98.3(4)	62.6(3)/23.0(2)/96.1(5)	95.8(1.5)/23.0(2)/99.5(3)	95.8(1.5)/23.0(2)/100.0(1.5)
BD9	66.2(4)/20.0(1)/98.0(5)	97.1(1)/15.0(3)/99.6(2)	24.2(6)/3.6(6)/99.2(3)	62.8(5)/13.7(5)/98.4(4)	95.0(2.5)/15.0(3)/97.4(6)	95.0(2.5)/15.0(3)/100.0(1)
Avg. Rks.	3.44/3.00/3.89	3.33/3.67/3.06	4.22/4.33/2.89	4.89/4.56/2.44	2.17/3.44/4.00	2.94/1.67/4.72

high ability of our model to cover a large part of the space and find more minority points.

- As for the sensitivity results, the proposed active learner achieved the best results. As shown in Table VIII, DimAL achieved the best results in seven out of nine datasets and the second-best results once. This is also evident from the average ranks, where DimAL significantly outperformed the other algorithms. Further, as the imbalance ratio increases, the probability of finding minority points decreases; consequently, the sensitivity results decrease. As shown, the results of the other active learners were strongly negatively affected at high IR, while the proposed active learner achieved the best sensitivity results. This is because our active learner always successfully finds minority points even at high IR and low query budget, while the others failed to find the minority class in many cases.
- In terms of specificity results, as shown, the results of all active learners are close to each other. For example, with BD8, DimAL and LLR achieved the best results with 100%, and the random model achieved the worst results with 95.7%. These high specificity results were obtained because it is easy for all active learners to find points from the majority class, which increases the specificity results. However, from the average ranks, LHCE achieved the best results, while DimAL was the fifth-best model without a considerable difference.

2) Multi-Class Datasets: The goal of this experiment is to compare the performance of different active learners with imbalanced multi-class datasets. We used five multi-class imbalanced datasets with different numbers of classes and with IR ranging from 1.5 to 71.5. Table VI shows more details about these datasets, and as shown, in some cases the minority classes have a small number of data points. As shown in Table VI,

- with MD1, each of the first, second, and the third classes has about 33%, 40%, and 27% of the total number of data points, respectively. Hence, the third class is the minority one.
- with MD2, each of the first and the second classes has approximately 14% and 16%, respectively, and the third class has about 70% of the total number of points; therefore, the first two classes are considered as minority classes and the third class is the majority one.
- with MD3, the first class is the minority one and it has only 7.8% of the total number of points.
- with MD4, there are six classes, each of the first two classes has 32.7% and 35.5% of the total number of points, and each of the other classes has approximately, 8%, 6%, 4%, and 13.5%. This means that the last four classes are considered in our experiments as minority classes.
- with MD5, there are eight classes, each of the first, second, and fourth classes represents 42.6%, 23%, and 15.5% respectively, while the total of the other five classes represents only 19% of the total number of points. Therefore, these five classes are considered as minority classes. Among these minority classes, each of the last two classes has only 2 points (0.6%).

This big gap between the majority and minority classes in some datasets increases the challenge of finding points from minority classes, especially when the query budget is small as in our experiment. Here, we used a query budget of 5% and 10% in two separate sub-experiments. For evaluating the different active learners, we will use three different assessment metrics: i) the number of annotated points in each class, ii) TMPs, and iii) NMMCs (see the explanation of TMPs and NMMCs in the initial part of Section 4). The results of this experiment are summarized in Tables IX and X. From these tables, we can conclude the following:

- with a query budget of only 5% (see Table IX), the proposed active learner clearly achieved the best results, since it covered almost all classes, including the minority ones, while the other active learners failed in some cases to find at least one point from the minority classes. For example, with MD1, LLR surprisingly failed to find even one of the majority classes, and only points from one class were annotated, while our active learner covered all classes including the minority class. With MD4 that has four minority classes, only our active learner managed to find points from all minority classes, and it covered the minority classes much better than the other algorithms. With the MD5 dataset, the most challenging dataset with five minority classes and with high IR, DimAL clearly annotated more minority points than the other algorithms; as a result, DimAL covered minority classes better than the other active learners. It is also clear from the NMMCs results that DimAL performed the best, where DimAL only

TABLE IX
COMPARISON BETWEEN THE PROPOSED MODEL (DIMAL) AND THE RANDOM, LLR, CB, LHCE, AND LQBAL MODELS WITH IMBALANCED MULTI-CLASS REAL DATASETS, QUERY BUDGET 5%, AND IN TERMS OF THE NUMBER OF ANNOTATED POINTS FROM EACH CLASS

Ds.	Random	LLR	CB	LHCE	LQBAL	DimAL
MD1	2.9/3.9/**2.3**	9.0/0.0/**0.0**	2.8/3.3/**2.8**	2.8/3.8/**2.3**	3.2/2.9/**2.9**	6.2/1.4/**1.4**
MD2	**1.4**/**1.4**/8.2	**3.0**/**2.0**/6.0	**1.7**/**2.2**/7.1	**2.1**/**2.2**/6.7	**2.2**/**2.0**/6.8	**3.0**/**2.0**/6.0
MD3	**2.0**/15.1/14.9	**2.5**/15.2/14.3	**2.8**/14.5/14.7	**3.3**/13.6/15.2	**8.2**/12.3/11.5	**4.3**/20.8/6.9
MD4	3.6/4.1/**0.8**/**0.6**/**0.5**/**1.4**	5.0/4.0/**0.0**/**0.0**/**0.0**/**2.0**	3.6/3.6/**1.2**/**0.5**/**0.6**/**1.5**	4.3/4.3/**0.6**/**0.4**/**0.4**/**1.1**	4.7/4.0/**0.1**/**0.2**/**0.8**/**1.2**	1.0/2.0/**2.0**/**2.0**/**1.0**/**3.0**
MD5	7.2/4.1/**1.3**/2.2/**0.1**/**2.0**/**0.1**/**0.1**	6.0/6.0/**0.0**/2.0/**1.0**/**2.0**/**0.0**/**0.0**	7.0/3.6/**1.2**/3.0/**0.2**/**1.9**/**0.1**/**0.1**	8.2/3.5/**0.9**/2.3/**0.2**/**1.8**/**0.0**/**0.1**	7.1/3.1/**1.8**/2.9/**0.0**/**2.1**/**0.0**/**0.0**	3.9/6.1/**1.5**/2.9/**1.0**/**1.1**/**0.0**/**0.5**
TMPs	10.6	8.5	13.4	12.5	18.2	18.7
NMMCs	6	8	5	7	6	2

TMPs: The total number of annotated points from minority classes.
NMMCs: Number of the missed minority classes, i.e., the number of minority classes that the active learner failed to cover them.
The highlighted results with bold and underline text represent the minority class(es) results.

TABLE X
COMPARISON BETWEEN THE PROPOSED MODEL (DIMAL) AND THE RANDOM, LLR, CB, LHCE, AND LQBAL MODELS WITH IMBALANCED MULTI-CLASS REAL DATASETS, QUERY BUDGET 10%, AND IN TERMS OF THE NUMBER OF ANNOTATED POINTS FROM EACH CLASS

Ds.	Random	LLR	CB	LHCE	LQBAL	DimAL
MD1	6.1/6.8/**5.0**	15.0/0.0/**3.0**	5.4/7.6/**5.0**	2.6/4.1/**2.2**	5.2/7.8/**5.0**	13.0/2.5/**2.5**
MD2	**2.9**/**3.7**/15.5	**3.0**/**5.0**/14.0	**3.0**/**3.1**/15.9	**2.1**/**2.3**/6.7	**3.2**/**4.4**/14.3	**8.0**/**4.0**/10.0
MD3	**4.7**/29.3/29.0	**5.6**/29.9/27.6	**5.3**/27.9/29.8	**3.3**/13.5/15.1	**13.4**/22.0/27.6	**10.0**/40.0/13.0
MD4	7.0/7.8/**1.8**/**1.4**/**1.0**/**2.9**	8.0/8.0/**1.0**/**2.0**/**0.0**/**3.0**	6.9/8.5/**1.4**/**1.3**/**1.1**/**2.7**	4.2/4.3/**0.6**/**0.5**/**0.3**/**1.2**	5.1/8.8/**1.2**/**1.7**/**1.9**/**3.3**	2.0/7.0/**3.0**/**6.0**/**1.0**/**3.0**
MD5	14.7/7.7/**2.2**/5.0/**0.5**/**3.3**/**0.3**/**0.2**	13.0/8.0/**2.0**/7.0/**2.0**/**2.0**/**0.0**/**0.0**	11.2/8.4/**2.5**/6.5/**0.6**/**4.2**/**0.2**/**0.3**	8.2/3.5/**0.9**/2.3/**0.2**/**1.8**/**0.0**/**0.1**	11.2/9.0/**2.8**/6.6/**0.2**/**4.1**/**0.1**/**0.0**	5.2/18.5/**2.6**/2.8/**1.0**/**3.8**/**0.0**/**0.0**
TMPs	23.7	23.6	23.8	12.5	33.9	38.1
NMMCs	3	4	3	7	3	2

The highlighted results with bold and underline text represent the minority class(es) results.

missed two classes with all datasets, while LLR that obtained the second-best results missed five classes. Finally, the proposed active learner also obtained the best TMPs results.

- increasing the query budget to 10% of the total number of the unlabeled points increases the chance of finding more minority points, which is evident from the TMPs results. Therefore, as shown in Table X, the NMMCs results decrease (i.e., more minority classes are covered). For example, with MD4 and LLR, there are three minority classes that were not covered with a query budget of 5%, but increasing the query budget to 10% increases the covered minority classes; and thus decreases the NMMCs. This means that increasing the query budget increases the chance of finding and covering more minority classes, which might increase the classification performance. However, with TMPs=38.1 and NMMCs=2, DimAL significantly obtained the best results among all the other active learners.

In summary, due to our special search strategy, the proposed algorithm achieved promising results on imbalanced binary and multi-class datasets and was more successful in finding minority points than other active learning methods in most cases. This reflects the good query strategy of our algorithm, which helps to find minority points better than some state-of-the-art active learning algorithms. Consequently, the proposed model achieved the best sensitivity results, especially for large IR.

V. CONCLUSIONS AND FUTURE WORK

In this article, a novel dimensionality reduction-based active learning algorithm (DimAL) is presented to select the most informative and representative data points. The proposed active learner balances the selection of informative and representative points through two phases. The first is the PCA-inspired exploration phase, in which our active learner searches for the regions with high variances to explore. The second phase is the LDA-inspired exploitation phase, in which our active learner selects borderline points between classes. This strategy, based mainly on the geometric basis of two of the most popular dimensionality reduction methods (PCA and LDA), improves the ability of our model to cover large parts of the space and to scan a large part of the subspace of the minority classes when the data is imbalanced and with multi-class scenario. This purely geometric strategy, which does not depend on a machine learning model, increases the flexibility of our model to handle different variations of the received unlabeled data. Another advantage of our model is that it could obtain good results even with a low query budget. Further, our active learner does not require any predefined knowledge (e.g., the number of classes or the initial training data), which allows our model to work with different applications that have different initial knowledge. Furthermore, our active learner is parameter-free, i.e., no parameter tuning steps are required. Experimental results on synthetic and real imbalanced and balanced datasets with different numbers of classes, different imbalance ratios, and different numbers of minority classes demonstrate the effectiveness of our approach.

However, using a purely geometric strategy might still pose some problems (see some of the practical considerations in Section III-D) to increase the applicability of our active learner in different real-world settings. One of these practical considerations that we have not mentioned is high-dimensional data. From the complexity analysis in Section III-C, our active learner requires high computational power compared to other active learners (this is also evident in our experiments (see Table V)), and this required computational power is likely to be higher with high-dimensional data; this would be a fruitful area for future work to increase the applicability of not only our active learner, but also other active learners who have a similar strategy and suffer in high-dimensional spaces. One of the solutions is to

process the data not in their original spaces, but in low-dimensional spaces, without losing important information that could degrade the performance of active learners.

REFERENCES

[1] A. Krishnamurthy, A. Agarwal, T. K. Huang, H. Daumé III, and J. Langford, "Active learning for cost-sensitive classification," *J. Mach. Learn. Res.*, vol. 20, no. 65, pp. 1–50, 2019.

[2] M. Wang, K. Fu, F. Min, and X. Jia, "Active learning through label error statistical methods," *Knowl.-Based Syst.*, vol. 189, 2020, Art. no. 105140.

[3] O. Sener and S. Savarese, "Active learning for convolutional neural networks: A core-set approach," 2017, *arXiv:1708.00489*.

[4] D. Yoo and I. S. Kweon, "Learning loss for active learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 93–102.

[5] Z. Liu, H. Ding, H. Zhong, W. Li, J. Dai, and C. He, "Influence selection for active learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9274–9283.

[6] A. Tharwat and W. Schenck, "A survey on active learning: State-of-the-art, practical challenges and research directions," *Mathematics*, vol. 11, no. 4, 2023, Art. no. 820.

[7] A. Tharwat and W. Schenck, "Balancing exploration and exploitation: A novel active learner for imbalanced data," *Knowl.-Based Syst.*, vol. 210, 2020, Art. no. 106500.

[8] D. Ienco, A. Bifet, I. Žliobaitė, and B. Pfahringer, "Clustering based active learning for evolving data streams," *Int. Conf. Discov. Sci.*, 2013, pp. 79–93.

[9] Z. Xu, R. Akella, and Y. Zhang, "Incorporating diversity and density in active learning for relevance feedback," in *Proc. Eur. Conf. Inf. Retrieval*, 2007, pp. 246–257.

[10] B. Settles, "Active learning literature survey," Dept. Comput. Sci., Univ. Wisconsin-Madison, Madison, WI, USA, Tech. Rep. 1648, 2009.

[11] H.S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *Proc. 5th Annu. Workshop Comput. Learn. Theory*, 1992, pp. 287–294.

[12] Y. Baram, R. E. Yaniv, and K. Luz, "Online choice of active learning algorithms," *J. Mach. Learn. Res.*, vol. 5, pp. 255–291, 2004.

[13] J. Gao, W. Fan, J. Han, and P. S. Yu, "A general framework for mining concept-drifting data streams with skewed distributions," in *Proc. SIAM Int. Conf. Data Mining*, 2007, pp. 3–14.

[14] S. Chen and H. He, "SERA: Selectively recursive approach towards nonstationary imbalanced stream data mining," in *Proc. Int. Joint Conf. Neural Netw.*, 2009, pp. 522–529.

[15] R. Elwell and R. Polikar, "Incremental learning of concept drift in nonstationary environments," *IEEE Trans. Neural Netw.*, vol. 22, no. 10, pp. 1517–1531, Oct. 2011.

[16] Ł Korycki, A. Cano, and B. Krawczyk, "Active learning with abstaining classifiers for imbalanced drifting data streams," in *Proc. IEEE Int. Conf. Big Data*, 2019, pp. 2334–2343.

[17] W. Liu, H. Zhang, Z. Ding, Q. Liu, and C. Zhu, "A comprehensive active learning method for multiclass imbalanced data streams with concept drift," *Knowl.-Based Syst.*, vol. 215, 2021, Art. no. 106778.

[18] L. Wang, X. Hu, B. Yuan, and J. Lu, "Active learning via query synthesis and nearest neighbour search," *Neurocomputing*, vol. 147, 2015, pp. 426–434.

[19] A. Tharwat, "Principal component analysis: An overview," *Pattern Recognit.*, vol. 3, no. 3, pp. 197–240, 2016.

[20] A. Tharwat and W. Schenck, "A. novel low-query-budget active learner with pseudo-labels for imbalanced data," *Mathematics*, vol. 10, no. 7, 2022, Art. no. 1068.

[21] A. Tharwat, T. Gaber, A. Ibrahim, and A. E. Hassanien, "Linear discriminant analysis: A detailed tutorial," *AI Commun.*, vol. 30, no. 2, 2017, pp. 169–190.

[22] B. Scholkopft and K. R. Mullert, "Fisher discriminant analysis with kernels," *Neural Netw. Signal Process. IX*, vol. 1, no. 1, pp. 41–48, 1999.

[23] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Regularization studies of linear discriminant analysis in small sample size scenarios with application to face recognition," *Pattern Recognit. Lett.*, vol. 26, no. 2, pp. 181–191, 2005.

[24] Y. N. Zhu and Y. F. Li, "Semi-supervised streaming learning with emerging new labels," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 7015–7022.

[25] T. Hastie, S. Rosset, J. Zhu, and H. Zou, "Multi-class adaboost," *Statist. Interface*, vol. 2, no. 3, pp. 349–360, 2009.

[26] A. Tharwat, "Classification assessment methods," in *Appl. Comput. Inf.*, vol. 17, no. 1, pp. 168–192, 2020.

[27] L. Zhang, C. Chen, J. Bu, D. Cai, X. He, and T. S. Huang, "Active learning based on locally linear reconstruction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 10, pp. 2026–2038, Oct. 2011.

[28] A. Atkinson et al., *Optimum Experimental Designs, With SAS*. Oxford, U.K.: Oxford Univ. Press, 2007.

[29] S. Dasgupta and D. Hsu, "Hierarchical sampling for active learning," in *Proc. 25th Int. Conf. Mach. Learn.*., 2008, pp. 208–215.

[30] A. Fernández, S. García, M. J. del Jesus, and F. Herrera, "A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets," *Fuzzy Sets Syst.*, vol. 159, no. 18, 2008, pp. 2378–2398.

Alaa Tharwat received the PhD degree in the area of optimization and machine learning from Suez Canal University, Egypt, in 2017. Since 2019, he has been a postdoc for engineering informatics with the Bielefeld University of Applied Sciences. His main research interest includes machine learning with focus on active learning, optimization, and learning on data streams.

Wolfram Schenck received the doctoral degree in the area of cognitive robotics from Bielefeld University, Germany, in 2008. He is professor for engineering informatics with the Bielefeld University of Applied Sciences. His main research interest includes machine learning with focus on active learning, clustering, learning on data streams, and applied deep learning.