Multi-annotator Probabilistic Active Learning

—

Supplementary Material

Marek Herde, Daniel Kottke, Denis Huseljic, and Bernhard Sick
Intelligent Embedded Systems
University of Kassel
Germany
{marek.herde,dkottke,dhuseljic,bsick}@uni-kassel.de

I. Multi-annotator Probabilistic Active Learning

In this section, we illustrate the selection algorithm of our AL strategy *multi-annotator probabilistic active learning* (MaPAL) for a two-dimensional toy data set with four annotators having instance-dependent performance values. The corresponding pseudo-code of MaPAL's selection algorithm is given in Fig. 1.

Input: data set $\mathcal{D}$, annotators $\mathcal{A}$, similarity function s, prior number of correct annotations β_0, maximum number of modeled annotation acquisitions M_{max}
Output: query set $\mathcal{S}$

1: for each instance-annotator pair $(\mathbf{x}_n, a_m) \in \mathcal{X} \times \mathcal{A}$ compute the annotation performance value and update the weight of the data set $\mathcal{D}$ according to Eq. 19 in the main article
2: specify the set $\mathcal{R}$ consisting of pairs of instances and annotator rankings according to Eq. 20 in the main article
3: compute the utility value for each pair of instance and annotator ranking $(\mathbf{x}_n, \mathbf{a}_n) \in \mathcal{R}$ according to Eq. 21 in the main article
4: specify the best pair of instance and annotator ranking according to Eq. 22 in the main article
5: determine the query set $\mathcal{S} = \{(\mathbf{x}_{n^*}, a_{n_1^*})\}$
6: **return** $\mathcal{S}$

Fig. 1. Pseudo-code of MaPAL's selection algorithm.

In Fig. 2, we illustrate the performance estimates of our *Beta annotator model* (BAM) for different values of β_0. Each column represents one annotator, while each row shows the estimated performance values for a specific value of β_0. A decreasing value of β_0, i.e., decreasing amount of prior knowledge about the annotators, leads to stronger influence of the observed annotations on the performance estimates. As a result, providing a small amount of falsely annotations in a region can strongly decrease the performance in this region.

To show the impact of MaPAL's hyperparameter M_{max} on its selection of instance-annotator pairs, we illustrate the computed instance utility values and annotation performance values in Fig. 3. Each column represents one annotator, while each block of two row shows the computed instance utility and performance values for a specific value of M_{max} after $B = 50$ actively acquired annotations. Considering the decision boundary of the SbC using the data set $\mathcal{D}$, we observe that an increasing value of M_{max} leads to an improved selection of instance-annotator pairs.

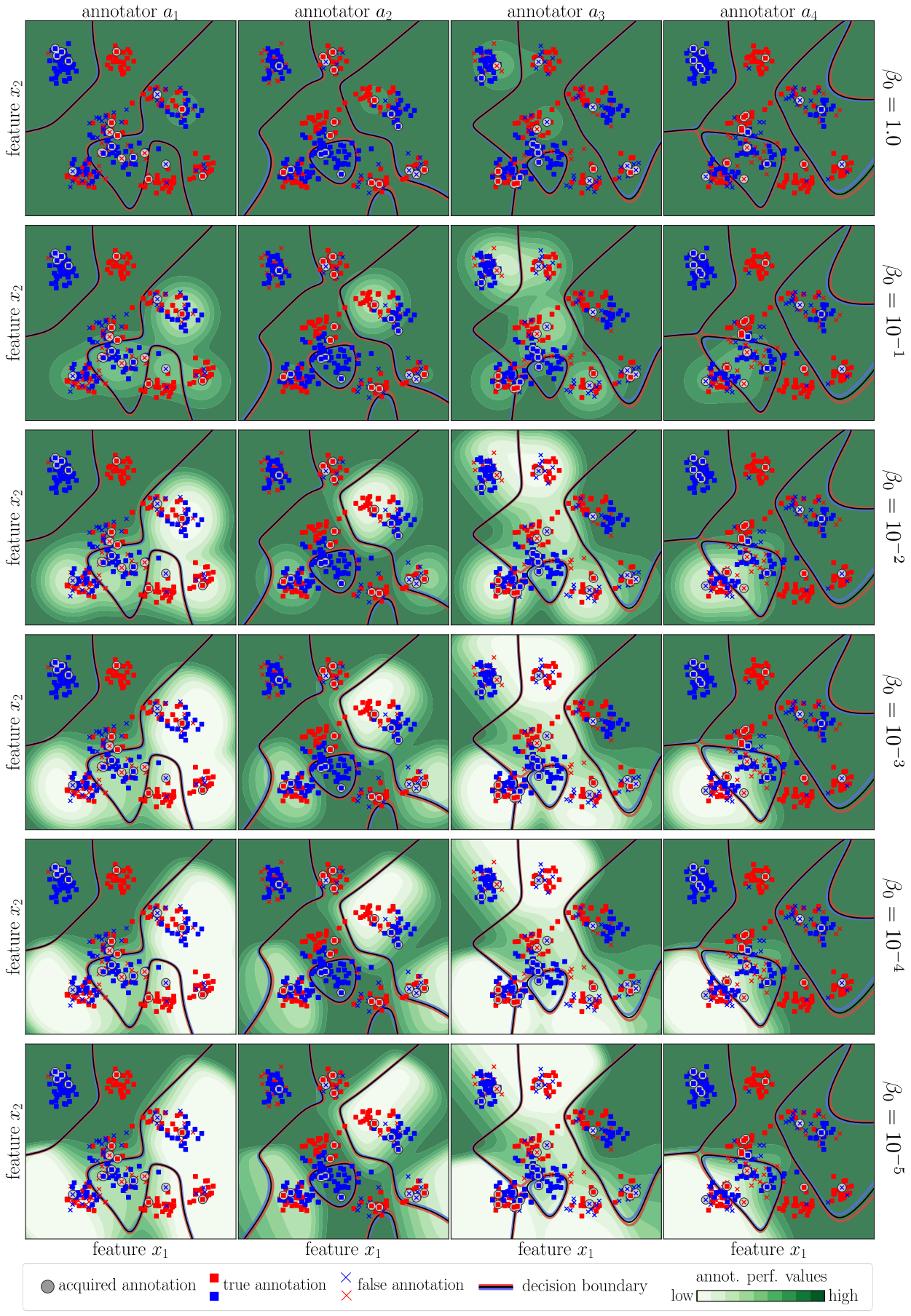


Fig. 2. Visualization of BAM: The 24 plots visualize the same two-dimensional toy data set with instances of two classes (blue vs. red). Each annotator provided annotations for 20 randomly selected instances. The green color indicates the performance values of the annotators. Each black line represents the decision boundary of the SbC using the data set $\mathcal{D}_{\overline{m}}, m \in \{1, \ldots, 4\}$. The blue and red lines show which class is predicted on which side of this decision boundary.

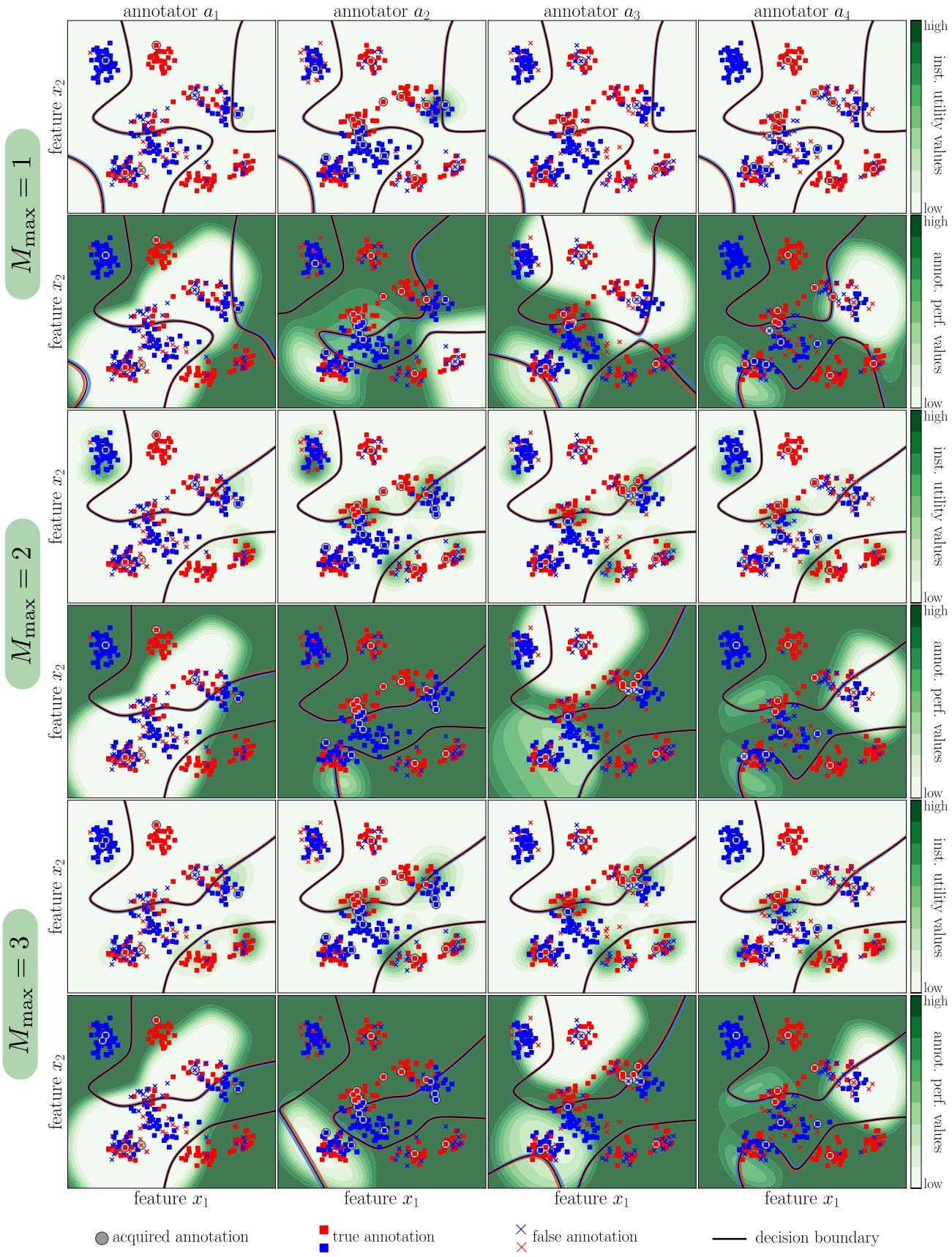


Fig. 3. Visualization of MaPAL's selection algorithm with $\beta_0 = 10^{-4}$ for three different values $M_{\max}$ after $B = 50$ actively acquired annotations: The 24 plots visualize the same two-dimensional toy data set with instances of two classes (blue vs. red). On the one hand, there are plots showing the estimated performance values according to Fig. 2. The black line of such a plot represents the decision boundary of the SbC using the data set $\mathcal{D}_{\overline{m}}, m \in \{1, \ldots, 4\}$. On the other hand, there are plots showing the instance utility values as a function of the annotators. In these plots, the black line represents the decision boundary of the SbC using the data set $\mathcal{D}$.

II. Experimental Evaluation

In this Section, we provide more details on the experimental setup. Moreover, learning curves, tables of *area under learning curve* (AULC) values, and execution times are given.

A. Data Sets

In total, we conducted an experimental evaluation on 29 data sets. Four data sets were annotated by **real-world** annotators and an overview of them is given in Table I. The data sets grid and medical were annotated by six annotators, whereas five annotations of five annotators are available for the data sets mozilla and compendium. Since we had no access to the true class labels of the instances for these data sets, we used the majority votes of all annotators as estimates of the true class labels.

TABLE I

Overview of data sets with **real-world** annotators: The table lists the main characteristics of the used data sets, namely data set name with reference, the number of features, the number of instances per class, and the estimated annotation accuracy of each annotator averaged over all instances of a data set.

data set	no. of features	no. of instances per class	estimated annotation accuracies of the annotators
medical [5]	62	71, 216	0.80, 0.84, 0.78, 0.84, 0.85, 0.71
grid [1]	10	47, 88, 124, 41, 0[1]	0.58, 0.54, 0.68, 0.34, 0.67, 0.44[2]
mozilla [6]	100	155, 60, 293, 167	0.78, 0.71, 0.64, 0.61, 0.85
compendium [6]	56	87, 315, 221, 339	0.81, 0.77, 0.67, 0.87, 0.87

In favor of a more meaningful and reliable evaluation, we additionally selected 25 popular data sets being publicly available at OpenML [11]. On these data sets, we had access to the true class labels of the instances. However, there were no annotations from multiple error-prone annotators. Therefore, we used the true class labels to **simulate** annotators with different types of annotation performances, namely uniform, class-dependent, and instance-dependent. As a result, there were three groups of four to six annotators for each of the 25 five data sets. We conducted experiments separately for each of these annotator groups. This way, we can analyze the robustness of AL strategies for different assumptions regarding the performances of annotators. A detailed description of the simulation procedures is given in Section II-B. An overview of the data sets with simulated annotators is given in Table II.

TABLE II

Overview of data sets with **simulated** annotators having *uniform* (unif) / *class-dependent* (class) / *instance-dependent* (inst) performance values: The table lists the main characteristics of the used data sets, namely data set name with OpenML identifier (ID) [11], the number of features, the number of instances per class, and the actual annotation accuracy of each annotator averaged over all instances of a data set.

data set (ID)	no. of features	no. of instances per class	actual annotation accuracies of the annotators
iris (61)	4	50, 50, 50	unif: 0.51, 0.57, 0.53, 0.63, 0.86, 0.88 class: 0.73, 0.68, 0.55, 0.59, 0.56 inst: 0.63, 0.67, 0.65, 0.69
wine (187)	13	59, 71, 48	unif: 0.38, 0.63, 0.57, 0.66 class: 0.75, 0.71, 0.54, 0.71 inst: 0.63, 0.61, 0.65, 0.55, 0.69
parkinsons (1488)	22	48, 147	unif: 0.46, 0.61, 0.78, 0.92 class: 0.83, 0.56, 0.69, 0.88, 0.95 inst: 0.65, 0.74, 0.69, 0.76, 0.72
prnn-craps (446)	7	100, 100	unif: 0.66, 0.68, 0.69, 0.77, 0.69, 0.89 class: 0.52, 0.87, 0.79, 0.62, 0.69 inst: 0.74, 0.73, 0.73, 0.81
sonar (40)	60	111, 97	unif: 0.52, 0.57, 0.66, 0.69, 0.82, 0.86 class: 0.77, 0.77, 0.61, 0.87 inst: 0.75, 0.81, 0.72, 0.80
seeds (1499)	7	70, 70, 70	unif: 0.33, 0.42, 0.56, 0.63, 0.64, 0.78 class: 0.53, 0.61, 0.49, 0.59, 0.61, 0.93 inst: 0.67, 0.58, 0.58, 0.59, 0.55, 0.71
glass (41)	9	70, 76, 13, 29, 9, 17	unif: 0.35, 0.51, 0.54, 0.88 class: 0.56, 0.42, 0.62, 0.60 inst: 0.70, 0.40, 0.44, 0.38, 0.71
vertebra-column (1523)	6	60, 100, 150	unif: 0.45, 0.62, 0.71, 0.70 class: 0.78, 0.65, 0.47, 0.63, 0.78 inst: 0.66, 0.59, 0.58, 0.48, 0.67

CONTINUED ON NEXT PAGE.

[1] There is no majority vote annotation of this class.

[2] This last number does not represent the estimated accuracy of a human annotator, but the annotations were obtained from a stochastic simulation [1].

TABLE II – CONTINUED FROM PREVIOUS PAGE.

data set (ID)	no. of features	no. of instances per class	actual annotation accuracies of the annotators
ecoli (39)	7	143, 77, 2, 2, 35, 20 5, 52	unif: 0.21, 0.57, 0.64, 0.76 class: 0.32, 0.58, 0.49, 0.42 inst: 0.74, 0.43, 0.33, 0.75
ionosphere (59)	34	126, 225	unif: 0.52, 0.74, 0.82, 0.83 class: 0.67, 0.69, 0.68, 0.84, 0.68 inst: 0.73, 0.75, 0.70, 0.77
user-knowledge (1508)	5	102, 129, 122, 24, 26	unif: 0.20, 0.36, 0.67, 0.62 class: 0.58, 0.57, 0.62, 0.72, 0.77 inst: 0.61, 0.61, 0.61, 0.59
chscase-vine (814)	2	212, 256	unif: 0.58, 0.63, 0.66, 0.79, 0.83 class: 0.75, 0.60, 0.73, 0.68, 0.73 inst: 0.69, 0.65, 0.73, 0.69, 0.70, 0.72
kc2 (1063)	21	415, 107	unif: 0.60, 0.59, 0.74, 0.70, 0.80, 0.89 class: 0.77, 0.65, 0.67, 0.96, 0.61, 0.88 inst: 0.79, 0.62, 0.69, 0.79, 0.80
wdbc (1510)	30	357, 212	unif: 0.52, 0.67, 0.71, 0.75, 0.90 class: 0.69, 0.76, 0.73, 0.72, 0.65, 0.83 inst: 0.78, 0.64, 0.64, 0.65, 0.61, 0.78
balance-scale (11)	4	49, 288, 288	unif: 0.39, 0.49, 0.55, 0.63, 0.68, 0.77 class: 0.60, 0.83, 0.55, 0.63, 0.64 inst: 0.56, 0.60, 0.62, 0.62, 0.69
blood-transfusion (1464)	4	570, 178	unif: 0.51, 0.69, 0.67, 0.72, 0.88 class: 0.77, 0.90, 0.66, 0.58, 0.76, 0.97 inst: 0.78, 0.72, 0.63, 0.69, 0.68, 0.69
pima-indians (37)	8	500, 268	unif: 0.54, 0.64, 0.65, 0.80, 0.78, 0.88 class: 0.72, 0.64, 0.72, 0.90, 0.87, 0.83 inst: 0.68, 0.63, 0.79, 0.67, 0.78, 0.72
vehicle (54)	18	218, 212, 217, 199	unif: 0.42, 0.45, 0.50, 0.68, 0.86 class: 0.67, 0.64, 0.59, 0.57 inst: 0.58, 0.71, 0.72, 0.54
qsar-biodeg (1494)	41	699, 356	unif: 0.55, 0.59, 0.67, 0.73, 0.87 class: 0.69, 0.57, 0.73, 0.65, 0.73 inst: 0.78, 0.72, 0.8 , 0.77, 0.68
banknote (1462)	4	762, 610	unif: 0.53, 0.58, 0.72, 0.76, 0.88 class: 0.71, 0.69, 0.86, 0.87 inst: 0.81, 0.74, 0.72, 0.74
steel-plates-fault (1504)	33	1268, 673	unif: 0.58, 0.68, 0.71, 0.71, 0.83 class: 0.84, 0.72, 0.89, 0.78 inst: 0.71, 0.75, 0.67, 0.72, 0.70, 0.65
segment (36)	19	330, 330, 330, 330, 330, 330, 330	unif: 0.21, 0.37, 0.48, 0.68, 0.86 class: 0.51, 0.56, 0.54, 0.41, 0.48 inst: 0.60, 0.55, 0.57, 0.65
phoneme (1489)	5	3818, 1586	unif: 0.62,0.58,0.72,0.76,0.82 class: 0.76, 0.65, 0.81, 0.73, 0.63, 0.89 inst: 0.70, 0.68, 0.71, 0.76, 0.70
satimage (182)	36	1531, 703, 1356, 625, 707, 1508	unif: 0.35, 0.38, 0.57, 0.66, 0.8, 0.78 class: 0.80, 0.51, 0.57, 0.51 inst: 0.51, 0.50, 0.58, 0.51, 0.53
wind (847)	14	3073, 3501	unif: 0.57, 0.63, 0.67, 0.73, 0.81 class: 0.76, 0.58, 0.83, 0.78, 0.68, 0.84 inst: 0.70, 0.66, 0.69, 0.75, 0.75

B. Simulation of Annotators

In the following, we describe the three different techniques for simulating annotators with different types of performances.

Uniform: We simulate an annotator a_m with uniform performance values by specifying a probability $q_m \in [1/C, 0.9]$. Given an instance $\mathbf{x}_n$, the annotator a_m provides the true class label $z_{n,m} = y_n$ with the probability q_m and provides a false annotation $z_{n,m} \in \Omega_Y \setminus \{y_n\}$ with the probability $1 - q_m$. For each data set, the number M of available annotators is randomly chosen from the set $\{4, 5, 6\}$. The exact value of q_m depends on the number M of annotators and the number C of classes. We draw q_m for each annotator a_m from a uniform distribution U according to

$$q_m \sim U\left(\frac{1}{C} + (m-1)\cdot d, \frac{1}{C} + (m+1)\cdot d\right),\ d = \frac{0.9 - \frac{1}{C}}{M+1}. \tag{1}$$

This sampling procedure ensures that no annotator is worse than random guessing and that no omniscient annotator is available. Moreover, it simulates scenarios where the performance values of the annotators are different. The application of this simulation procedure to a two-dimensional data set is illustrated by the first row of plots in Fig 4.

Class-dependent: We simulate an annotator a_m with class-dependent performance values by specifying the probability $q_{m,y} \in [1/C, 1.0]$ for each class $y \in \Omega_y$. Given an instance $\mathbf{x}_n$ of class $y_n = y$, the annotator a_m provides the true class label $z_{n,m} = y_n$ with the probability $q_{m,y}$ and provides a false annotation $z_{n,m} \in \Omega_Y \setminus \{y_n\}$ with the probability $1 - q_{m,y}$. For each data set, the number M of available annotators is randomly chosen from the set $\{4, 5, 6\}$. The exact value of $q_{m,y}$ depends on the number C of classes. We draw $q_{m,y}$ for each annotator a_m and class y from a uniform distribution U according to

$$q_{m,y} \sim U\left(\frac{1}{C}, 1\right). \tag{2}$$

This sampling procedure ensures that no annotator is worse than random guessing. The application of this simulation procedure to a two-dimensional data set is illustrated by the second row of plots in Fig 4.

Instance-dependent: To simulate annotators with instance-dependent performance values, we adopt the idea of the simulation procedure proposed in [12]. Given a data set, we perform a k-means clustering [8] on its instances. The number of clusters is given by the number of annotators ($k = M$). We randomly draw the number M of annotators from the set $\{4, 5, 6\}$. Once the clusters are determined, each annotator is seen as an "expert" on two randomly selected clusters and not familiar with the remaining $M - 2$ clusters. For each pair of cluster $c \in \{1, \ldots, M\}$ and annotator a_m, we draw a probability $q_{m,c} \in [\frac{1}{C}, 1]$ according to:

$$q_{m,c} \sim \begin{cases} U(0.8, 1.0) \text{ if annotator } a_m \text{ is expert on cluster } c, \\ U(\frac{1}{C}, \frac{1}{C} + 0.2) \text{ otherwise.} \end{cases} \tag{3}$$

For an instance $\mathbf{x}_n$ belonging to cluster c, the annotator a_m gives the correct class label $z_{n,m} = y_n$ with the probability $q_{m,c}$ and a false annotation $z_{n,m} \in \Omega_Y \setminus \{y_n\}$ with the probability $1 - q_{m,c}$. The application of this simulation procedure to a two-dimensional data set is illustrated by the third row of plots in Fig 4.

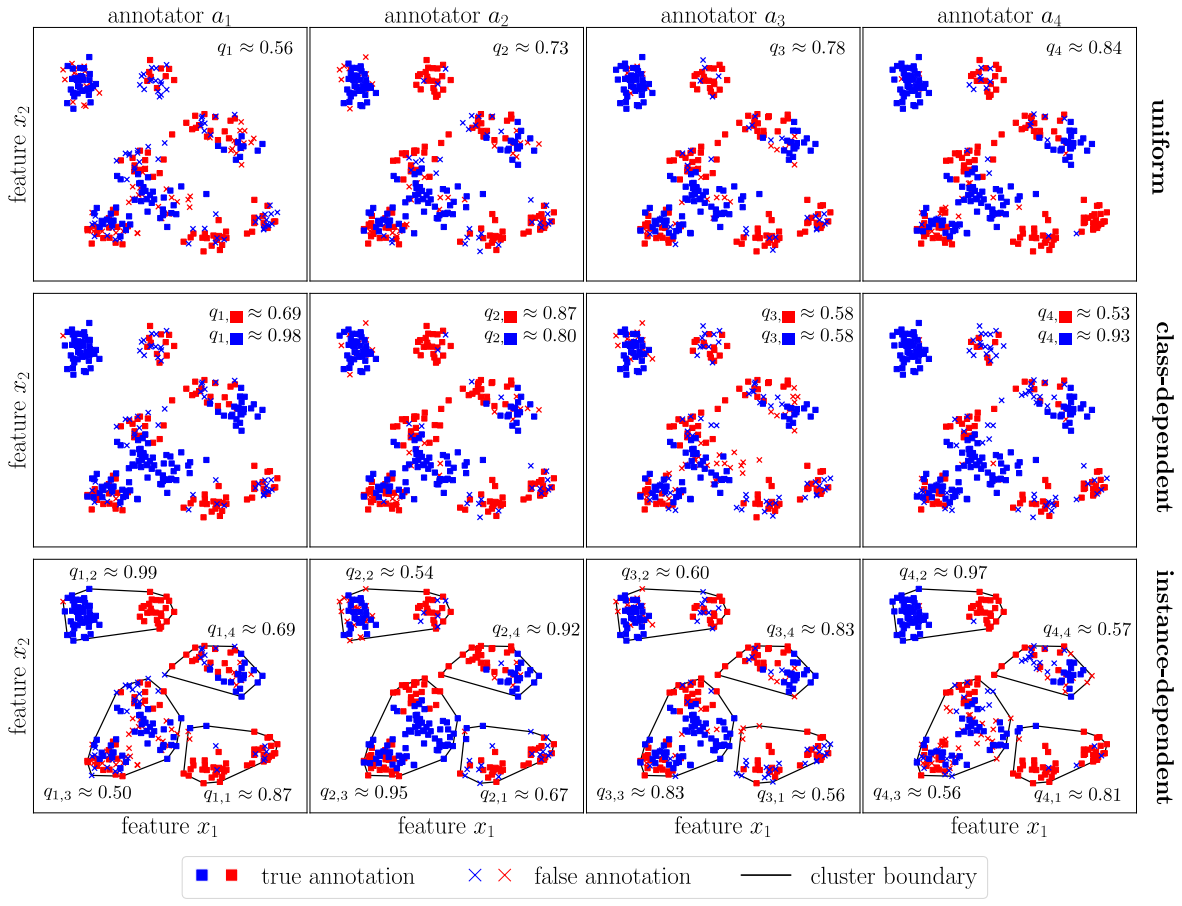


Fig. 4. Simulation of annotators with different types of performance values: The plots show the same two-dimensional data set annotated by different annotators. The first row of plots illustrates four annotators with uniform performance values, where each annotator a_m has a constant probability q_m of providing a correct annotation for an instance. The second row of plots illustrates four annotators with class-dependent performance values, where each annotator a_m has a constant probability $q_{m,y}$ of providing a correct annotation for an instance of class y. The third row of plots illustrates four annotators with instance-dependent performance values, where each annotator a_m has a constant probability $q_{m,c}$ of providing a correct annotation for an instance of cluster c.

C. Similarity Functions and Hyperparameters

Similarity Functions: We applied two different similarity functions depending on the type of data. For numerical data, we z-standardized all features and employed a *radial basis function* (RBF) kernel with bandwidth $s \in \mathbb{R}_{>0}$:

$$K_{\text{RBF}}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||_2^2}{2s^2}\right). \tag{4}$$

We set the bandwidth of the kernel according to the mean bandwidth criterion proposed in [3] with $\sigma_p = 1$:

$$s = \sqrt{\frac{2N\sum_{j=1}^{K}\sigma_p^2}{(N-1)\ln\frac{N-1}{\delta^2}}} \text{ with } \delta = \sqrt{2}\cdot 10^{-6}. \tag{5}$$

For the text data sets mozilla and compendium, which contain TF-IDF features, we applied the cosine similarity kernel:

$$K_{\text{COS}}(\mathbf{x}, \mathbf{x}') = \frac{\mathbf{x}^{\text{T}}\cdot\mathbf{x}'}{||\mathbf{x}||_2\cdot||\mathbf{x}'||_2}. \tag{6}$$

Hyperparameters: In the following, we provide an overview of the hyperparameters used by the different AL strategies. For this purpose, we use the notation of the respective articles.

- MaPAL: If not other stated, the parameters $M_{\max} = 2$ and $\beta_0 = 10^{-4}$ were employed.
- IEThresh [4]: This AL strategy has a threshold parameter $\epsilon \in [0, 1]$. Since no concrete value was proposed, we followed the evaluation in [13] and used $\epsilon = 0.8$.
- IEAdjCost [13]: This AL strategy has several hyperparameters, for which we used the following recommended values: $\epsilon \in [0, 1]$. Since no concrete value was proposed, we follow the evaluation in [13] and used $\epsilon = 0.8, R = 0.95, \delta = 0.4, \lambda = 0.4$.
- Proactive [9]: This AL strategy has no real hyperparameter but requires an initial set of instances assigned to their true class labels. Since such a set was not available, the 20% of the total budget B was spent to approximate these true class labels for randomly selected instances through the majority vote of all annotators. This approximation was proposed by the authors of Proactive.
- CEAL [7]: Similar to Proactive, this AL strategy requires an initial set of instances assigned to their true class labels. Therefor, we employed the same procedure as for Proactive. Additionally, CEAL needs a similarity function. In our evaluation, we employed the the one used by the SbC. The number of nearest neighbors was set to $t = 10$.
- ALIO [2]: Similar to Proactive, this AL strategy requires an initial set of instances assigned to their true class labels. Therefor, we employed the same procedure as for Proactive. Additionally, ALIO trains logistic regression models. Since o concrete parameters were proposed for these models, we relied on the default parameters of Scikit-learn [10].

D. Execution Times

We conducted all runs of the experiments on a computer cluster, which was equipped with multiple CPUs named `Intel(R) Xeon(R) CPU E5-2683 v4 @ 2.10GHz`. Additionally, we used the Slurm workload manager[3] to execute 100 runs of experiments in parallel. We evaluated seven AL strategies on 79 data sets:

- 25 data sets with simulated annotators having uniform performance values,
- 25 data sets with simulated annotators having class-dependent performance values,
- 25 data sets with simulated annotators having instance-dependent performance values,
- 4 data sets with real-world annotators.

Moreover, we repeated each test of an AL strategy of a data set 100 times. As a result, there were $7 \cdot 79 \cdot 100 = 55,300$ runs in total. The average execution times for the different experiments are reported in Table III. Since the data sets with simulated annotators only differ in the simulation of performance values, the execution times did not much vary between them. Hence, we report only the execution times for the 25 data sets with annotators having instance-dependent performance values.

E. Further Results

In this section, we provide further results of our experimental evaluation.

- Fig. 5 shows the robustness of MaPAL's performance regarding its hyperparameter β_0.
- Table IV gives an overview of the average AULC values in combination with their standard deviations for all AL strategies evaluated on data sets with real-world annotators and simulated annotators having instance-dependent performance values. Since these plot do not reveal much more information compared to the ranking statistics, the results for simulated annotators

[3] Documentation of Slurm workload manager: https://slurm.schedmd.com/documentation.html (last access: 06/11/2020)

having class-dependent performance values and simulated annotators having uniform performance values are not shown in favor of a more manageable presentation, i.e., smaller table.

- Figs. 5, 5, 5, and 5 show the learning curves for a subset of data sets. Next to learning curves showing the trend of the test and train misclassification error, there are also learning curves indicating the number of instances with at least one acquired annotation. This statistic provides information how the different AL strategies control the trade-off between re-annotating an instance and annotating an instance without any assigned annotations yet. Moreover, we report the number acquired annotations being false as learning curves. This way, we can investigate how the different annotator models help in improving the annotator selection. The learning curves for the remaining data sets are given in a separate file.

TABLE III

OVERVIEW OF EXECUTION TIMES FOR DATA SETS WITH SIMULATED ANNOTATORS: THE TABLE LISTS THE AVERAGE EXECUTION TIMES IN SECONDS THAT AN AL STRATEGY REQUIRED TO SELECT A QUERY SET $\mathcal{S}$ FOR A GIVEN DATA SET.

data set (no. of instances/classes)	MaPAL	CEAL	IEthresh	IEAdjCost	Proactive	ALIO	Random
data sets with simulated annotators							
iris (150/3)	0.0452	0.0010	0.0009	0.0010	0.0025	0.0010	0.0001
wine (168/3)	0.0638	0.0012	0.0009	0.0010	0.0031	0.0012	0.0001
parkinsons (195/2)	0.0435	0.0012	0.0009	0.0008	0.0028	0.0012	0.0001
prnn-craps (200/2)	0.0309	0.0011	0.0009	0.0008	0.0030	0.0011	0.0001
sonar (208/2)	0.0743	0.0012	0.0009	0.0008	0.0028	0.0012	0.0001
seeds (210/3)	0.0622	0.0012	0.0008	0.0011	0.0031	0.0013	0.0001
glass (214/6)	0.1870	0.0013	0.0014	0.0014	0.0028	0.0014	0.0001
vertebra-column (310/3)	0.1303	0.0015	0.0009	0.0011	0.0035	0.0015	0.0001
ecoli (336/8)	0.2484	0.0015	0.0014	0.0014	0.0033	0.0015	0.0001
ionosphere (351/2)	0.0802	0.0015	0.0010	0.0009	0.0032	0.0016	0.0001
user-knowledge (403/5)	0.2940	0.0016	0.0011	0.0012	0.0038	0.0016	0.0001
chscase-vine (468/2)	0.0964	0.0022	0.0010	0.0009	0.0042	0.0024	0.0002
kc2 (522/2)	0.0715	0.0023	0.0010	0.0011	0.0042	0.0026	0.0002
wdbc (569/2)	0.1223	0.0028	0.0010	0.0010	0.0052	0.0029	0.0002
balance-scale (625/3)	0.2255	0.0031	0.0011	0.0014	0.0054	0.0028	0.0002
blood-transfusion (748/2)	0.1221	0.0043	0.0012	0.0010	0.0061	0.0040	0.0002
pima-indians (768/2)	0.1714	0.0044	0.0012	0.0009	0.0066	0.0041	0.0002
vehicle (846/4)	0.3831	0.0039	0.0017	0.0017	0.0063	0.0038	0.0002
qsar-biodeg (1055/2)	0.3256	0.0063	0.0016	0.0022	0.0079	0.0055	0.0002
banknote (1372/2)	0.1408	0.0050	0.0019	0.0025	0.0080	0.0050	0.0002
steel-plates-fault (1941/2)	0.6056	0.0091	0.0016	0.0019	0.0115	0.0094	0.0003
segment (2310/7)	1.9963	0.0130	0.0038	0.0058	0.0156	0.0139	0.0003
phoneme (5404/2)	1.6302	0.0514	0.0043	0.0141	0.0488	0.0512	0.0005
satimage (6430/6)	6.1682	0.0595	0.0116	0.0147	0.0633	0.0573	0.0006
wind (6574/2)	3.0672	0.0891	0.0080	0.0263	0.0955	0.0906	0.0008
data sets with real-world annotators							
medical (287/2)	0.0566	0.0017	0.0009	0.0006	0.0037	0.0018	0.0002
grid (300/5)	0.1319	0.0017	0.0013	0.0016	0.0038	0.0017	0.0002
mozilla (675/4)	0.3014	0.0025	0.0015	0.0012	0.0046	0.0025	0.0002
compendium (962/4)	0.4476	0.0025	0.0013	0.0011	0.0050	0.0026	0.0002

TABLE IV

AREA UNDER THE LEARNING CURVE: THE TABLE LISTS THE MEANS AND STANDARD DEVIATIONS OF THE AREA UNDER THE LEARNING CURVE FOR ALL COMBINATIONS OF AL STRATEGY AND DATA SET ACROSS 100 RUNS. LOW VALUES ARE CONSIDERED BEST. THE BEST AL STRATEGY PER DATA SET IS PRINTED IN BOLD. THE WILCOXON-SIGNED-RANK TEST SHOWS PAIRWISE SIGNIFICANCE (P-VALUE 0.001) BETWEEN MAPAL($\beta_0 = 10^{-4}, M_{\text{MAX}} = 2$) AND ITS COMPETITOR ($*$ MAPAL BETTER, † COMPETITOR BETTER).

data set (no. of instances/classes)	MaPAL	CEAL	IEthresh	IEAdjCost	Proactive	ALIO	Random
instance-dependent							
iris (150/3)	**0.15** $\pm$ 0.049	0.24 $\pm$ 0.069*	0.26 $\pm$ 0.10*	0.26 $\pm$ 0.092*	0.24 $\pm$ 0.066*	0.23 $\pm$ 0.077*	0.21 $\pm$ 0.060*
wine (168/3)	**0.15** $\pm$ 0.053	0.26 $\pm$ 0.076	0.18 $\pm$ 0.062	0.19 $\pm$ 0.069*	0.23 $\pm$ 0.067*	0.24 $\pm$ 0.069*	0.25 $\pm$ 0.061*
parkinsons (195/2)	**0.24** $\pm$ 0.060	0.29 $\pm$ 0.065*	0.25 $\pm$ 0.064	0.27 $\pm$ 0.061	0.29 $\pm$ 0.062*	0.29 $\pm$ 0.074*	0.27 $\pm$ 0.060*
prnn-craps (200/2)	**0.25** $\pm$ 0.041	0.33 $\pm$ 0.054*	0.32 $\pm$ 0.054*	0.32 $\pm$ 0.056*	0.33 $\pm$ 0.052*	0.32 $\pm$ 0.051*	0.29 $\pm$ 0.045*
sonar (208/2)	**0.30** $\pm$ 0.052	0.33 $\pm$ 0.039*	0.34 $\pm$ 0.038*	0.33 $\pm$ 0.045*	0.33 $\pm$ 0.037*	0.33 $\pm$ 0.044*	0.32 $\pm$ 0.046*
seeds (210/3)	**0.13** $\pm$ 0.037	0.20 $\pm$ 0.045*	0.22 $\pm$ 0.078*	0.22 $\pm$ 0.076*	0.19 $\pm$ 0.043*	0.20 $\pm$ 0.053*	0.19 $\pm$ 0.039*
glass (214/6)	**0.47** $\pm$ 0.047	0.55 $\pm$ 0.057*	0.52 $\pm$ 0.052*	0.52 $\pm$ 0.052*	0.55 $\pm$ 0.057*	0.53 $\pm$ 0.055*	0.52 $\pm$ 0.052*
vertebra-column (310/3)	0.32 $\pm$ 0.040	0.34 $\pm$ 0.049	**0.32** $\pm$ 0.055	0.32 $\pm$ 0.053	0.35 $\pm$ 0.052*	0.35 $\pm$ 0.051*	0.35 $\pm$ 0.037*

Continued on next page.

TABLE IV – Continued from previous page.

data set (no. of instances/classes)	MaPAL	CEAL	IEthresh	IEAdjCost	Proactive	ALIO	Random
ecoli (336/8)	**0.20** ± 0.030	0.27 ± 0.043*	0.25 ± 0.043*	0.26 ± 0.048*	0.26 ± 0.043*	0.25 ± 0.037*	0.27 ± 0.043*
ionosphere (351/2)	**0.19** ± 0.046	0.27 ± 0.052*	0.23 ± 0.046*	0.24 ± 0.055*	0.26 ± 0.053*	0.24 ± 0.051*	0.24 ± 0.048*
user-knowledge (403/5)	**0.40** ± 0.045	0.43 ± 0.041*	0.43 ± 0.046*	0.45 ± 0.050*	0.44 ± 0.040*	0.43 ± 0.047*	0.45 ± 0.041*
chscase-vine (468/2)	**0.27** ± 0.032	0.33 ± 0.044*	0.34 ± 0.044*	0.33 ± 0.043*	0.33 ± 0.043*	0.32 ± 0.044*	0.31 ± 0.039*
kc2 (522/2)	0.19 ± 0.034	0.19 ± 0.051	0.19 ± 0.035	0.20 ± 0.048	0.20 ± 0.042	0.19 ± 0.051	**0.19** ± 0.023
wdbc (569/2)	**0.09** ± 0.024	0.12 ± 0.026*	0.09 ± 0.025	0.09 ± 0.025	0.13 ± 0.029*	0.12 ± 0.026*	0.13 ± 0.027*
balance-scale (625/3)	**0.23** ± 0.025	0.24 ± 0.031*	0.24 ± 0.026	0.24 ± 0.031*	0.26 ± 0.033*	0.24 ± 0.032*	0.26 ± 0.029*
blood-transfusion (748/2)	**0.26** ± 0.031	0.28 ± 0.038*	0.26 ± 0.030	0.26 ± 0.032	0.29 ± 0.044*	0.29 ± 0.037*	0.28 ± 0.032*
pima-indians (768/2)	**0.31** ± 0.020	0.31 ± 0.021*	0.32 ± 0.021*	0.32 ± 0.020*	0.32 ± 0.021*	0.31 ± 0.021	0.32 ± 0.022*
vehicle (846/4)	**0.36** ± 0.022	0.42 ± 0.026*	0.43 ± 0.028*	0.44 ± 0.035*	0.43 ± 0.026*	0.41 ± 0.025*	0.41 ± 0.025*
qsar-biodeg (1055/2)	**0.22** ± 0.024	0.25 ± 0.026*	0.23 ± 0.026	0.23 ± 0.031	0.25 ± 0.028*	0.25 ± 0.028*	0.25 ± 0.024*
banknote (1372/2)	**0.02** ± 0.007	0.07 ± 0.017*	0.03 ± 0.013*	0.04 ± 0.014*	0.06 ± 0.016*	0.06 ± 0.015*	0.06 ± 0.016*
steel-plates-fault (1941/2)	**0.13** ± 0.040	0.23 ± 0.032*	0.16 ± 0.044*	0.16 ± 0.041*	0.21 ± 0.029*	0.21 ± 0.029*	0.19 ± 0.025*
segment (2310/7)	**0.14** ± 0.019	0.24 ± 0.030*	0.34 ± 0.070*	0.37 ± 0.079*	0.23 ± 0.022*	0.26 ± 0.028*	0.22 ± 0.019*
phoneme (5404/2)	**0.23** ± 0.025	0.27 ± 0.030*	0.25 ± 0.019*	0.25 ± 0.025*	0.27 ± 0.025*	0.27 ± 0.033*	0.26 ± 0.018*
satimage (6430/6)	**0.17** ± 0.019	0.28 ± 0.034*	0.29 ± 0.046*	0.30 ± 0.065*	0.27 ± 0.028*	0.24 ± 0.025*	0.23 ± 0.016*
wind (6574/2)	**0.22** ± 0.016	0.24 ± 0.027*	0.24 ± 0.030*	0.24 ± 0.030*	0.25 ± 0.026*	0.25 ± 0.023*	0.25 ± 0.018*
real-world							
medical (287/2)	**0.25** ± 0.029	0.27 ± 0.033*	0.27 ± 0.036*	0.26 ± 0.038	0.26 ± 0.032*	0.27 ± 0.034*	0.26 ± 0.027
grid (300/5)	**0.49** ± 0.037	0.52 ± 0.035*	0.49 ± 0.037	0.49 ± 0.035	0.50 ± 0.039*	0.51 ± 0.039*	0.52 ± 0.037*
mozilla (675/4)	0.41 ± 0.038	0.46 ± 0.048*	**0.41** ± 0.025	0.43 ± 0.031*	0.50 ± 0.041*	0.47 ± 0.046*	0.46 ± 0.038*
compendium (962/4)	**0.46** ± 0.018	0.51 ± 0.026*	0.51 ± 0.023*	0.50 ± 0.023*	0.51 ± 0.027*	0.51 ± 0.026*	0.49 ± 0.024*

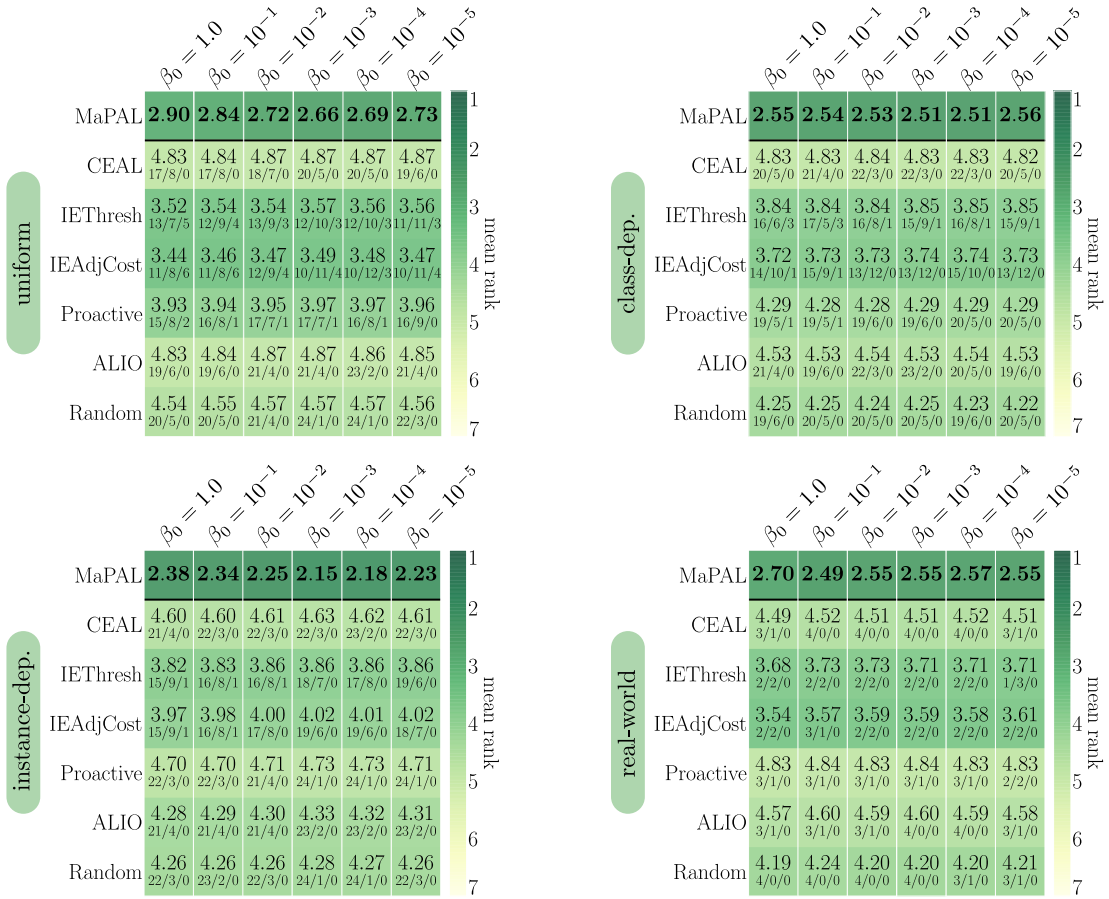
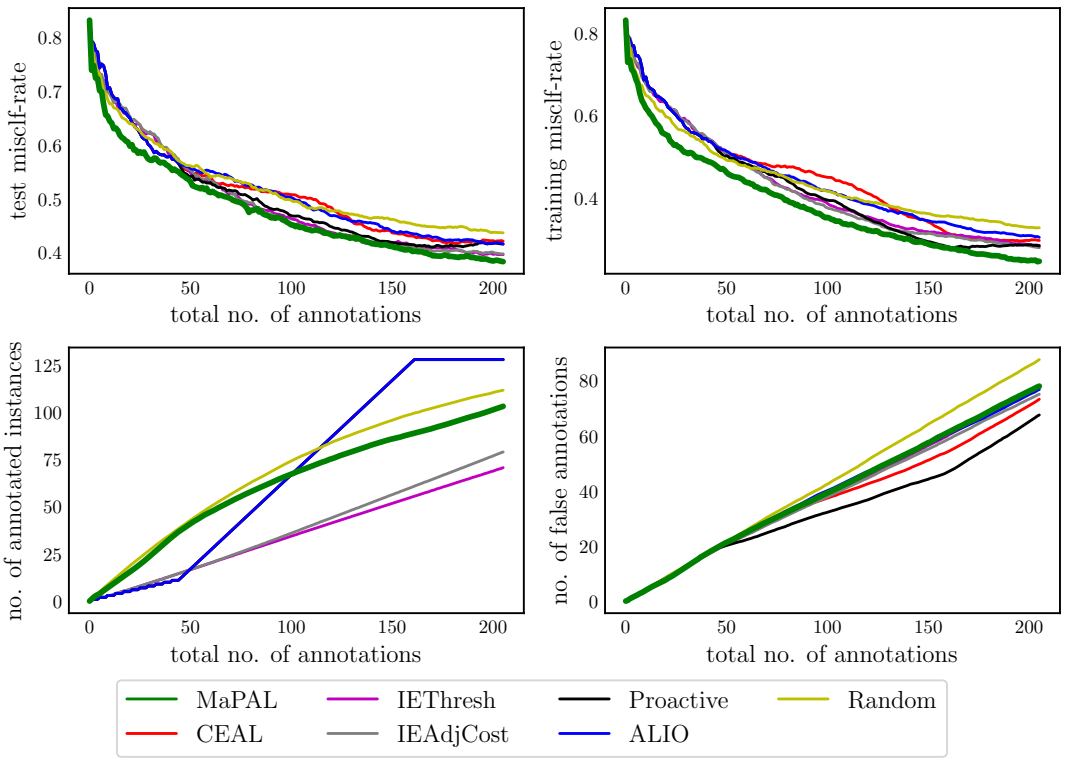
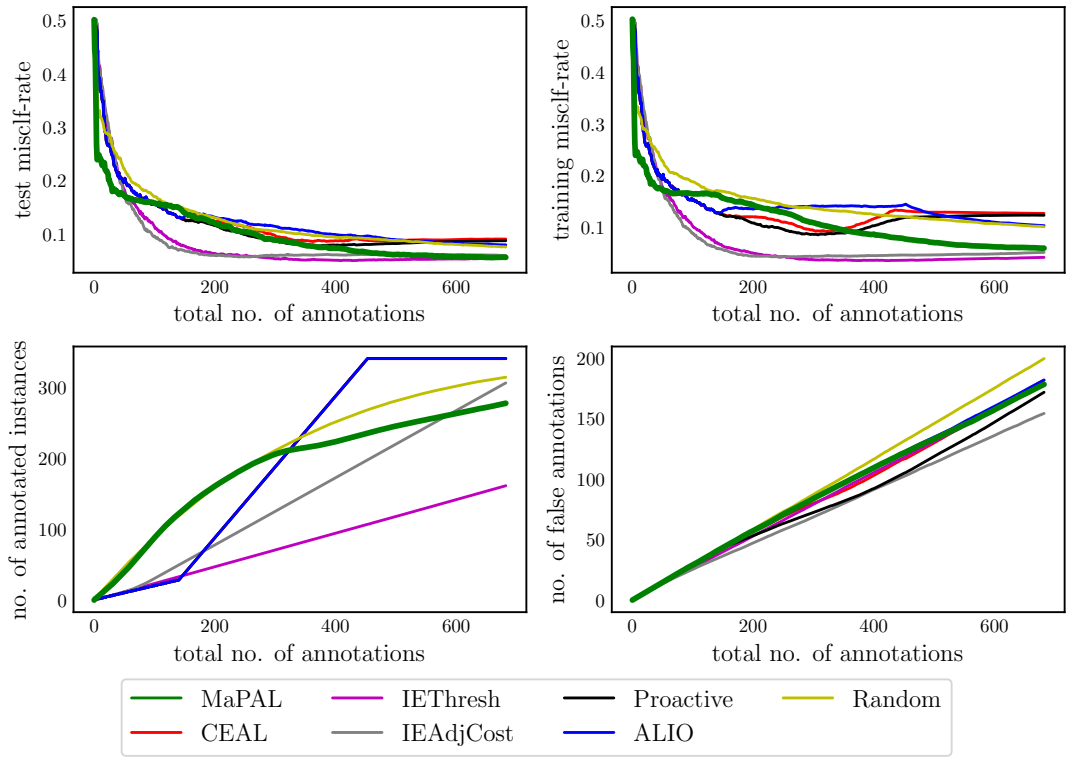


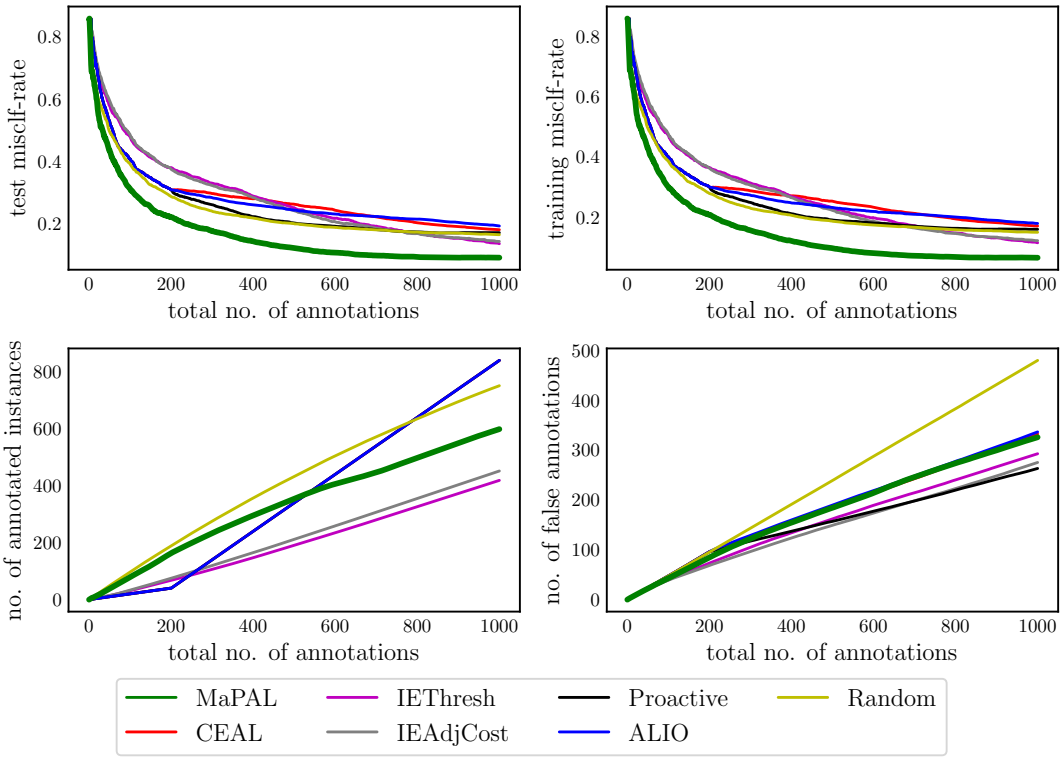
Fig. 5. Ranking statistics for different values of β_0: The above plots give an overview of the mean rank of each AL strategy taken over each of the four collections of data sets, i.e., uniform, class-dependent, instance-dependent, and real-world. Additionally, for each of these data set collections, it indicates the total number of wins (i.e., number of $*$) / ties / losses (i.e., number of †) of MaPAL for a given β_0 and $M_{\max} = 2$ in comparison to its respective competitor.



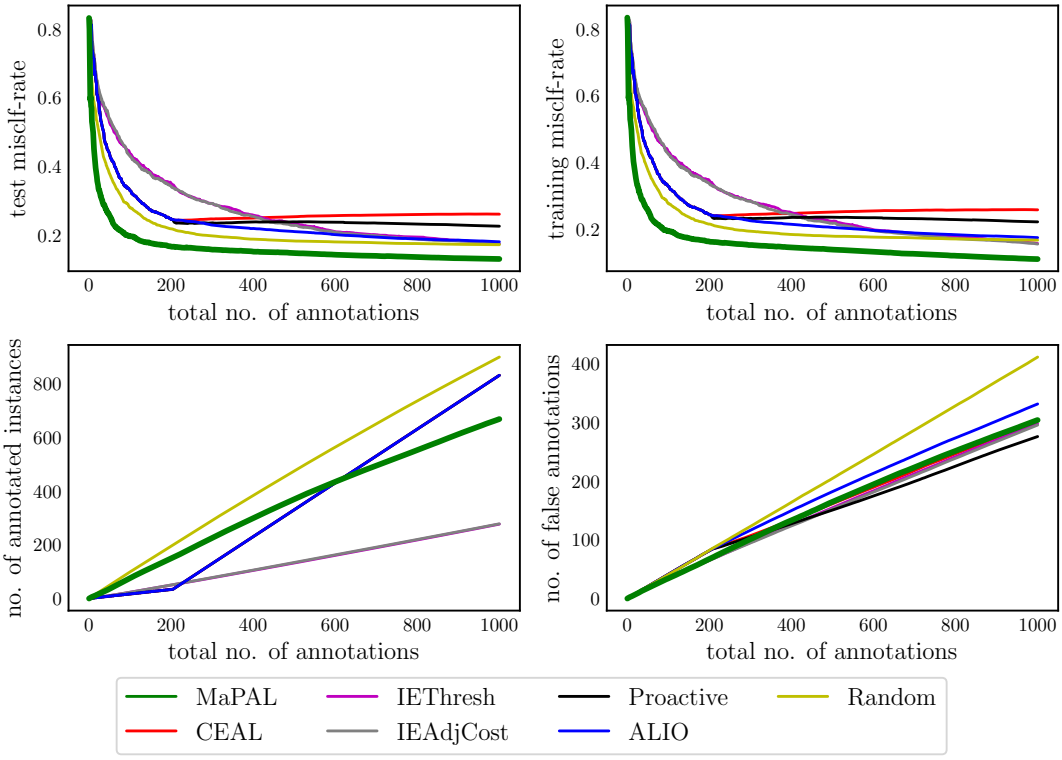
(a) glass – uniform annotation performances



(b) wdbc – uniform annotation performances

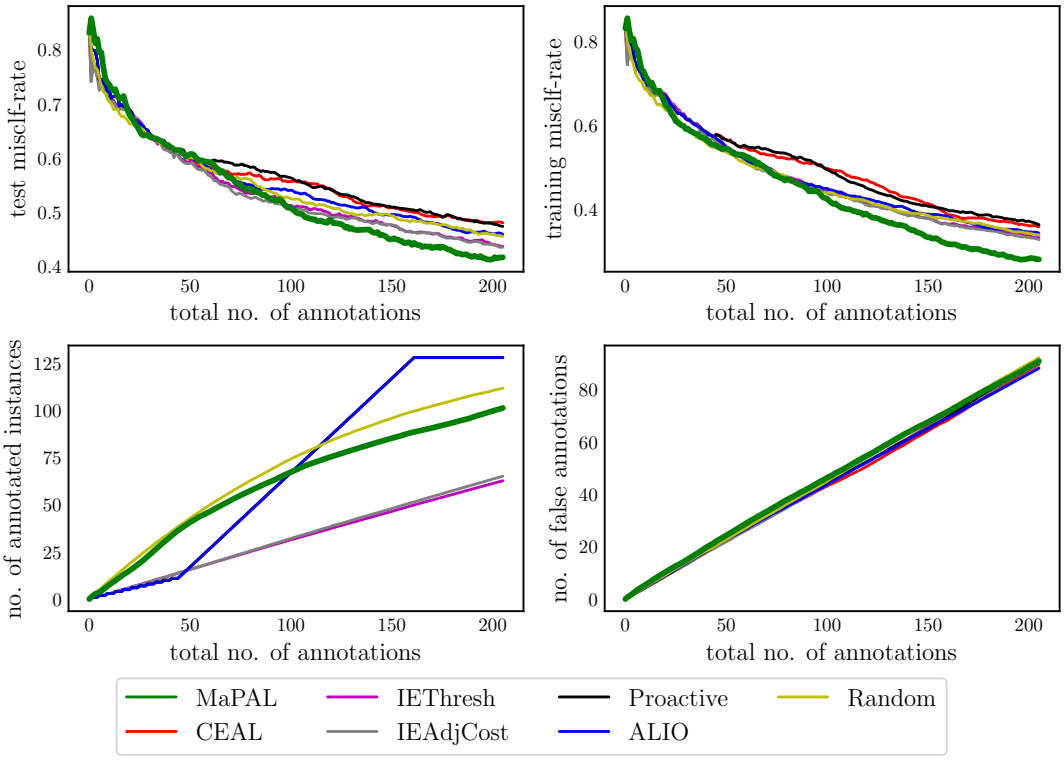


(c) segment – uniform annotation performances

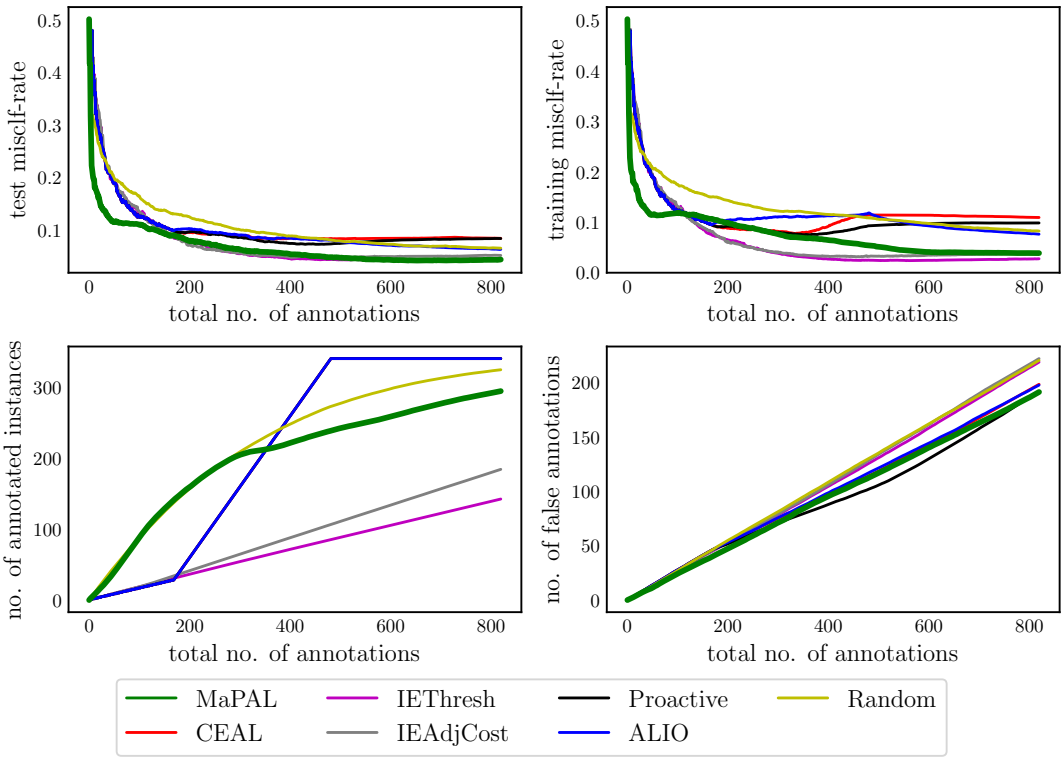


(d) satimage – uniform annotation performances

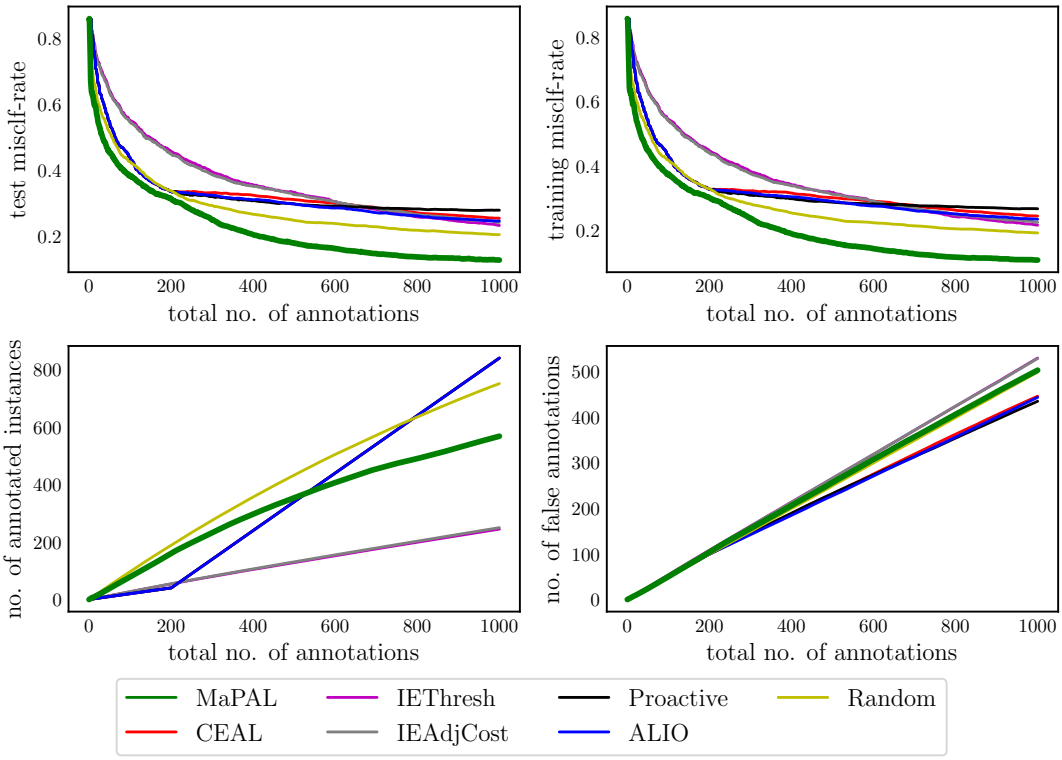
Fig. 5. Learning curves for data sets with simulated annotators having **uniform** performance values.



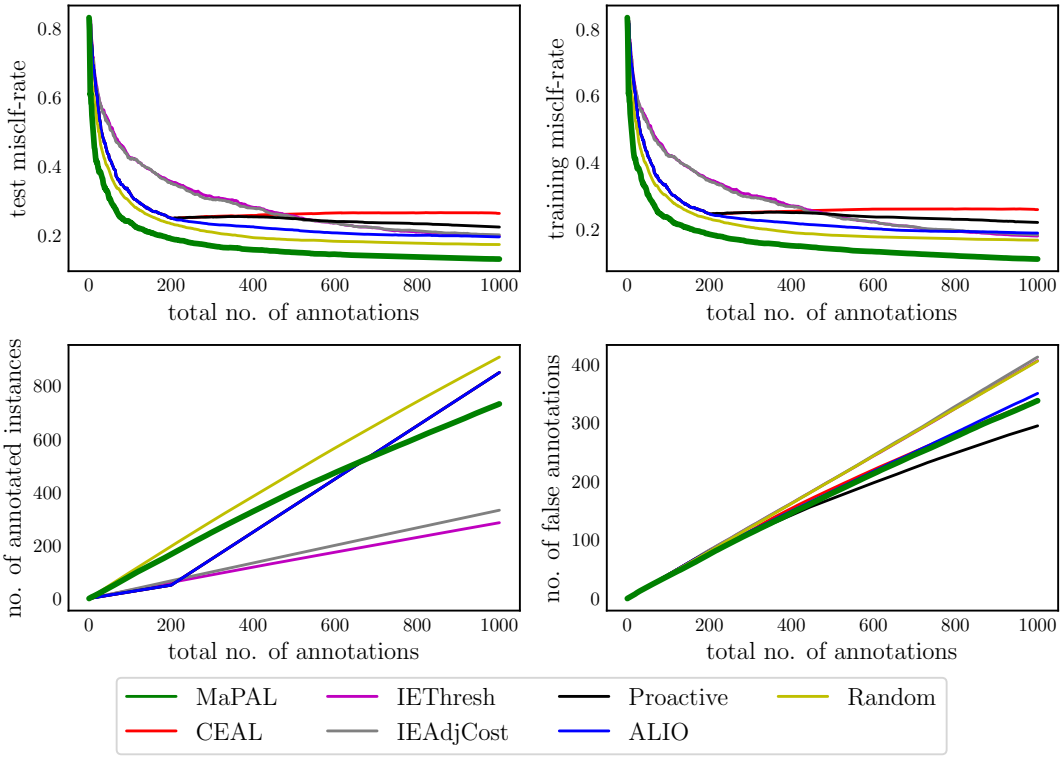
(a) glass – class-dependent annotation performances



(b) wdbc – class-dependent annotation performances

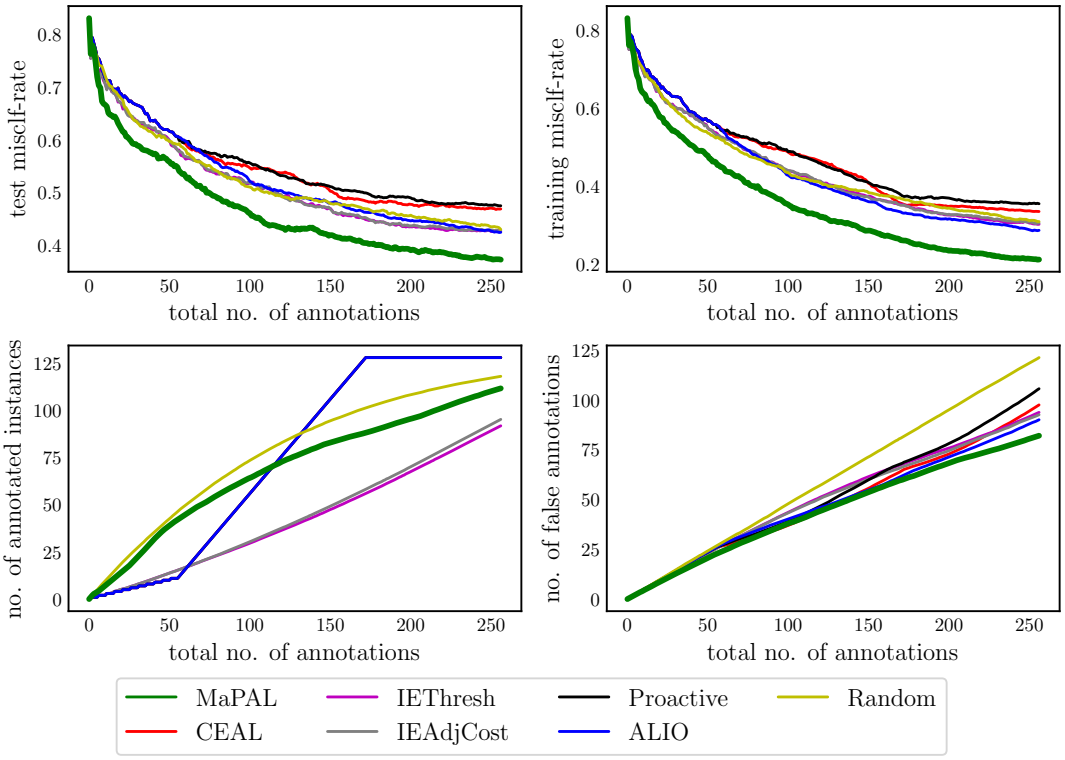


(c) segment – class-dependent annotation performances

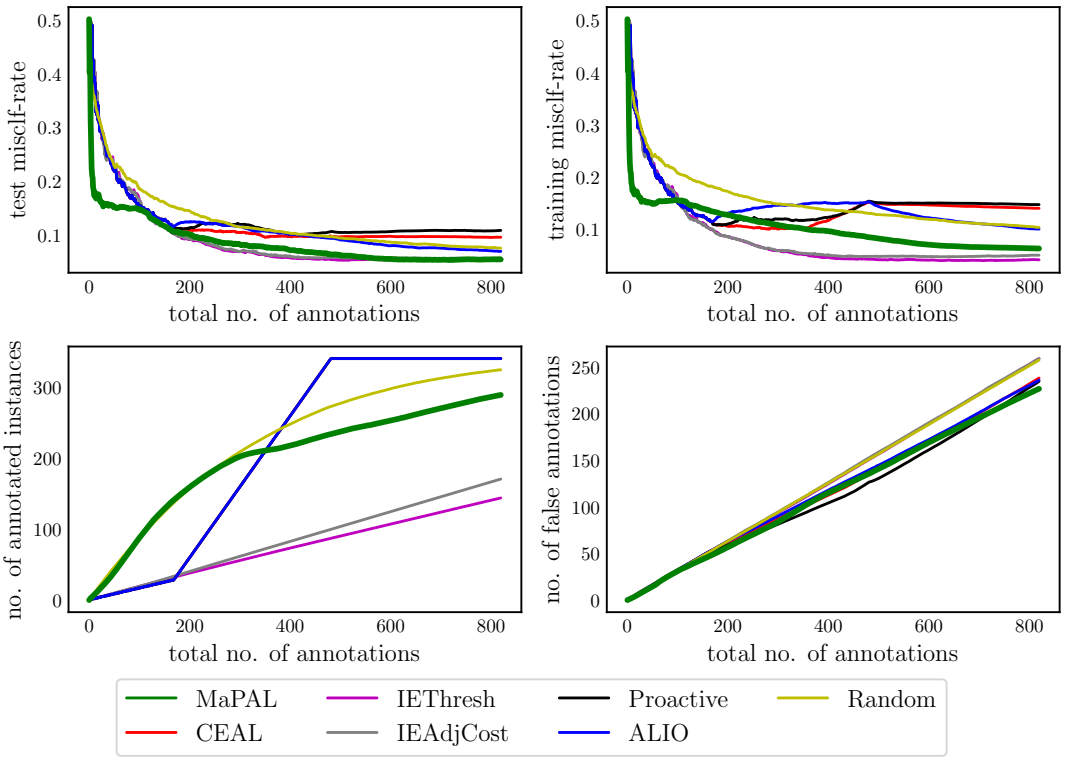


(d) satimage – class-dependent annotation performances

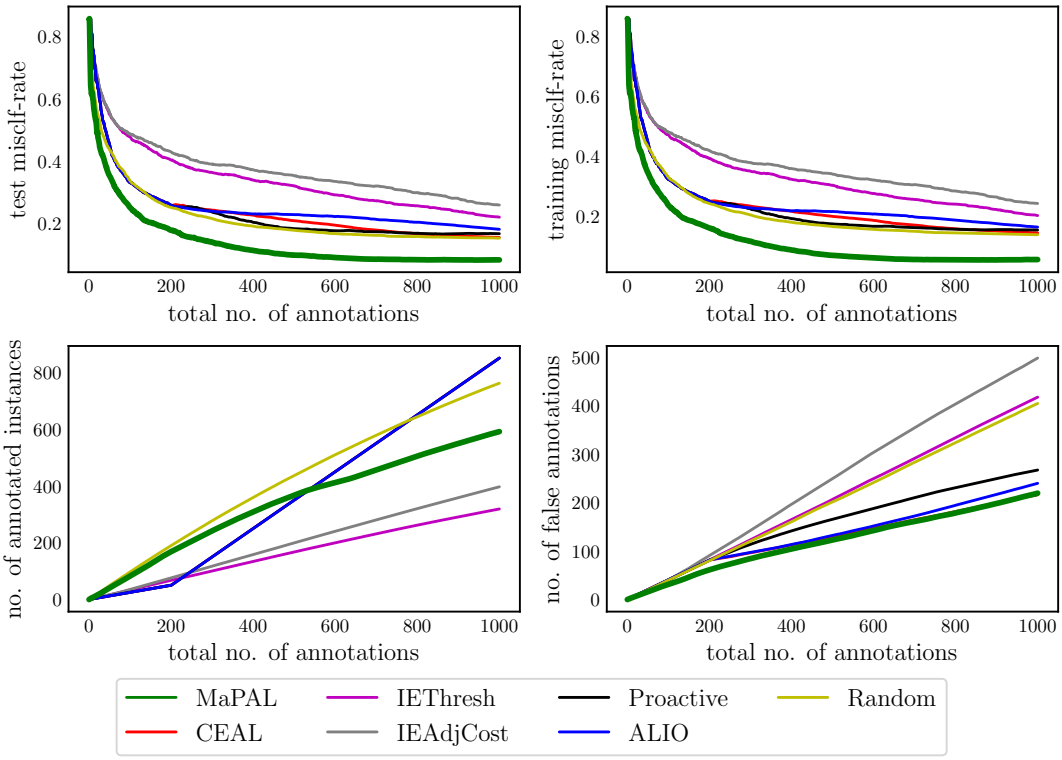
Fig. 5. Learning curves for data sets with simulated annotators having **class-dependent** performance values.



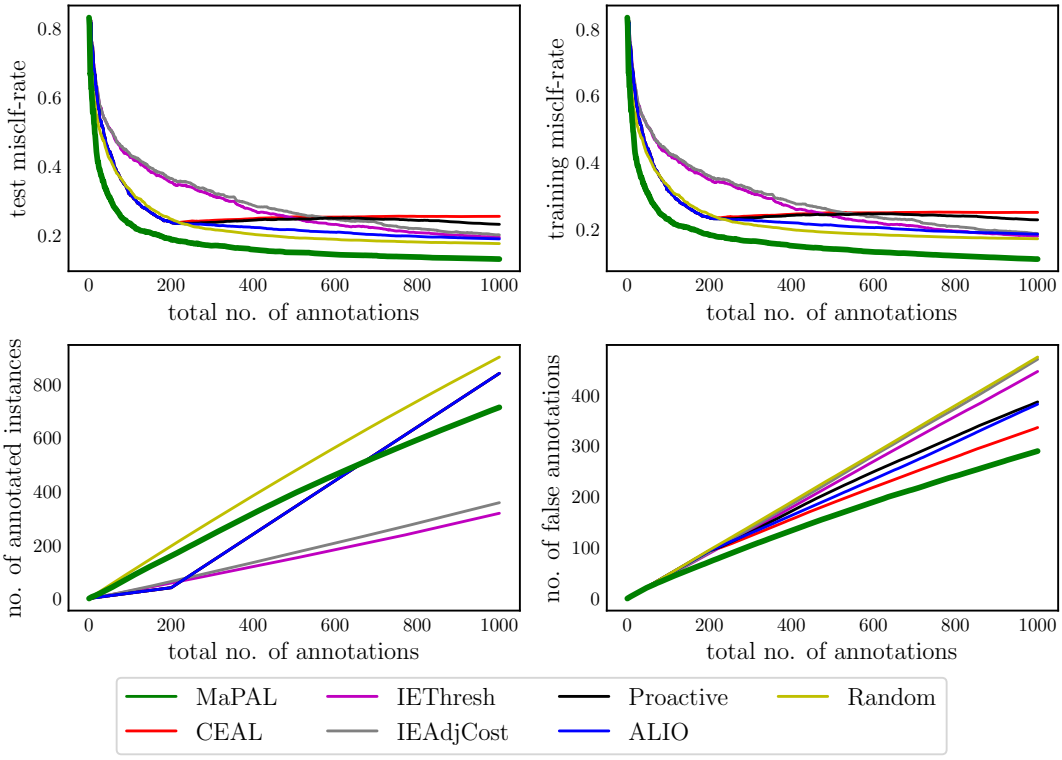
(a) glass – instance-dependent annotation performances



(b) wdbc – instance-dependent annotation performances

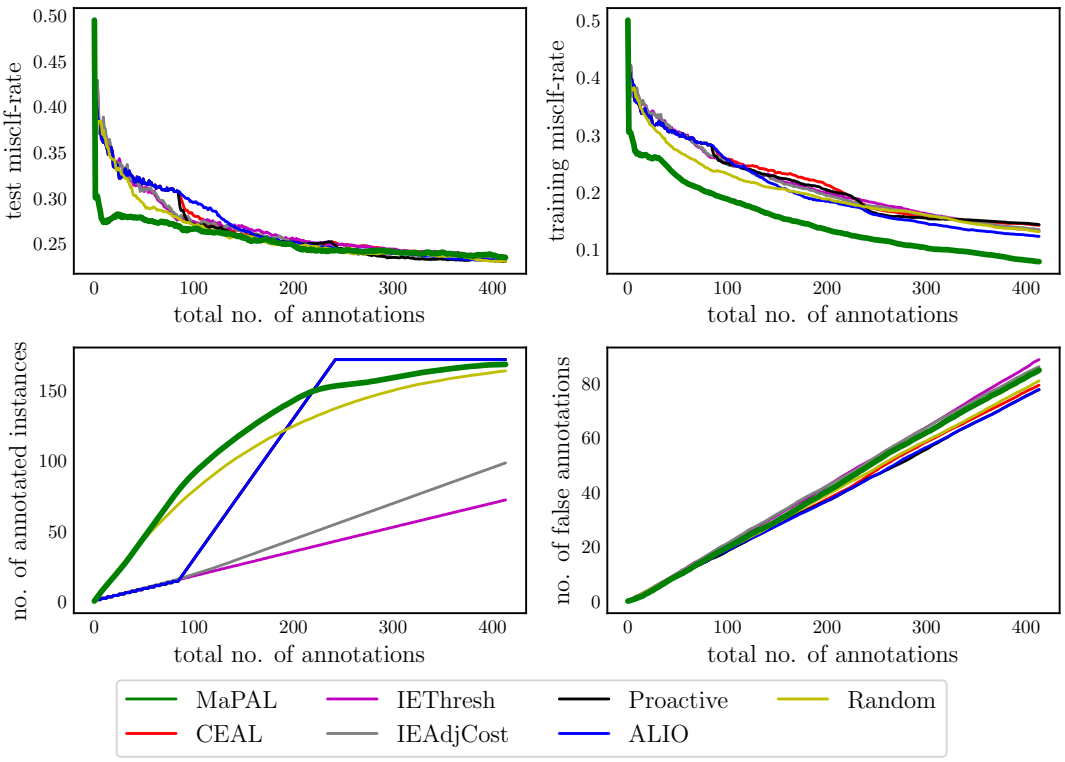


(c) segment – instance-dependent annotation performances

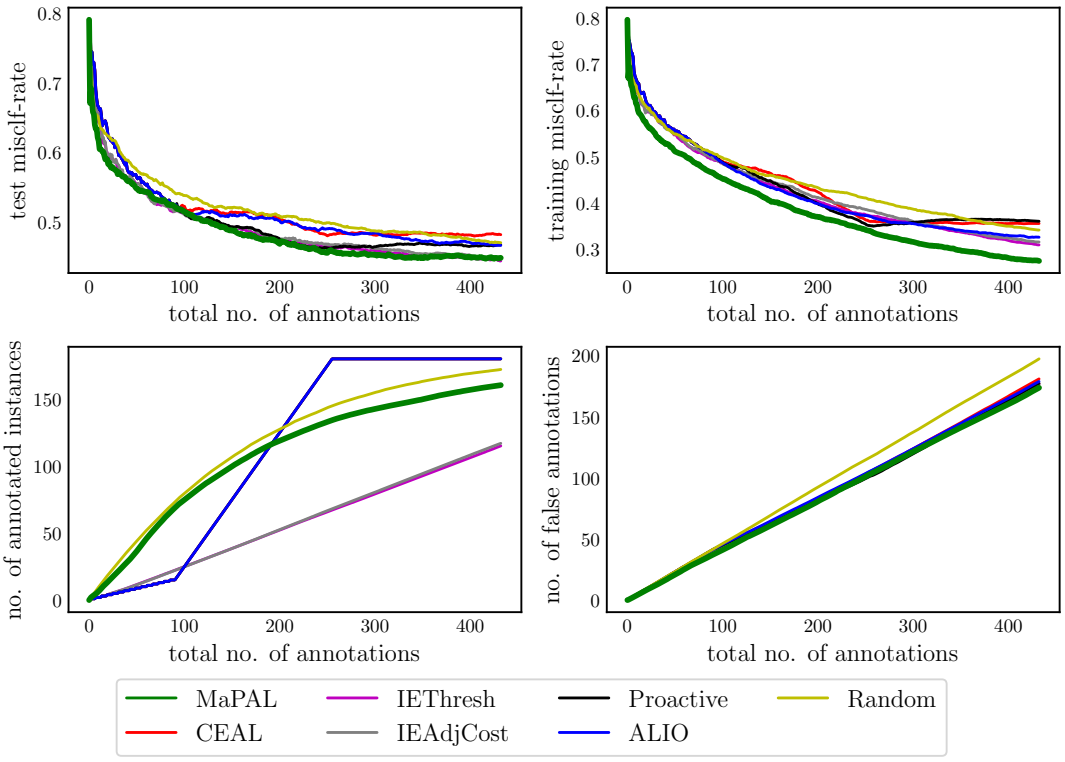


(d) satimage – instance-dependent annotation performances

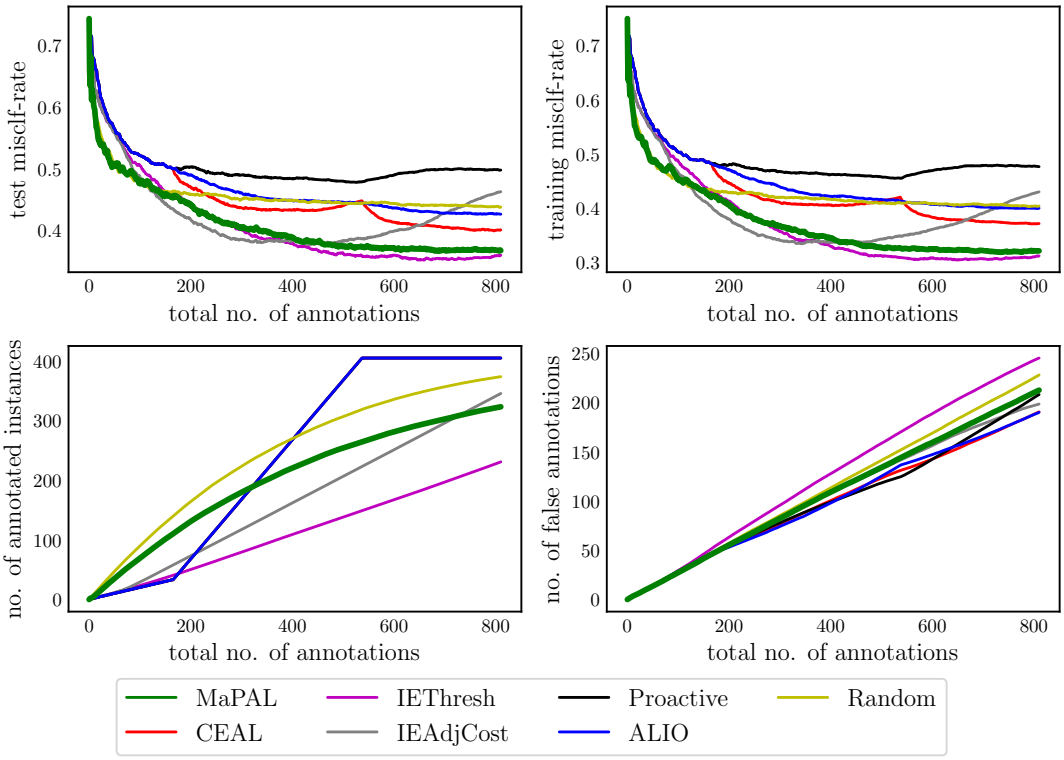
Fig. 5. Learning curves for data sets with simulated annotators having **instance-dependent** performance values.



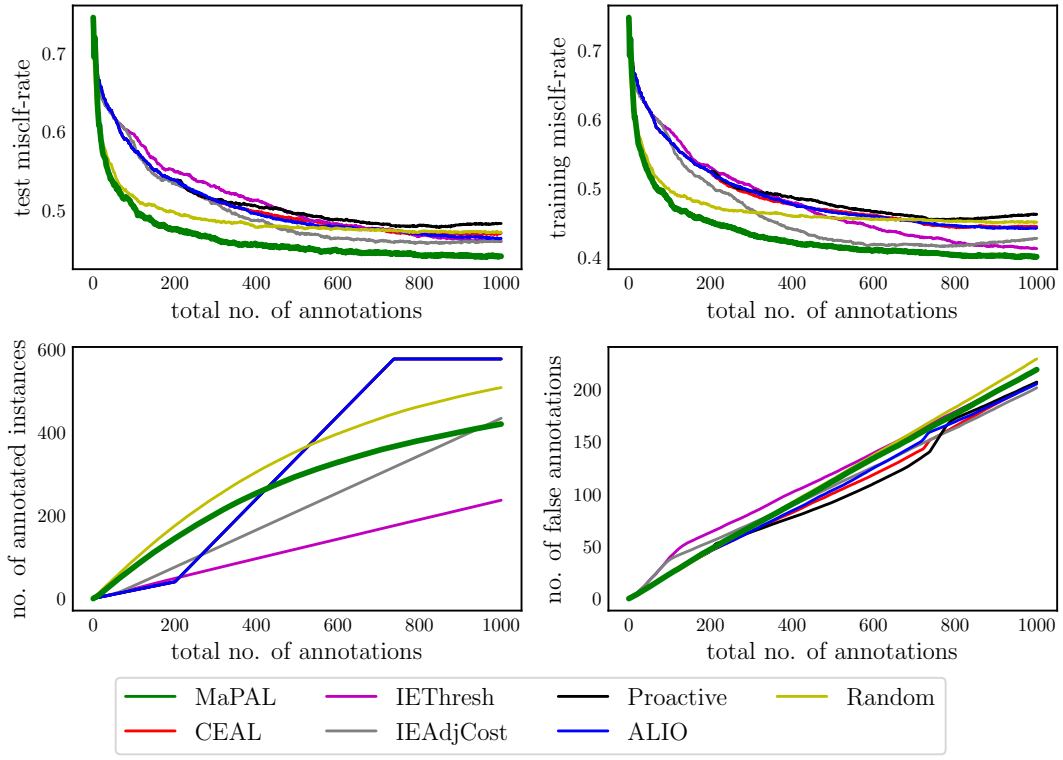
(a) medical – real-world annotators



(b) grid – real-world annotators



(c) mozilla – real-world annotators



(d) compendium – real-world annotators

Fig. 5. Learning curves for **real-world** data sets.

REFERENCES

[1] S. Breker, A. Claudi, and B. Sick. Capacity of Low-Voltage Grids for Distributed Generation: Classification by Means of Stochastic Simulations. *IEEE Transactions on Power Systems*, 30(2):689–700, 2015.

[2] S. Chakraborty. Asking the Right Questions to the Right Users: Active Learning with Imperfect Oracles. In *AAAI Conference on Artificial Intelligence*, 2020.

[3] A. Chaudhuri, D. Kakde, C. Sadek, L. Gonzalez, and S. Kong. The Mean and Median Criteria for Kernel Bandwidth Selection for Support Vector Data Description. In *IEEE International Conference on Data Mining Workshop*, pages 842–849, 2017.

[4] P. Donmez, J. Carbonell, and J. Schneider. A Probabilistic Framework to Learn from Multiple Annotators with Time-Varying Accuracy. In *SIAM International Conference on Data Mining*, pages 826–837, 2010.

[5] K. Fernandes, J. S. Cardoso, and J. Fernandes. Transfer Learning with Partial Observability Applied to Cervical Cancer Screening. In *Pattern Recognition and Image Analysis*, pages 243–250, Faro, Portugal, 2017.

[6] J. Hernández-González, D. Rodriguez, I. Inza, R. Harrison, and J. A. Lozano. Two datasets of defect reports labeled by a crowd of annotators of unknown reliability. *Data in Brief*, 18:840–845, 2018.

[7] S. J. Huang, J. L. Chen, X. Mu, and Z. H. Zhou. Cost-effective Active Learning from Diverse Labelers. In *International Joint Conference on Artificial Intelligence*, pages 1879–1885, 2017.

[8] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[9] S. Moon and J. G. Carbonell. Proactive learning with multiple class-sensitive labelers. In *International Conference on Data Science and Advanced Analytics*, pages 32–38, 2014.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[11] J. Vanschoren, J. N. van Rijn, B. Bischl, and L. Torgo. OpenML: Networked Science in Machine Learning. *SIGKDD Explorations*, 15(2):49–60, 2013.

[12] Y. Yan, R. Rosales, G. Fung, and J. G. Dy. Active Learning from Crowds. In *International Conference on Machine Learning*, pages 1161–1168, Bellevue, WA, 2011.

[13] Y. Zheng, S. Scott, and K. Deng. Active Learning from Multiple Noisy Labelers with Varied Costs. In *IEEE International Conference on Data Mining*, pages 639–648, 2010.