JMLR: Workshop and Conference Proceedings 10 (2010) 1–26 Workshop on Active Learning and Experimental Design

Results of the Active Learning Challenge

Isabelle Guyon ISABELLE@CLOPINET.COM
Clopinet, California

Gavin Cawley GCC@CMP.UEA.AC.UK
University of East Anglia, UK

Gideon Dror GIDEON@MTA.AC.IL
Academic College of Tel-Aviv-Yaffo, Israel

Vincent Lemaire VINCENT.LEMAIRE@ORANGE-FTGROUP.COM
Orange Labs, France

Editor: Neil Lawrence

Abstract

We organized a machine learning challenge on "active learning", addressing problems where labeling data is expensive, but large amounts of unlabeled data are available at low cost. Examples include handwriting and speech recognition, document classification, vision tasks, drug design using recombinant molecules and protein engineering. The algorithms may place a limited number of queries to get new sample labels. The design of the challenge and its results are summarized in this paper and the best contributions made by the participants are included in these proceedings. The website of the challenge remains open as a resource for students and researchers (`http://clopinet.com/al`).

1. Background

The accumulation of massive amounts of unlabeled data and the cost of labeling have triggered a resurgence of interest in active learning. However, the newly proposed methods have never been evaluated in a fair and open contest. The challenge we organized has stimulated research in the field and provides a comparative study free of "inventor bias".

Modeling can have a number of objectives, including understanding or explaining the data, developing scientific theories, and making predictions. We focus in this challenge on predictive modeling, in a setup known in machine learning as "supervised learning". The goal is to predict an outcome y given a number of predictor variables $\mathbf{x} = [x_1, x_2, ...x_n]$, also called features, attributes, or factors. During training, the model (also called the learning machine) is provided with example pairs $\{\mathbf{x}, y\}$ (the training examples) with which to adjust its parameters. After training, the model is evaluated with new example pairs (the test examples) to estimate its generalization performance. In our framework, example pairs can only be obtained at a cost; optimal data acquisition must compromise between selecting many informative example pairs and incurring a large expense for data collection. Typically, either a fixed budget is available and the generalization performance must be maximized or the data collection expenses must be minimized to reach or exceed a given generalization performance. Data pairs $\{\mathbf{x}, y\}$ are drawn identically and independently from an unknown distribution $P(\mathbf{x}, y)$. In the regular machine learning setting (passive learning),

a batch of training pairs is made available from the outset. In the active learning setting, the labels y are withheld and can be purchased from an oracle. The learning machine must select the examples, which look most promising in improving the predictive performance of theodel. There exist several variants of active learning:

- **Pool-based active learning:** A large pool of examples of $\mathbf{x}$ is made available from the outset of training.
- **Stream-based active learning:** Examples are made available continuously.
- **De novo query synthesis:** The learner can select arbitrary values of x, i.e. use examples not drawn from $P(\mathbf{x})$.

Other scenarios, not considered here, include cases in which data are not *i.i.d.*. Such situations occur in time series prediction, speech processing, unsegmented image analysis and document analysis.

Of the variants of active learning considered, pool-based active learning is of considerable importance in current applications of machine learning and data mining, because of the availability of large amounts of unlabeled data in many domains, including pattern recognition (handwriting, speech, airborne or satellite images, etc.), text processing (internet documents, archives), chemo-informatics (untested molecules from combinatorial chemistry), and marketing (large customer databases). These are typical examples of the scenarios we intended to study via the organization of this challenge. Stream-based active learning is also important when sensor data is continuously available and data cannot be easily stored. However, it is more difficult to evaluate in the context of a challenge, so we focus instead purely on pool-based active learning. Several of the techniques thus developed may also be applicable to stream-based active learning. The last type of active learning, "de-novo" query synthesis, will be addressed in upcoming experimental design challenges in which we will allow participants to intervene on $\mathbf{x}$. In this challenge, however we limit the actions of the participants to sampling from a set of fixed points in input space and query for y, we do not allow interventions on $\mathbf{x}$, such as setting particular values of x_i.

A number of query strategies with various criteria of optimality have been devised. Perhaps the simplest and most commonly used query strategy is uncertainty sampling (Lewis and Gale, 1994). In this framework, an active learner queries the instances that it can label with least confidence. This of course requires the use of a model that is capable of assessing prediction uncertainty, such as a logistic model for binary classification problems. Another general active learning framework queries the labels of the instances that would impart the greatest change in the current model (expected model change), if we knew the labels. Since discriminative probabilistic models are usually trained with gradient-based optimization, the "change" imparted can be measured by the magnitude of the gradient (Settles et al., 2008a). A more theoretically motivated query strategy is query-by-committee (QBC) (Seung et al., 1992). The QBC approach involves maintaining a committee of models, which are all trained on the current set of labeled samples, but represent competing hypotheses. Each committee member votes on the labels of query candidates and the query considered most informative is the one on which they disagree most. It can be shown that this is the query that potentially gives the largest reduction in the space of hypotheses (models) consistent with the current training dataset (version space). A related approach is Bayesian active

Table 1: **Development datasets.** ALEX is a toy dataset given for illustrative purpose. The other datasets match the final datasets by application domain (see text).

Dataset	Feat. type	Feat. num.	Sparsity (%)	Missing (%)	Pos. lbls (%)	Tr & Te num.
ALEX	binary	11	0	0	72.98	5000
HIVA	binary	1617	90.88	0	3.52	21339
IBN SINA	mixed	92	80.67	0	37.84	10361
NOVA	binary	16969	99.67	0	28.45	9733
ORANGE	mixed	230	9.57	65.46	1.78	25000
SYLVA	mixed	216	77.88	0	6.15	72626
ZEBRA	continuous	154	0.04	0.0038	4.58	30744

learning. In the Bayesian setting, a prior over the space of hypotheses is revised to give the posterior after seeing the data. Bayesian active learning algorithms (Tong and Koller, 2000, for example) maximize the expected Kullback-Leibler divergence between the revised posterior distribution (after learning with the new queried example) and the current posterior distribution given the data already seen. Hence this can be seen both as an extension of the expected model change framework for a Bayesian committee and a probabilistic reduction of hypothesis space. A more direct criterion of optimality seeks queries that are expected to produce the greatest reduction in generalization error, i.e. the error on data not used for training drawn from $P(\mathbf{x}, y)$ (expected error reduction). Cohn and collaborators (Cohn et al., 1996) proposed the first statistical analysis of active learning, demonstrating how to synthesize queries that minimize the learner's future error by minimizing its variance. However, their approach applies only to regression tasks and synthesizes queries de novo. Another more direct, but very computationally expensive approach is to tentatively add to the training set all possible candidate queries with one of the opposite labels and estimate how much generalization error reduction would result by adding it to the training set (Roy and McCallum, 2001). It has been suggested that uncertainty sampling and QBC strategies are prone to querying outliers and therefore are not robust. The information density framework (Settles et al., 2008b) addresses that problem by considering instances for which not only the predictions are uncertain, but which also lie in regions of high density, in order to find the most informative instances This last approach addresses the problem of monitoring the trade-off between exploration and exploitation. Methods such as "uncertainty sampling" often yield mediocre results because they stress only "exploitation" while "random sampling" performs only "exploration". For a more comprehensive survey, see (Settles, 2009).

2. Datasets and evaluation method

The challenge was comprised of two phases: a development phase (Dec. 1, 2009 - Jan. 31, 2010) during which the participants could develop and tune their algorithms, using six development datasets and a final test phase (Feb. 3, 2010 - Mar. 10, 2010). Six new datasets

Table 2: **Final test datasets.** The fraction of positive labels was not available to the participants.

Dataset	Feat. type	Feat. num.	Sparsity (%)	Missing (%)	Pos. lbls (%)	Tr & Te num.
A	mixed	92	79.02	0	13.35	17535
B	mixed	250	46.89	25.76	9.14	25000
C	mixed	851	8.6	0	8.1	25720
D	binary	12000	99.67	0	25.52	10000
E	continuous	154	0.04	0.0004	9.04	32252
F	mixed	12	1.02	0	7.58	67628

were provided for the final test phase. One of the exciting aspects of the organization of this challenge has been the abundance of data, which clearly signals that this problem is ripe for study, and solving it will have immediate impact. Several practitioners in need of good active learning solutions offered to donate data from their study domain. The data statistics are summarized in Table 1 for the development datasets and Table 2 for the final test sets. The data may be downloaded from: `http://www.causality.inf.ethz.ch/activelearning.php?page=datasets#cont` . All datasets are large (between 20000 and 140000 examples). We selected six different application domains, illustrative of the fields in which active learning is applicable: Chemo-informatics, embryology, marketing, text ranking, handwriting recognition, and ecology. The problems chosen offer a wide range of difficulty levels, including heterogeneous noisy data (numerical and categorical variables), missing values, sparse feature representation, and unbalanced class distributions. All problems are two-class classification problems.

The datasets from the final phase were matched by application domain to those of the development phase. During the challenge, the final test datasets were named alphabetically, so as not to make that matching explicit. However, the final datasets have mnemonic nicknames (unknown to the participants during the competition) such that the correspondences are more easily remembered:

- **A** is for AVICENNA, the Latin name of IBN SINA. This is a handwriting recognition dataset consisting of Arabic manuscripts by the 11^{th} century Persian author Ibn Sina.
- **B** is for BANANA, a fruit like ORANGE. This is a marketing dataset donated by Orange Labs.
- **C** is for CHEMO, this is a chemo-informatics dataset for the problem of identifying molecules that bind to pyruvate kinase. It is matched to HIVA, another chemo-informatics dataset for identifying molecules active against the HIV virus.
- **D** is for DOCS. This is a document analysis dataset, matched with NOVA.
- **E** is for EMBRYO, an embryology dataset, matched with ZEBRA.
- **F** is for FOREST, an ecology dataset. The problem is to find forest cover types as in the SYLVA dataset.

A report describing the datasets is available (Guyon et al., 2010).

The protocol of the challenge was simple. The participants were given unlabeled data and could purchase labels on-line for some amount of virtual cash. In addition, the index of a single positive example was given to bootstrap the active learning process. Participants were free to purchase batches of labels of any size, by providing the sample numbers of the labels they requested. To allow the organizers to draw learning curves, the participants were asked to provide prediction values for all the examples every time they made a purchase of new labels. Half of each dataset could not be queried and was considered a test set.

The prediction performance was evaluated according to the Area under the Learning Curve (ALC). A learning curve plots the Area Under the ROC curve (AUC) computed on all the samples with unknown labels, as a function of the number of labels queried. To obtain our ranking score, we normalized the ALC as follows:

$$globalscore = (ALC - Arand)/(Amax - Arand)$$

where $Amax$ is the area under the best achievable learning curve and $Arand$ is the area under the average learning curve obtained by making random predictions. See `http://www.causality.inf.ethz.ch/activelearning.php?page=evaluation#cont` for details.

An obvious way of "cheating" would have been to use an "associate" or register under an assumed name to gain knowledge of all the labels, then submit results under one's real name. Preventing this kind of cheating is very difficult. We resorted to the following scheme, which gives us confidence that the participants respected the rules of the challenge:

- The participants had to register as mutually exclusive teams for the final phase. The membership of teams were manually verified.
- The team leaders had to electronically sign an agreement that none of his team member would attempt to exchange information about the labels with other teams.
- We announced that we would perform some verification steps to deter those participants who would otherwise be tempted to cheat.
- For one of the datasets (dataset A), we provided a different set of target labels to each participant, without letting them know. In this way, if two teams exchanged labels, their resulting poor performance should be suspicious. This would alert us and require us to proceed with further checks, such as asking the participants to provide their code.

Our analysis of the performances on dataset A did not give us any reason to suspect that anyone had cheated (see Appendix A for details). During the verification phase we asked the participants to repeat their experiments on dataset A, this time providing the same labels to everyone. Those are the results provided in the result tables.

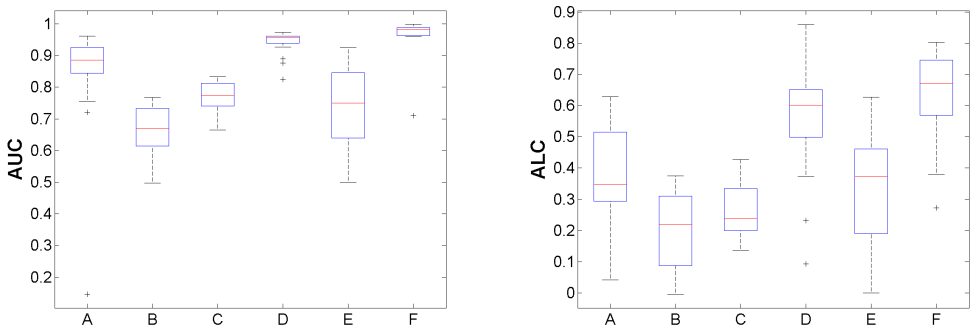
Table 3: Best benchmark results for the development and final datasets. The mean rank on test datasets is 3.833.

Dataset	Experiment	Classifier	Strategy	AUC	ALC	Rank
HIVA	gcchiva4	Naïve Bayes	Bayesian	0.805504	0.328535	—
IBN_SINA	gccibnsina1	Linear KRR	Random	0.978585	0.813690	—
NOVA	gccnova1	Linear KRR	Random	0.991841	0.715582	—
ORANGE	gccorange1	Linear KRR	Random	0.814340	0.283319	—
SYLVA	gccsylva1	Linear KRR	Random	0.996240	0.921228	—
ZEBRA	gcczebra1	Linear KRR	Random	0.785913	0.416948	—
Avicena	gccA004v	Linear KRR	Random	0.883768	0.586001	3
Banana	gccb1	Linear KRR	Passive	0.720291	0.370762	3
Chemo	gccc4	Linear KRR	Random	0.814450	0.301776	5
Docs	gccd2	Linear KRR	Random	0.962951	0.651222	6
Embryo	gcce1	Linear KRR	Passive	0.773262	0.496610	5
Forest	gccf2	Linear KRR	Random	0.954557	0.821711	1

3. Baseline results

We uploaded baseline results to the website for the development datasets under the name "Reference". The majority of these submissions used linear kernel ridge regression as the base classifier, where the regularisation parameter was tuned by minimising the virtual leave-one-out cross-validation estimate of the sum-of-squared errors, i.e. Allen's PRESS statistic (Allen, 1974; Saadi et al., 2007). The best results, shown in Table 3, were generally obtained using either passive learning (all labels queried at once) or random active learning, where samples are chosen at random for labeling by the oracle. The one exception was the HIVA dataset, where a naïve Bayes classifier was found to work well, with a Bayesian active learning strategy, where samples were submitted to be labelled in decreasing order of the variance of the posterior prediction of the probability of class membership. For a full description of the baseline methods, see Cawley (2010).

It is interesting to note that a simple linear classifier, with passive learning, or random active learning strategies performs so well (as indicated by the rankings, shown in Table 3). The overall rank of 3.833 for those submissions is subject to a strong selection bias resulting from choosing the best of the four baseline submissions for each benchmark. A more realistic overall ranking is obtained by looking at the results for linear KRR with random active learning (gccA002v, gccb2, gccc2, gccd2, gcce2 and gccf2), which gives an overall rank of 4.667, which would have been sufficient to achieve runner-up status in the challenge. This shows that active learning is an area where further research and evaluation may be necessary to reliably improve on such basic strategies.



(*a*) (*b*)

Figure 1: **Distribution of results**. *We show box-whiskers plots for the various datasets. The red line represents the median, the blue boxes represent the quartiles, and the whiskers represent the range, excluding some outliers plotted individually as crosses. (a) Area under the ROC curve for the last point on the learning curve. (b) Area under the learning curve.*

4. Challenge results

The challenge attracted a large number of participants. Over 300 people registered to gain access to the data and participate in the development phase. For the final test phase 30 teams were formed, each comprised of between 1 and 20 participants. This level of participation is remarkable for a challenge that requires a deep level of commitment for participation because of the specialized nature of the problem and the iterative submission protocol (participants must query for labels and make predictions by interacting with the website).

It is difficult to make a fair assessment of the results on development data sets because the participants were allowed to perform multiple experiments on the same dataset, the knowledge of the labels obtained in previous experiments may have implicitly or explicitly been used in later experiments. Hence we report only the results on the final test sets for which the participants were only allowed to perform one single experiment. The distribution of performance with respect to AUC and ALC are shown in Figure 1. The results of the top ranking teams for each final dataset are found in Table 4. We also plotted the learning curves of the top ranking participants overlaid on top of all the other learning curves (Figures 2 to 7).

We encouraged the participants to enter results on multiple datasets by exponential scaling of the prizes with the number of wins. However, no team ended up winning on more than one dataset. The remaining prize money has been used to provide travel grants to encourage the winners to attend the workshop. For those participants who entered results on all 6 datasets, we performed a global ranking according to their average rank on the individual datasets. The overall winner by average rank (average rank 4.2) is the Intel team (Alexander Borisov and Eugene Tuv), who already ranked among the top entrants in

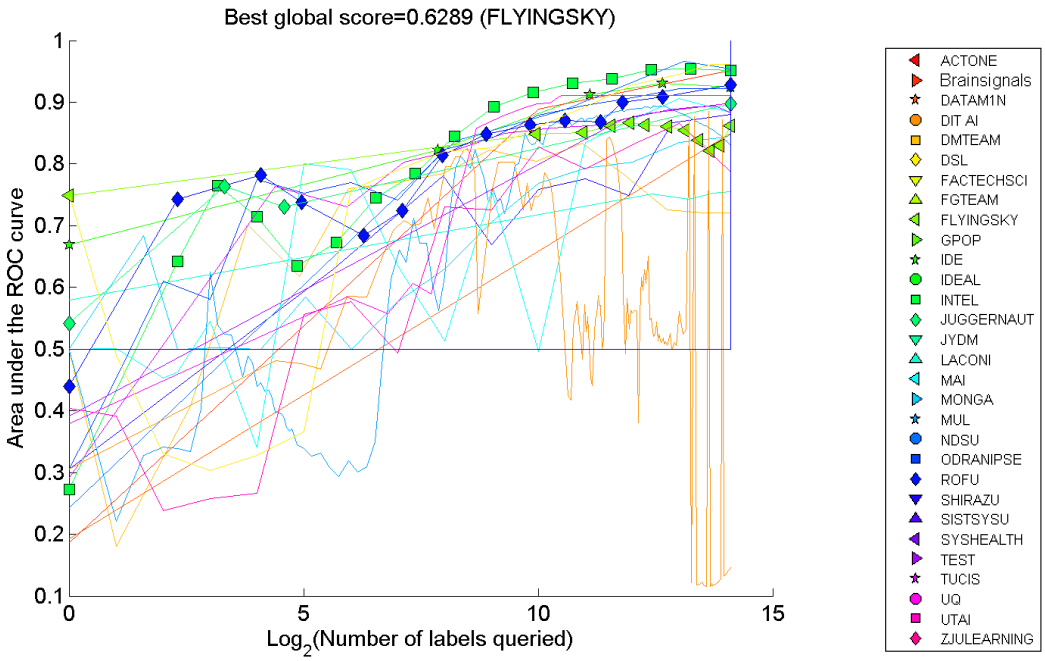


Figure 2: *Learning curves for dataset A.*

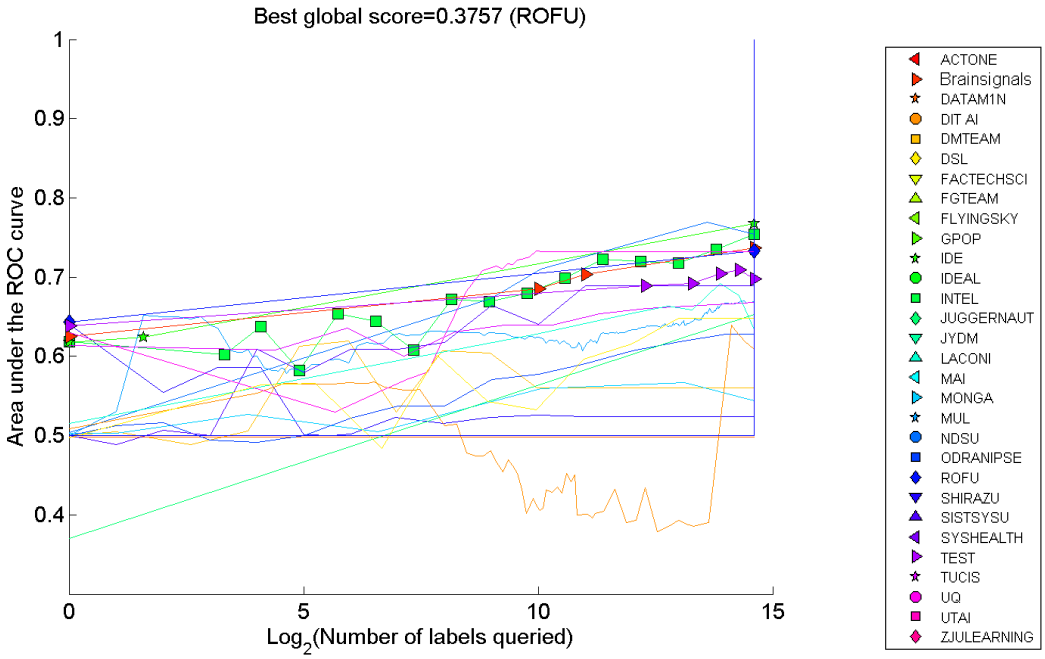


Figure 3: *Learning curves for dataset B.*

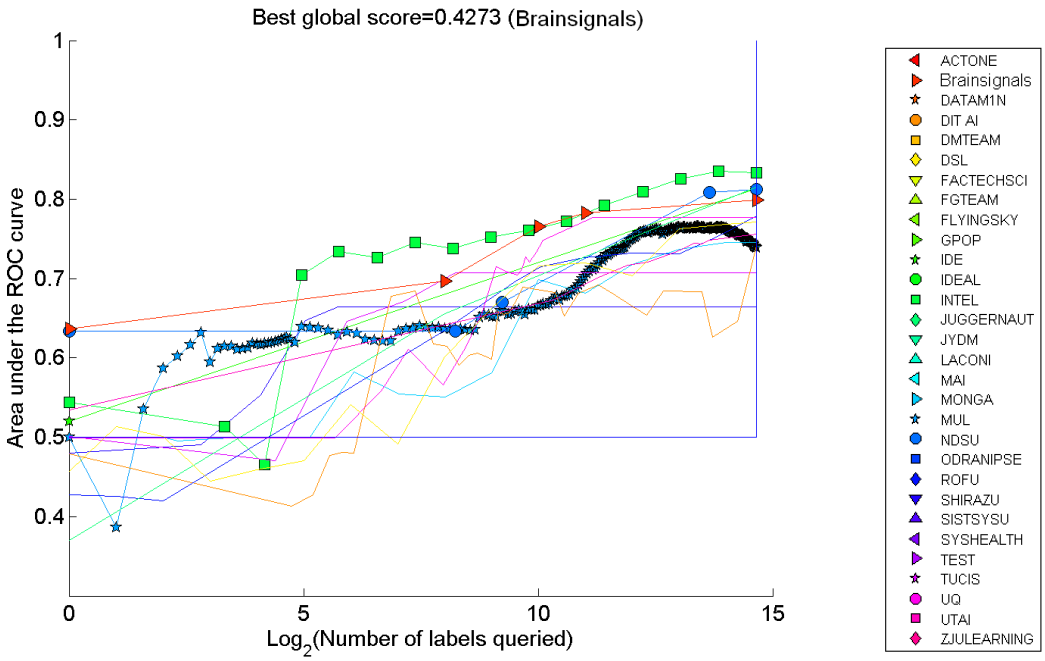


Figure 4: *Learning curves for dataset C.*

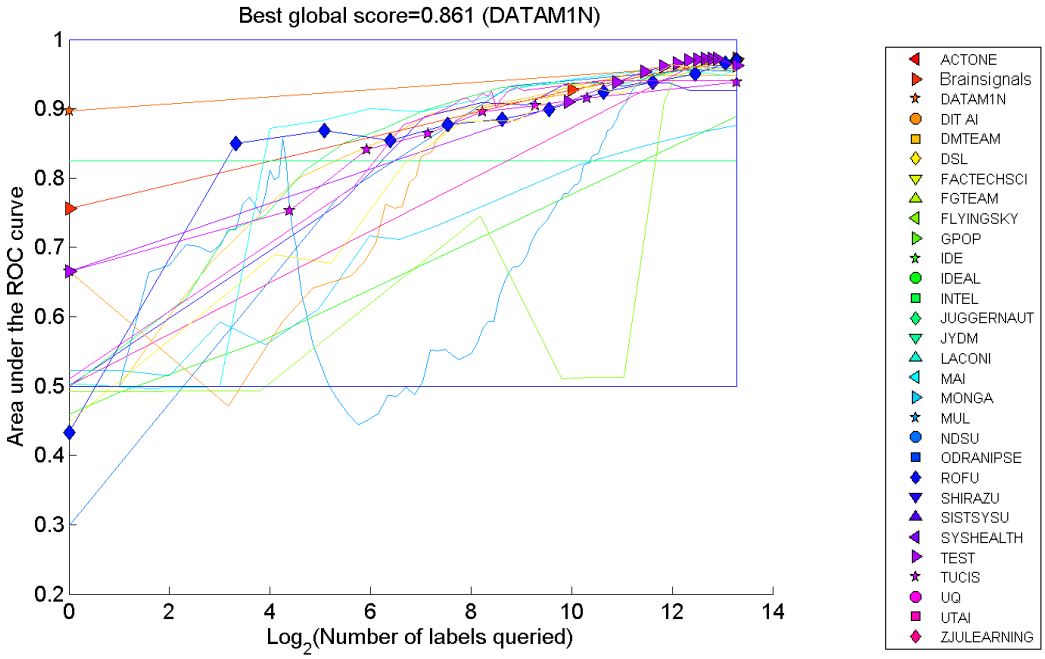


Figure 5: *Learning curves for dataset D.*

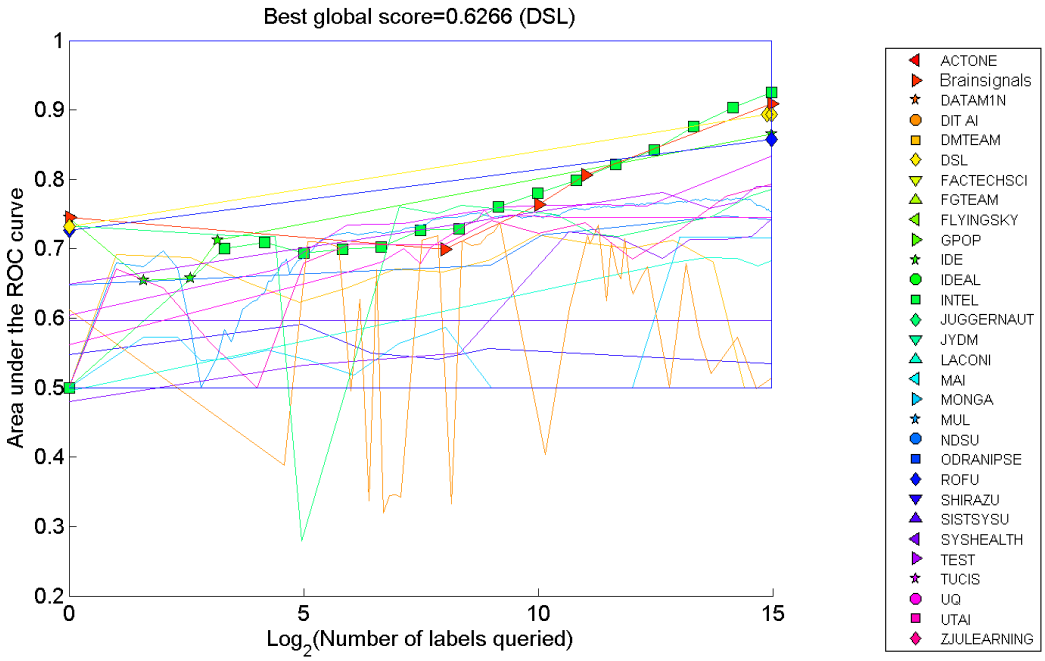


Figure 6: *Learning curves for dataset E.*

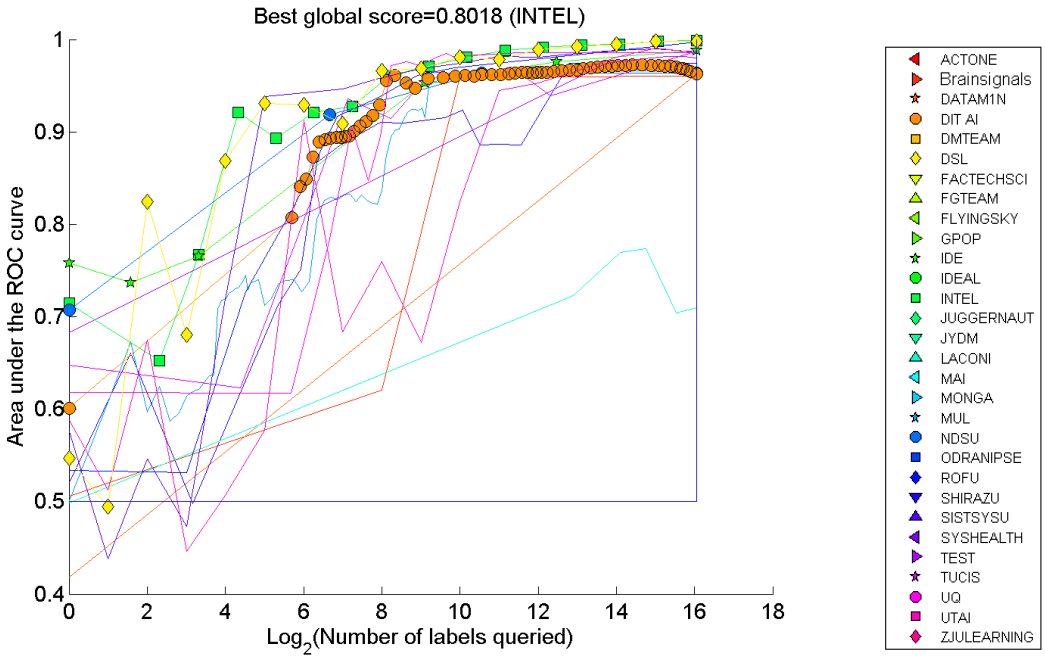


Figure 7: *Learning curves for dataset F.*

several previous challenges. The runner up by average rank (average rank 4.8) is the ROFU team of National Taiwan University (Ming-Hen Tsai and Chia-Hua Hu). Other members of this research group headed by Chin en Lin have also won several other machine learning challenges. The next best ranking teams are IDE (average rank 5.7) and Brainsignals (average rank 6.7). The team TEST (Zhili Wu) made entries on only 5 datasets, but did also very well (average rank 6.4).

We briefly comment on the methods used by these top entrants:

- The Intel team used a probabilistic version of the query-by-committee algorithm (Freund et al., 1997) with boosted Random Forest classifiers as committee members (Borisov et al., 2006). The batch size was exponentially increasing, disregarding the estimated model error. Some randomness in the selection of the samples was introduced by randomly sampling examples from a set of top candidates. No use was made of unlabeled data. The technique used generated very smooth learning curves and reached high levels of accuracy for large numbers of training samples. The total run time on all development datasets on one machine is approximately 6-8 hours depending on model optimization settings. The method does not require any pre-processing, and naturally deals with categorical variables and missing values. The weakness of the method is at the beginning of the learning curve. Other methods making use of unlabeled data perform better in this domain.

- The ROFU team used Support Vector Machines (SVMs) (Boser et al., 1992) as a base classifier and a combination of uncertainty sampling and query-by-committee as active learning strategy (Tong and Koller, 2002). They made use of the unlabeled data (Sindhwani and Keerthi, 2005, 2006) and they avoided sampling near points already labeled. No active learning was performed on dataset B and E (inferring from the development dataset results that active learning would not be beneficial for such data). The method employed for learning from unlabeled data must not have been very effective because the results at the beginning of the learning curves are quite bad on some datasets (dataset C and F), but the performances for a large number of labeled examples are good. The authors report that using SVMs is fast so they could optimize the hyper-parameters by cross-validation.

- The IDE team used hybrid approaches. In the first few queries, they used semi-supervised learning (*i.e.,* make use of both labeled and unlabeled data with cluster-and-label and self-training strategies), then switch to supervised learning. For active learning, they combined uncertainty sampling and random sampling. Logistic regression and k-means clustering were used when the number of labeled examples is very small (≤ 100). Boosted decision trees were used when the amount of labeled examples is large (≥ 500). The authors think that getting the representative positive examples in the first few queries was key to their success. Indeed, the authors had very few points on their learning curve and their performance for small number of examples on several datasets determined their good rank in the challenge on most datasets.

- The TEST team did not make any attempt to use unlabeled data and queried a large number of labels at once ($\simeq$ 2000 examples). Hence its good performance in the challenge are essentially based on the second part of the learning curve. This

is a conservative strategy that takes no risk in the first part of the learning curve, which from our point of view was the most interesting. The classifier used is logistic regression and the active learning strategy is uncertainty sampling. The learning curves are smooth. Hence, the use of uncertainty sampling with an sufficiently large initial pool of example seems to be a viable strategy.

- The Brainsignals team did not perform active learning per se. The strength of the entries made and their good ranking in the challenge stem from a good first point in the learning curve obtained with a semi-supervised learning method based on spectral clustering (Zhu, 2005). Then very few points are made on the learning curve at 256, 1024, and all samples. Random sampling was used and classical model selection techniques with cross-validation to select among ensemble of decision trees, linear classifiers, and kernel-based classifiers.

Several participants found that uncertainty sampling and query-by-committee, without introducing any randomness in the selection process, may perform worse than random sampling (see also Section 3 on baseline methods). To illustrate how things can go wrong when strictly using uncertainty sampling, we show in Figure 8 the learning curve of one team on dataset D who used such strategy for active learning. There is a catastrophic decrease in performance in the middle of the learning curve. Query by committee performs better than uncertainty sampling both in randomized and non-randomized settings. Techniques for pro-actively sampling in regions with low densities of labels were reported not to yield significant improvements. Ensemble methods combined with query-by-committee active learning strategies yielded smooth learning curves. Good performances for very small number of examples (≤ 100) were achieved only by teams using semi-supervised learning strategies.

4.1. Methods employed

In what follows, for each category of methods (active learning, pre-processing, feature selection, etc.) we report the fraction of participants having used each method. Note that the sums of these numbers do not necessarily add up to 100%, because the methods are not mutually exclusive and some participants did not use any of the methods.

We analyzed the information provided by the participants in the fact sheets:

- **Active Learning and use of Unlabeled Data:** In Figure 9, we show a histogram of the type of active learning methods employed. **Most of the participants used "uncertainty sampling"** as part of their strategy (81%) or random sampling (47%). Query-by-committee was also very popular (38%). Interestingly no participant used Bayesian active learning, **20% of the participants used no active learning at all** but **57% made use of unlabeled data** – not shown on the figure.

- **Pre-processing and Feature Selection:** In Figure 10 we show histograms of the algorithms employed for pre-processing, feature selection. Few participants did not use any pre-processing (14%) and most participants performed **data normalizations** (71%). A large fraction of the participants used **replacement of missing values** by the mean or the median or a fixed value (43%). Principal Component Analysis (PCA)

Table 4: Result tables for the top ranking teams.

Dataset A		
Team	AUC (Ebar)	ALC
Flyingsky	0.8622 (0.0049)	0.6289
IDE	0.9250 (0.0044)	0.6040
ROFU	0.9281 (0.0040)	0.5533
JUGGERNAUT	0.8977 (0.0036)	0.5410
Intel	0.9520 (0.0045)	0.5273

Dataset B		
Team	AUC (Ebar)	ALC
ROFU	0.7327 (0.0034)	0.3757
IDE	0.7670 (0.0038)	0.3754
Brainsignals	0.7367 (0.0043)	0.3481
TEST	0.6980 (0.0044)	0.3383
Intel	0.7544 (0.0044)	0.3173

Dataset C		
Team	AUC (Ebar)	ALC
Brainsignals	0.7994 (0.0053)	0.4273
Intel	0.8333 (0.0050)	0.3806
NDSU	0.8124 (0.0050)	0.3583
IDE	0.8137 (0.0051)	0.3341
MUL	0.7387 (0.0053)	0.2840

Dataset D		
Team	AUC (Ebar)	ALC
DATAM1N	0.9641 (0.0033)	0.8610
Brainsignals	0.9717 (0.0033)	0.7373
ROFU	0.9701 (0.0032)	0.6618
TEST	0.9623 (0.0033)	0.6576
TUCIS	0.9385 (0.0037)	0.6519

Dataset E		
Team	AUC (Ebar)	ALC
DSL	0.8939 (0.0039)	0.6266
ROFU	0.8573 (0.0043)	0.5838
IDE	0.8650 (0.0042)	0.5329
Brainsignals	0.9090 (0.0039)	0.5267
Intel	0.9253 (0.0037)	0.4731

Dataset F		
Team	AUC (Ebar)	ALC
Intel	0.9990 (0.0009)	0.8018
NDSU	0.9634 (0.0018)	0.7912
DSL	0.9976 (0.0009)	0.7853
IDE	0.9883 (0.0013)	0.7714
DIT AI	0.9627 (0.0017)	0.7216

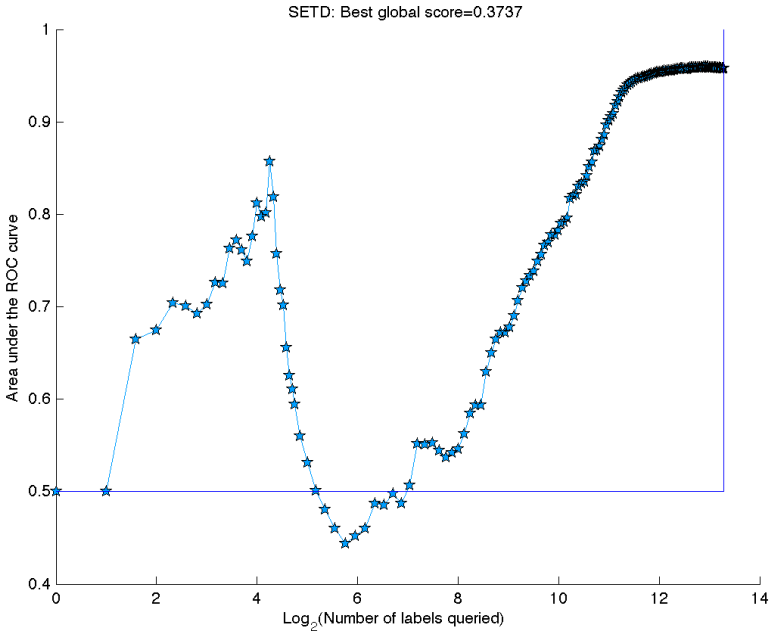


Figure 8: *Example of learning curves for dataset D using the uncertainty sampling strategy.*

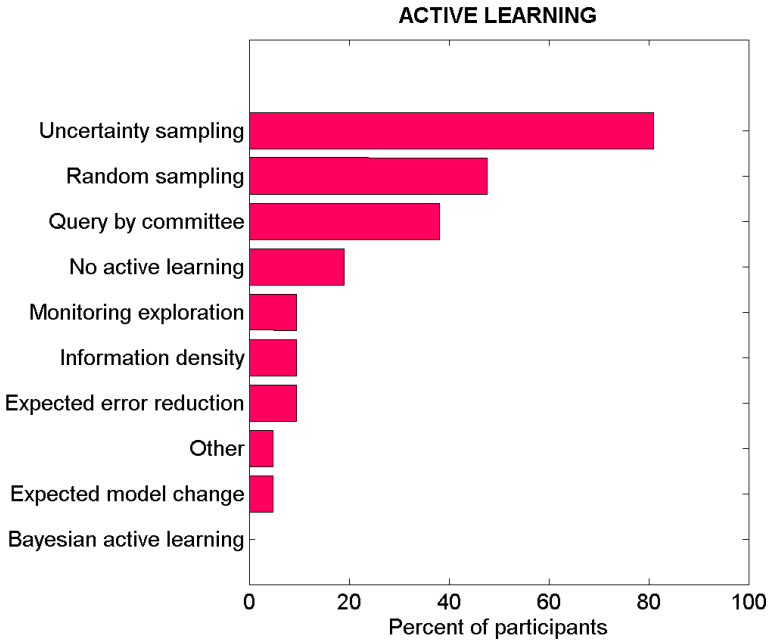


Figure 9: *Active Learning Methods Employed.* Also worth noting: 57% of the participants used unlabeled data (with or without active learning).

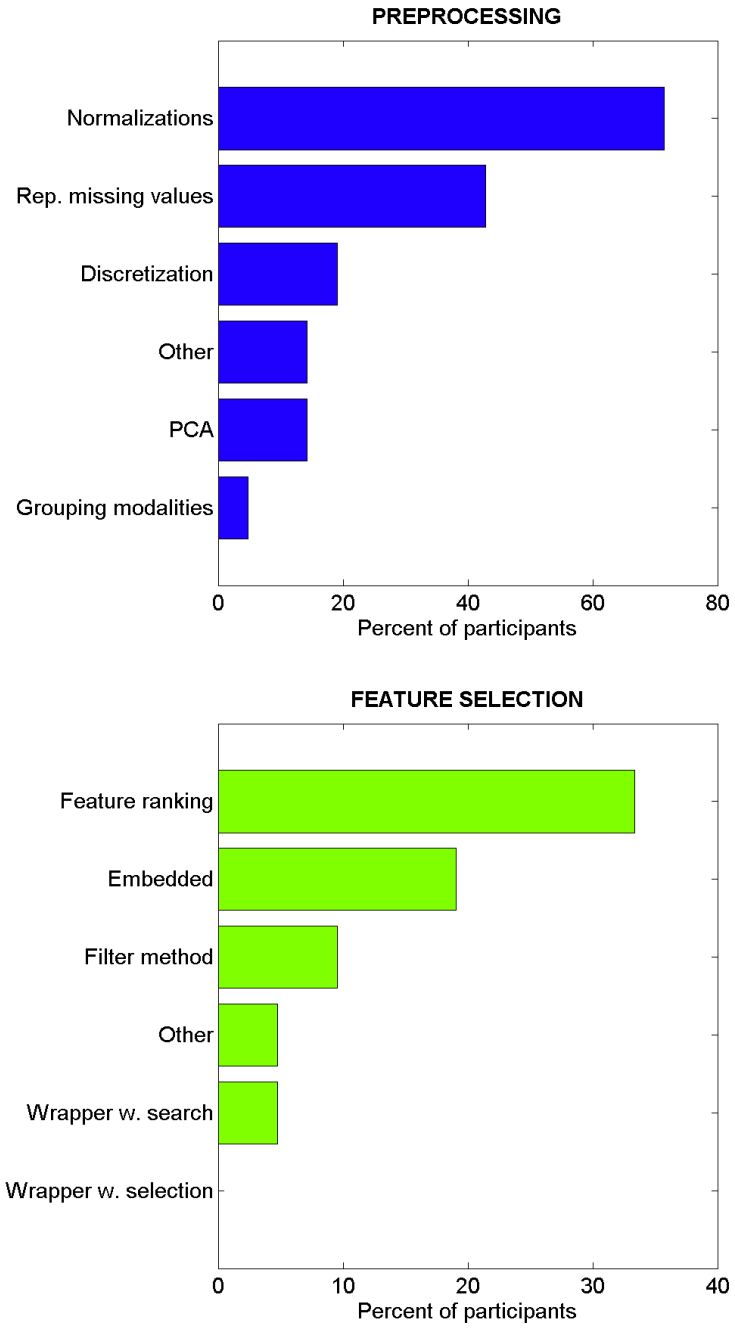


Figure 10: *Preprocessing and feature selection.*

was seldom used and reported not to bring performance improvements. Discretization and other types of pre-processing were not used very much in this challenge. About half of the participants (52%) used some form of **feature selection**. **Feature ranking methods** were the most widely used feature selection methods (33%) and other filter methods were also used (9%). Compared to previous challenges, many participants used embedded methods (19%), but consistent with previous challenges, wrapper methods with extensive search were unpopular.

- **Classification algorithm and model selection:** In Figure 11 we show the classification algorithms used. Most participants either used as part of their design a **linear classifier** (62%) or a **non-linear kernel method** (43%). **Decision trees** and **Naïve Bayes** are also quite popular (each have a 33% usage), while all the other methods, including neural networks, are much less popular (less than 20% usage). The statistics on loss function usage reveal the increasing popularity of the **logistic loss** (38%). The **hinge loss** used for SVMs remains popular (29%). Other loss functions, including the exponential loss (boosting) and the square loss (ridge regression and LS-SVMs) each have less than 20% reported usage. We also collected statistics not shown on those figures about regularizers, ensemble methods and model selection. Consistent with the popularity of linear and kernel methods in this challenge, a large fraction of the participants used regularization, with 43% usage of **2-norm regularizers** and 19% usage of 1-norm regularizers. Most participants made use of some ensemble method (about 80%) including 33% usage of bagging and 29% usage of boosting. The wide use of ensemble methods may explain the relatively low use of model selection methods (62%); cross-validation methods such as K-fold and leave-one-out remain the most popular (43%).

We also analyzed the fact sheets with respect to the software and hardware implementation:

- **Hardware:** Many participants made use of parallel implementations (67% used multiple processor computers and 24% ran experiments in parallel). Memory usage was relatively modest (38% used less than 2 GB and 33% less than 8 GB).

- **Software:** Few participants are Mac users; most use either Windows or Linux (about half and half). In terms of programming languages (Figure 12), higher level languages (Matlab, R, and Weka) are most popular, Matlab coming first (67%), but C/C++ and Java are also used significantly. About 70% of the participants wrote their own code, 30% of which are making it freely available.

The amount of human effort involved in adapting the code to the problems of the challenge varied but was rather significant because about half of the participants reported spending less than two weeks of programming while half reported more than 2 weeks. The amount of time spent experimenting (computer effort) was distributed similarly. While the majority of participants reported having enough development time (67%) and enough test time (62%), a large fraction of participants ran out of time to do what they wanted.

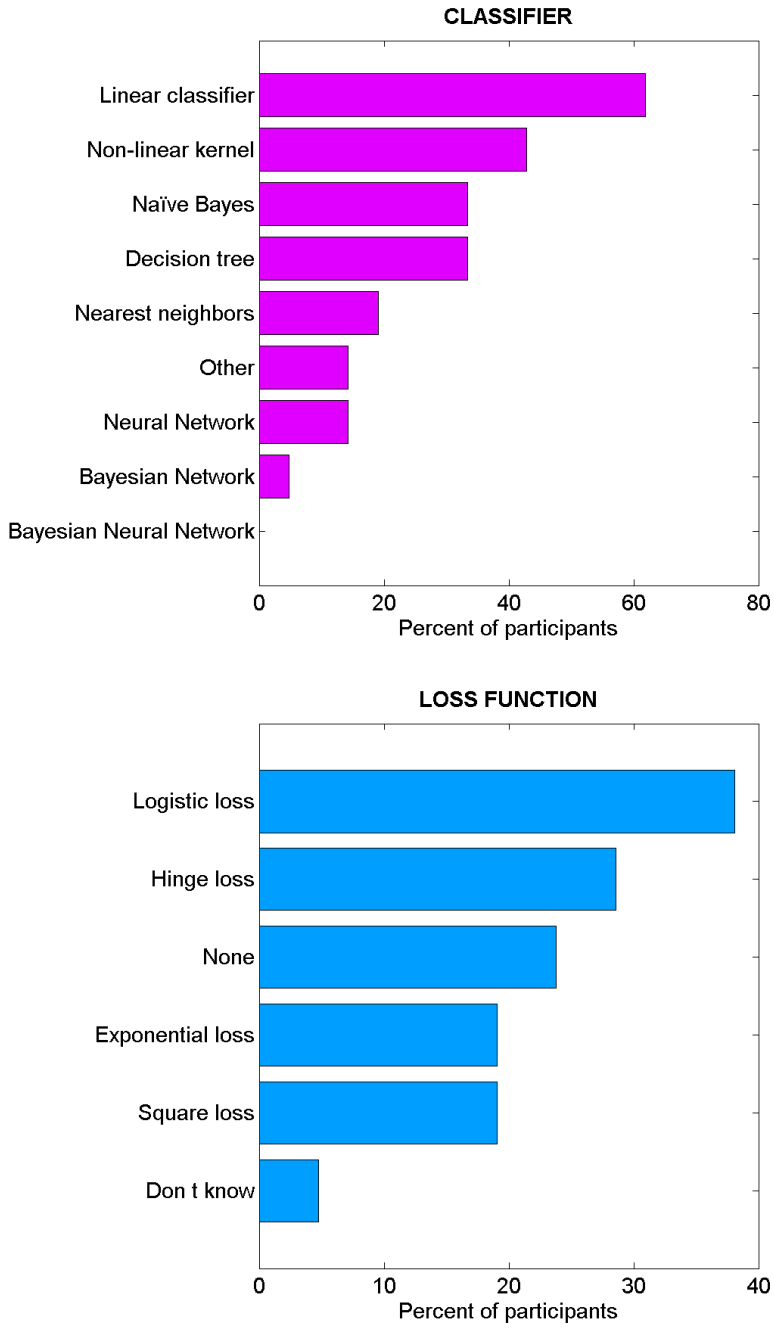


Figure 11: *Classification algorithms and loss functions.*

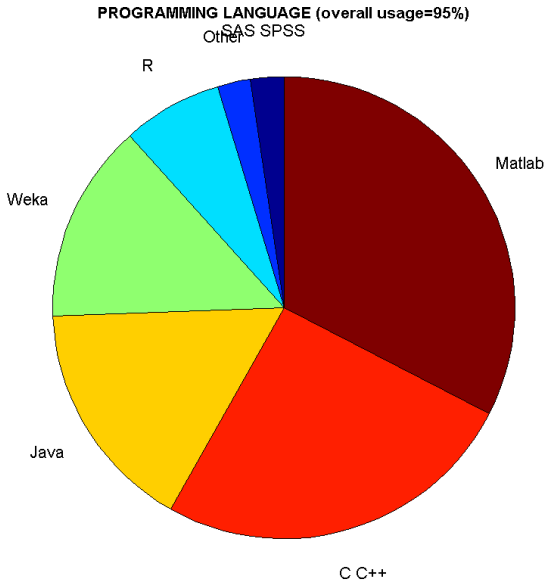


Figure 12: *Programming Languages.*

5. Discussion

5.1. Statistical significance of the results

One of the aims of this competition was to try to assess the quality of active learning methods in an unbiased manner. To this end, it is important to examine whether the differences between the top ranking methods and the lowest ranking methods can be attributed to chance or whether there are statistically significant differences.

We performed a statistical test, specifically designed for settings in which multiple classifiers are being tested with multiple datasets, namely the Friedman test (Demšar, 2006). The ranks of the algorithms on each dataset are used. A tabulated statistic is derived from the average ranks of the algorithms to test the null hypothesis that all algorithms are equivalent so their ranks should be equal. The above tests call for a full score matrix (teams vs. datasets) so we restricted our analysis to the set of 11 teams that submitted the results for all 6 datasets. We performed the tests with a significance level of $\alpha = 0.05$. The results of the Friedman test turned out to be highly significant (with a p-value of 0.0019), so there are significant differences in performances among the teams.

Since this first test was successful (i.e., the null hypothesis of equivalence between algorithms was rejected) we ran a post-hoc test as recommended by Demšar (2006): the Nemenyi test. That test looks for significant differences between the performances of a given algorithm and those of the others. Accordingly, only two teams proved to be significantly different, namely the first (INTEL group) and the last (DIT AI Group).

5.2. Post-challenge experiments

Lemaire and Salperwyck (2010) performed systematic post-challenge experiments to assess a number of classifiers used by the challenge participants in a controlled manner. The authors decoupled the influence of the active learning strategy from that of the strength of the classifier by simply using a random sampling strategy (passive learning). To reduce the variance on the results, each experiment was repeated ten times for various drawings of the training samples, but always starting from the same seed example provided to the participants. Learning curves were drawn by averaging the performances obtained for training set sizes growing exponentially $2^1, 2^2, 2^3, 2^4, ...$ The performances of the various classifiers were compared with the ALC and the AUC of the final classifier trained on all the examples. The authors compared several variants of Naive Bayes, logistic regression, several decision trees and ensembles of decision trees. The study allowed the authors to confirm trends observed in the analysis of the challenge and reported elsewhere in the literature:

- Tree classifiers perform poorly, particularly for small number of examples, but ensembles or trees are among the best performing methods.
- Generative models illustrated by naive Bayes methods perform better than discriminative models for small numbers of examples.

Interestingly, the challenge participants did not capitalize on the idea of switching classifier between when progressing through the learning curve and performed model selection globally using the ALC for a fixed classifier. The study reveals a ranking of classification methods similar to that of the challenge, using only passive learning. This observation suggests that the choice of the classifier may have been a more of a determining factor in winning the challenge than the use of a good active learning strategy. The relative efficacy of active learning strategies for given classifiers remains to be systematically assessed and will be the object of further studies.

5.3. Comparison of the datasets

The datasets chosen presented a range of difficulties, as illustrated by the AUC results obtained at the end of the learning curve (Figure 1-a). The median values are ranked as follows: $F > D > A > C > E > B$. The best AUC results are uniformly obtained by ensembles of decision trees for all datasets[1]. For dataset F (Ecology), the best AUC result (0.999) is obtained by the Intel team using a combination of boosting and bagging of decision trees. This confirms results obtained with this dataset in previous challenges in which ensembles of tree classifiers won (Guyon et al., 2006). However, other methods including ensembles of mixed classifiers (NDSU) and SVMs (DSL and DIT AI) also work well. For dataset D (Text), the best AUC performance is also high (0.973) and it is obtained by the Intel team, but the profile of other top ranking methods is rather different: it includes mostly linear classifiers, which are close contenders. For dataset A (Handwriting Recognition), the best final AUC is also obtained by the Intel team with ensembles of decision trees (0.952), but other methods including heterogeneous ensembles and kernel methods such as SVMs

1. The results in Table 4 correspond to the top ranking participants by ALC. Some of the best AUC results are not included in that table.

work well too. The best results on dataset C (Chemoinformatics) are also obtained by the Intel team with the same method (0.833) and are significantly better than the next best result obtained with an heterogeneous ensemble of classifiers. Dataset E (Embryology) exhibits the largest variance in the results, hence a good model selection is crucial for that dataset. The AUC result of the Intel team with ensembles of decision trees (0.925) is statistically significantly better than the next best result. The best AUC result (0.766) on the hardest dataset (dataset B, Marketing) was obtained by a different team (IDE) but also with an ensemble of tree classifiers (the Intel result comes close).

Another point of comparison between datasets is the shape of the learning curves and the success of active learning strategies. According to Figure 1-b, the ranking of dataset median performance with respect to ALC is similar to the ranking by AUC $F > D > E > A > C > B$ (only the position of E differs), but the variance is a lot higher. This shows that when it comes to active learning, it is easy to do things wrong! The analysis of Figures 2 to 7 provides some insight into the learnability of the various tasks from few examples. For dataset F, the learning curve of the best ranking participants climbs quickly and with as little as 4 examples a performance larger than AUC=0.9 is achieved. For dataset D, making a good initial semi-supervised entry using the seed example was critical. Then, the learning curves climb rather slowly. This can be explained by the relatively polarized separation chosen for the task: computer related topics *vs.* everything else. Learning from just one example came a long way. Rather similar learning curves are obtained for dataset E. The teams who did not perform semi-supervised learning got much worse results in the first part of the learning curve. For dataset A, there is a lot of variance in the first part of the learning curve until about 32 examples are given. This high variance was also observed in the post-challenge experiments. It may be due to the high heterogeneity of the classes. For dataset C, the Intel team who obtained the best final AUC and has a learning curve dominating all others starting at 32 examples did not do well for the small number of examples. Other teams including MUL used semi-supervised learning strategies and as a result the learning curve climbed much faster. Dataset B presents the flatest learning curves. All top ranking learning curves start with an AUC between 0.61 and 0.65 and end up with and AUC between 0.68 and 0.72. Virtually all active learning strategies are found among the top ranking participants for that dataset but, according to our post-challenge experiments, random sampling does just as well.

5.4. Compliance with the rules of the challenge

We struggled to find a protocol that would prevent violations of the rules of the challenge. We needed a mechanism to ensure that the participants could not gain knowledge of the labels unless they had legitimately purchased them (for virtual cash) from the website. This implied that entries made to gain access to labels could not be corrected and that the participating teams could not exchange information about labels. Inevitably, in the course of the challenge, some participants made mistakes in their submissions that they could not correct. The verification process that we implemented (see Appendix A) was not uniformly well received because some participants felt that, had they known in advance that the verification round counted for ranking, they would have spent more effort on it. Hence, this generated some frustration. However, the validation process gave us confidence

that the participants respected the rules, which is important in order to be able to draw valid conclusions from the analysis of the challenge.

5.5. Lessons learned about the challenge protocol

This challenge is one of the most sophisticated challenges that we have organized because it involved a complex website back-end to handle the queries made by the participants and manage the status of their on-going experiments. Some participants complained that the manual query submission/answer retrieval through web form was very inconvenient and time consuming. In future challenges we plan to automate that process by providing scripts to make submissions.

The nature of the challenge also made the use of part of the dataset for validation impossible during the development period. We had to resort to using different datasets during the development period and the final test period. However, due to limited resources, not all the final datasets were completely different from their matched development dataset. For dataset A, we added more samples and changed the targets, for dataset B, we changed the features and the targets but the samples were the same, for dataset C, the task was entirely new. For dataset D, we changed the features and the targets, but the samples were the same, for datasets E and F, we sub-sampled the data differently and provided different targets. For F, we also changed the features. For all datasets, the order of samples and features were randomized. In this way, the participants could not re-use samples or models from the development phase in the final phase. However, even though we did not explicitly match the datasets between the two phases, it was not difficult to figure out the match. Some participants seem to have made use of this information to select strategies of active learning (or decide not to perform active learning at all on certain datasets).

To put most emphasis on learning from few labeled examples, our evaluation metric put higher weight on the beginning of the learning curve by choosing a logarithmic scaling of the x-axis. This was criticized because there is a lot of variance in the first part of the learning curve and can cause some participants to win by chance. This is a valid concern that, outside of the constraints of a challenge, may be addressed by averaging over multiple experiments performed on data sub samples. In the context of a challenge, samples for which labels have been purchases cannot be re-used, so averaging procedures are excluded; instead, we offered the possibility of working on several datasets. The participants who performed consistently well on all datasets are unlikely to have won by chance. In retrospect, giving prizes for individual datasets might have encouraged to use "gambling strategies": some participants provided only two points in the learning curve, the first and the last one. If by chance their first point was good on one of the 6 datasets, they could win one of the 6 prizes. In retrospect, we might have been better off imposing the condition that all the participants return results on all datasets and make a single ranking based on the average rank on all 6 tasks. We could also regularize the performance measure e.g. by a weighted average of performances obtained on different parts of the learning curve, such as to take into account the difference of their variances.

Table 5 details the empirical standard deviations of participants' performances on various datasets for the first and last points on the learning curve. As can be expected, for the majority of the datasets (A, C, D and F) the standard deviation on the first point and is

Table 5: The empirical means, (μ_0, μ_1) and standard deviations (σ_0 and σ) of the performances (AUC) on the first point (no label purchased) and last point (all available labels purchased) of the learning curve, respectively.

Dataset	μ_0	σ_0	μ_1	σ_1
A	0.416	0.164	0.879	0.068
B	0.572	0.063	0.658	0.086
C	0.509	0.080	0.782	0.034
D	0.535	0.122	0.953	0.021
E	0.595	0.106	0.733	0.142
F	0.586	0.101	0.981	0.015

considerably larger that that of the last point; On the remaining datasets (B and E) the differences are small. It may be a coincidence but the two datasets with largest standard deviations are exactly those for which the winning entries did not actually use active learning but used semi-supervised learning to optimize performance on the first point. Analyzing the spread of results by inter-quartile range as a measure for the spread of the performances, which is more resistant to outliers, produces the same picture.

6. Conclusion

The results of the Active Learning challenge exceeded our expectations in several ways. First we reached a very high level of participation despite the complexity of the challenge protocol and the relatively high level of expertise needed to enter the challenge. Second, the participants explored a wide variety of techniques. This allowed us to draw rather strong conclusions, which include that: (1) semi-supervised learning is needed to achieve good performance in the first part of the learning curve, and (2) some degree of randomization in the query process is necessary to achieve good active learning results. These findings have been confirmed in large on-going Monte-Carlo experiments, the initial results of which are presented in Cawley (2010). This challenge proved the viability of using our Virtual Laboratory in challenges requiring users to interact with a data generating system. We intend to improve it to further automate the query submission process and use it in upcoming challenges on experimental design.

Acknowledgments

This project is an activity of the Causality Workbench supported by the Pascal network of excellence funded by the European Commission and by the U.S. National Science Foundation under Grant N0. ECCS-0725746. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. Additional support was provided to fund prizes and travel awards by Microsoft and Orange FTP. We are very grateful to all the members of the causality workbench team for their contributions and in particular

to our co-founders Constantin Aliferis, Greg Cooper, André Elisseeff, Jean-Philippe Pellet Peter Spirtes, and Alexander Statnikov, and to the advisors and beta-testers Olivier Chapelle, Amir Reza Saffari Azar and Alexander Statnikov. We also thank Christophe Salperwyck for his contributions to the post-challenge analyses. The website was implemented by MisterP.net who provided exceptional support. This project would not have been possible without generous donations of data. We are very grateful to the data donors: Chemoinformatics – Charles Bergeron, Kristin Bennett and Curt Breneman (Rensselaer Polytechnic Institute, New York) contributed a dataset, used for final testing. Embryology – Emmanuel Faure, Thierry Savy, Louise Duloquin, Miguel Luengo Oroz, Benoit Lombardot, Camilo Melani, Paul Bourgine, and Nadine Peyriéras (Institut des systèmes complexes, France) contributed the ZEBRA dataset. Handwriting recognition – Reza Farrahi Moghaddam, Mathias Adankon, Kostyantyn Filonenko, Robert Wisnovsky, and Mohamed Chériet (Ecole de technologie supérieure de Montréal, Quebec) contributed the IBN SINA dataset. Marketing – Vincent Lemaire, Marc Boullé, Fabrice Clérot, Raphael Féraud, Aurélie Le Cam, and Pascal Gouzien (Orange, France) contributed the ORANGE dataset, previously used in the KDD cup 2009. We also reused data made publicly available on the Internet. We are very grateful to the researchers who made these resources available: Chemoinformatics – The National Cancer Institute (USA) for the HIVA dataset. Ecology – Jock A. Blackard, Denis J. Dean, and Charles W. Anderson (US Forest Service, USA) for the SYLVA dataset (Forest cover type). Text processing – Tom Mitchell (USA) and Ron Bekkerman (Israel) for the NOVA datset (derived from the Twenty Newsgroups dataset.

References

D. M. Allen. The relationship between variable selection and prediction. *Technometrics*, 16:125–127, 1974.

A. Borisov, V Eruhimov, and E. Tuv. Tree-based ensembles with dynamic soft feature selection. In I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh, editors, *Feature Extraction Foundations and Applications*, volume 207 of *Studies in Fuzziness and Soft Computing*. Springer, 2006.

B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Fifth Annual Workshop on Computational Learning Theory*, pages 144–152, Pittsburgh, 1992. ACM.

G. C. Cawley. Some baseline methods for the active learning challenge. *Journal of Machine Learning Research, Workshop and Conference Proceedings (*in preparation*)*, 10, 2010.

D. Cohn, Z. Ghahramani, and M.I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7:1–30, 2006. ISSN 1533-7928.

Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28:133–168, 1997.

I. Guyon, A. Saffari, G. Dror, and J. Buhmann. Performance prediction challenge. In *IEEE/INNS conference IJCNN 2006*, Vancouver, Canada, July 16-21 2006.

I. Guyon et al. Datasets of the active learning challenge. Technical Report, 2010. URL `http://clopinet.com/al/Datasets.pdf`.

D. Lewis and W. Gale. A sequential algorithm for training text classifiers. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12. ACM/Springer, 1994.

N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *International Conference on Machine Learning (ICML)*, pages 441–448. Morgan Kaufmann, 2001.

K. Saadi, G. C. Cawley, and N. L. C. Talbot. Optimally regularised kernel Fisher discriminant classification. 20(7):832–841, September 2007.

C. Salperwyck and V. Lemaire. Post-hoc experiments for the active learning challenge. Technical report, Orange Labs, 2010. URL `http://perso.rd.francetelecom.fr/lemaire/`.

B. Settles, M. Craven, and S. Ray. Multiple-instance active learning. In *Advances in Neural Information Processing Systems (NIPS)*, volume 20, pages 1289– 1296. MIT Press, 2008a.

B. Settles, M. Craven, and S. Ray. An analysis of active learning strategies for sequence labeling tasks. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, volume 20, pages 1069–1078. ACL Press, 2008b.

Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

H. S. Seung, M. Opper, and H. Sompolinskyy. Query by committee. In *ACM Workshop on Computational Learning Theory*, pages 287–294, 1992.

V. Sindhwani and S. S. Keerthi. *Newton Methods for Fast Solution of Semi-supervised Linear SVMs*. MIT Press, 2005.

V. Sindhwani and S. S. Keerthi. Large scale semi-supervised linear SVMs. In *SIGIR*, 2006.

S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2002.

Simon Tong and Daphne Koller. Active learning for parameter estimation in bayesian networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 647–653, 2000.

Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

Appendix A. Post-challenge verifications

The protocol of the challenge included several means of enforcing the rule that teams were not allowed to exchange information on the labels, including a manual verification of team membership and a method for detecting suspicious entries described in this appendix.

For one of the datasets (dataset A), we took advantage of the fact that the problem was a multi-class problem, with 15 classes. We assigned at random a different classification problem to each team. The teams were not informed that they were working on different problems.

To hide our scheme, we needed to provide the same seed example to all teams. To make this possible, we created 14 classifications problems for separating class1 or class i *vs.* all other classes, where i varies from 2 to 15. The seed was one example of class 1 (the same for all problems). We assigned the classification problems randomly to the teams. The intention was to detect eventual suspicious entries that could betray that the team tried to use label information acquired "illegally" from another team.

For each team, we computed the learning curves and the corresponding score (normalized ALC, Area under the Learning Curve) for all possible target values, hence 14 scores. We investigated whether the teams scored better on a problem they were not assigned to, hence for which they could not legitimately purchase labels. Our scheme obviously relies on our ability to determine the significance of score difference. Theoretical confidence intervals for our score (normalized ALC) are not known. There is naturally some variance in the results making it possible that by chance a team would get a better result on one of the problems to which it was not assigned. Furthermore, as illustrated in Figure 13, the problems are correlated, which increases the chances of getting a better score on another problem.

Twenty one teams turned in results on dataset A, including all the competition winners. Only a few teams submitted results in the final phase on other datasets than dataset A, so they could not be checked. However, they scored poorly in the challenge and were not in the top five for any dataset. Hence, our scheme allowed us to verify all the top ranking teams. We proceeded in the following way:

- The teams whose score on their assigned problem was better then their score on all other problems (11 among 21) were declared beyond suspicion. This includes the overall winner (Intel) and 3 other winning teams (Brainsignals, ROFU, and DATAM1N).

- Because the problems were assigned at random, some were not assigned to any team (target values 10 and 11). Among the remaining 10 teams, those having their best score on problems 10 or 11 (5 among 10 remaining teams) were declared beyond suspicion, since none of the teams could purchase label for these tasks. This includes one other winning team (Flyingsky).

- For the remaining 5 teams, we waived suspicion to 4 of them who had a score lower than 0.3 for *any* problem (their score was so close to random guessing that it was easy to score higher on another problem by chance). Most of those teams also scored low on all the other datasets. Only one of those teams scored high on some other datasets (NDSU), but did not win on any dataset.

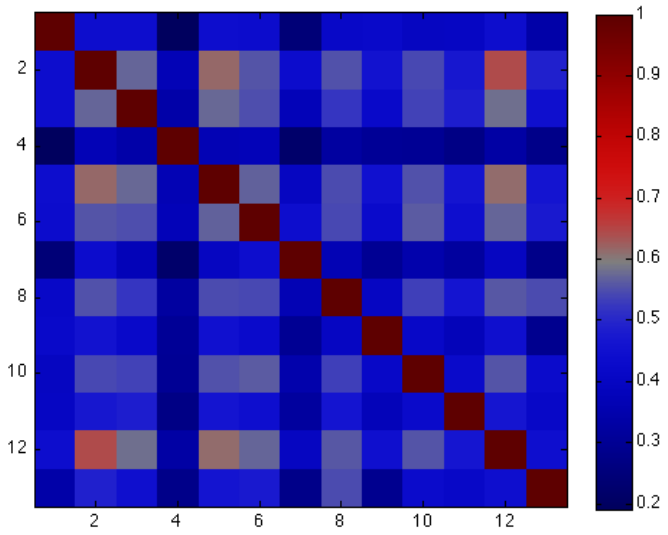


Figure 13: **Correlation matrix between target values for dataset A.** Targets 2 and 12 are particularly correlated.

- There remained only one team (IDE). Their results are better on problem 2 than on their assigned problem, which happens to be problem 12. These results are not suspicious either because those two problems are very correlated.
- For the least conclusive cases, we performed a visual examination of the learning curves to detect eventual suspicious progressions.

In conclusion, none of the verification results raised any suspicion. Admittedly, these results are not very strong because of the imperfections of the test due to noise and label correlation. However, combined with the other measures we took to enforce the rules of the challenge, this test gave us confidence in the probity of all the teams and did not justify further verifications by asking the teams to deliver their code.