



25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

# Integrating Iterative Machine Teaching and Active Learning into the Machine Learning Loop

Eduardo Mosqueira-Rey<sup>a,\*</sup>, David Alonso-Ríos<sup>a</sup>, Andrés Baamonde-Lozano<sup>a</sup>

<sup>a</sup>Universidad de Coruña, Centro de Investigación en TIC (CITIC), Elviña, 15071, A Coruña, Spain.

## Abstract

Scholars and practitioners are defining new types of interactions between humans and machine learning algorithms that we can group under the umbrella term of Human-in-the-Loop Machine Learning (HITL-ML). This paper is focused on implementing two approaches to this topic—Iterative Machine Teaching (iMT) and Active Learning (AL)—and analyzing how to integrate them in the learning loop. iMT is a variation of Machine Teaching in which a machine acts as a teacher that tries to transfer knowledge to a machine learning model. The focus of the problem in iMT is how to obtain the optimal training set given a machine learning algorithm and a target model. The idea is to learn a target concept with a minimal number of iterations with the smallest dataset. Active Learning, in contrast, is a specialized type of supervised learning in which humans are incorporated in the loop to act as oracles that analyze unlabeled data. AL allows us to achieve greater accuracy with less data and less training. Our proposal to incorporate iMT and AL into the machine learning loop is to use iMT as a technique to obtain the “Minimum Viable Data (MVD)” for training a learning model, that is, a dataset that allows us to increase speed and reduce complexity in the learning process by allowing to build early prototypes. Next, we will use AL to refine this first prototype by adding new data in an iterative and incremental way. We carried out several experiments to test the feasibility of our proposed approach. They show that the algorithms trained with the teachers converge faster than those that have been trained in a conventional way. Also, AL helps the model to avoid getting stuck and to keep improving after the first few iterations. The two approaches investigated in this paper can be considered complementary, as they correspond to different stages in the learning process.

© 2021 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0>)

Peer-review under responsibility of the scientific committee of KES International.

**Keywords:** Iterative Machine Teaching; Active Learning; Machine Learning; Human-in-the-Loop Machine Learning

\* Corresponding author. Tel.: +34-881-01-1343 ; fax: +34-981-167-160.

E-mail address: [eduardo@udc.es](mailto:eduardo@udc.es)

## 1. Introduction

There is currently great demand for Machine Learning (ML) solutions. This is because the advances that have occurred in recent years in this technology have popularized it and have brought it closer to the general public. But building machine learning systems is a complex process that requires deep knowledge of machine learning techniques.

Nowadays, humans are required at various points in the loop of the machine learning process but following a kind of monolithic conception in which the machine learning algorithm is modeled, built, tested, and then offered to the public without further changes.

Models that are developed in that way run the risk of not scaling well, becoming static, being hard to evaluate and degrading their performance due to changes in the context they are deployed into. Also, due to the limitations of the dominating connectionist approach, they usually lack logical reasoning and the possibility of identifying causal relations [13].

Researchers are defining new types of interactions between humans and machine learning algorithms, which we can group under the umbrella term of *Human-in-the-Loop Machine Learning* (HITL-ML). The idea is not only to make machine learning more accurate or to obtain the desired accuracy faster, but also to make humans more accurate and more efficient [19]. This interaction between humans and ML algorithms can be at several levels:

1. **Human experts act as teachers:** Humans can act not only as data-engineers or machine learning experts; their roles can be enhanced to behave as *teachers* of the machines [15].
2. **Users act as teachers:** In addition, the role of teachers can also be performed by non-expert users, who can also transfer their knowledge while interacting with products that include ML algorithms [29]. This way, those algorithms can, and should, adapt to changes that would cause the model to fail in the traditional monolithic paradigm.
3. **Algorithms explain their results to humans:** Explainable AI (XAI) [2] is a research field that aims to make the results of AI systems more understandable to humans. Currently, it has been noted that the humans' role has not been sufficiently studied in existing approaches to explainability [1].

Depending on who is in control of the learning process, we can identify different approaches to HITL-ML [13]:

1. **Active Learning (AL)** [23], in which the system remains in control of the learning process and treats humans as oracles to analyze unlabeled data.
2. **Interactive Machine Learning (IML)** [11][4], in which there is a closer interaction between users and learning systems, with people iteratively supplying information in a more focused, frequent and incremental way compared to traditional machine learning.
3. **Machine Teaching (MT)** [24], where human domain experts have control over the learning process by delimiting the knowledge that they intend to transfer to the machine learning model.

Shu et al. [27] classify these systems in a similar way according to who is in charge of picking the next data to label. Therefore, we have a *computer-initiated* approach where the learning algorithm picks the example, a *human-initiated* approach in which the human picks the example and a *mixed-initiative* approach, in which both computers and humans pick the next example to be labeled.

In this paper, we propose how to integrate some of these new techniques into the machine learning loop. To this effect, we will propose to use Machine Teaching (MT), specifically a variety of it known as Iterative Machine Teaching (iMT), to find the minimum data set that allows us to prototype models. Subsequently, we would use Active Learning (AL) to incorporate new emerging data into the learning process. In this way, we intend to make better use of the data available at a given time and favor the integration of new data into the model when necessary. We carried out several experiments in this regard to test the feasibility of this approach.

The paper is thus structured as follows: Section 2 presents a review of the MT techniques in general, and iMT techniques in particular. In section 3 we can see a review of the AL approach. Section 4 describes the experiments carried out including the details of our iMT and AL approach, the datasets used and the results obtained. We finish with the discussion, conclusions and future work in section 5.

## 2. Machine Teaching

In Machine Teaching (MT) we consider the data and the algorithms needed for ML as design materials, and our work consists in researching how to manage these “materials” into a final product [8]. Therefore, we can identify a new role in the machine learning approach, namely, the “*machine teacher*” who performs the “*machine teaching*” process for the “*learner*” (or student), that is, our machine learning algorithm.

Traditional machine learning is focused on creating new algorithms to increase accuracy when solving problems. MT, on the other hand, is focused on the effectiveness of the learning process, placing special emphasis on how the teacher filters the training set and gives the student the most relevant examples depending on the moment of learning in which the student is, something that is closely related to *curriculum learning* [6].

In the literature, we can find two ways of using MT that lead to two different objectives to be achieved. On the one hand, there is the MT as proposed by Simard [24], in which humans retain control over the learning process and there is a strong emphasis on the teacher’s interaction with data paying special attention to design principles of interaction and visualization. This is a human-centered approach to building ML models and leads us to what is known as Interactive Machine Teaching (IMT), using the ideas of Interactive Machine Learning (IML) within MT [21].

On the other hand, some authors have posed the question of “*could a machine, and not a human, take the role of a teacher in a Machine Teaching process?*” And the answer is yes, it is possible, but then there is a shift in the focus of the problem we are trying to solve. In this case, since the teacher is a machine who already knows the target model, we focus on how to obtain the optimal training set given a machine learning algorithm and a target model [30].

This particular version of Machine Teaching is called “Iterative Machine Teaching” (iMT) [16]. This approach—which should not be confused with IML or IMT—focuses on learning a target concept with a minimal number of iterations with the smallest dataset. It is called *iterative* because the teacher iteratively and intelligently feeds new examples to the learner based on the observable performance of the latter. It is not *interactive* since the teacher here is another machine, and there are no humans involved in the process.

What applications could iMT have, if we already have a machine that knows the model—why not use it directly, why use it to train another one? In this regard we have several proposals. Zhu [30] proposes changing roles and making the student a human being and not a machine learning model. The teacher transfers knowledge by optimizing the choice of examples shown to these students. An interesting application of this technique is when applied to crowdsourcing services such as Amazon’s Mechanical Turk Platform. Singla [25] explores how to teach workers in crowdsourcing services in order to improve their accuracy.

Zhu [30] also warns of malicious uses of this technique, this is called training-set attacks [18] or training-set poisoning [31], and it consists in attackers who contaminate the training data so that a specific learning algorithm would produce a model that is beneficial to them, e.g. manipulating spam filters to make malicious emails pass.

Other authors focus iMT on constructing the smallest dataset needed to learn a model. These efforts can be rooted to the Goldman and Kearns’s concept of *teaching dimension* [12], that is, the cardinality of the optimal teaching set. In this regard, Zhu [31] proposes a general framework and organizes iMT as a coherent set of ideas and Liu [17] proposes an active teacher model that queries the learner (i.e., takes exams) for estimating their status and for guiding it in order to achieve faster convergence.

In this work, we want to propose two other different applications for iMT. The first one is using iMT as a technique to obtain the “*Minimum Viable Data (MVD)*” for training a learning model. MVD is a term coined by van Allen [3] that refers to the minimum data needed to train the machine learning models. The name is borrowed from the agile world, in which we have the idea of a “*Minimum Viable Product (MVP)*”, a product with just enough features to satisfy early customers, and to provide feedback for future product development.

In this case, we propose the use of iMT not for finding the optimal minimum dataset, but a dataset that allows us to increase speed and reduce complexity in the learning process by allowing to build early prototypes and to grasp new insights about the characteristics of the data needed. Further on, researchers can add additional data to be integrated into the system. iMT can also be used as a data maintaining strategy, as new data become available, we can reduce the size of the data by keeping those examples that are most representative and useful for the learning process.

The second application is to focus on using iMT as a way to transfer the knowledge from a complex model—e.g., a deep neural network, which has usually better performance but suffers from transparency and understandability issues—to a more easily understood model with better explainability capabilities—such as a decision tree.

### 3. Active Learning

Active learning (AL) is a machine learning approach in which the learner asks an oracle (who acts as a teacher), to label examples that are not clear or that will provide relevant information in the learning process. As a result, the learner improves its learning performance.

Active learning uses an *interactive/iterative process* for obtaining training data, unlike passive or classical learning, where the data is provided in advance. It is said that the learner is curious and requests information from the oracle that it selects based on different query strategies [23].

AL was inspired by the family of instructional techniques by the same name in the education literature [7] whose intention is to make the student a partner in the learning process and thus not being overly dependent on the teacher.

There are several approaches to selection in Active Learning [19]. For example:

- **Random Sampling.** It selects randomly from unlabeled instances. It may not be representative enough because samples might not be equitably distributed, but it often performs better than other simple, naive approaches [10].
- **Uncertainty Sampling.** It identifies unlabeled elements that are close to the decision boundary and whose class membership is unclear [14]. These elements are most likely to be wrongly classified. The model is then trained again, with the aim of improving it by including new elements of the boundary regions.
- **Diversity Sampling.** It includes new elements in the model that are not necessarily close to the decision boundary. The goal is to detect elements that are novel and unknown to the model, which may lead to a more complete picture of the problem space and to improving the model's accuracy.

Active learning offers several advantages: (1) the system could achieve greater accuracy using fewer labeled instances, (2) being allowed to choose between the available unlabeled examples, the learner will gain accuracy faster, with less training and (3) incorporating humans into the learning process improves the process of adding and labeling new data and makes the ML model more accurate.

AL is of special interest when the labeling example process is expensive or time-consuming, and it also applies to the scenario of scarcity of examples (e.g.: rare diseases). The active learner aims to achieve high accuracy using as few labeled instances as possible, thereby minimizing the cost of obtaining labeled data [23].

Our proposal for using AL within the ML loop is to continue the steps initiated after the application of iMT. At this point we have a MVD dataset from from which we can create the first prototype of the model. We will then use AL to refine this first prototype by adding new data.

This refinement will follow a mix of exploitation and exploration techniques. Exploitation will be carried out with uncertainty sampling, trying to obtain more information on doubtful cases, which are located on the border of the decision boundary, whereas the exploration will be carried out following diversity sampling, trying to incorporate new data to the model looking for new possibilities that had not been taken into account.

People tend to focus on exploitation trying to improve the accuracy on the most difficult cases, but exploration is also relevant if we want the model to generalize (and we want it). We think that an approach that combines the two is the best approach. On this regard Munro [19] proposed a simple but effective strategy that consists in applying one technique at the output of the other. Roy and McCallum [22] proposed the *Expected Error Reduction* method with the aim of combining the two techniques into a single metric, but it is a computationally expensive method that does not scale well to large-scale datasets. Baram [5] utilizes an ensemble of active-learning algorithms and tracks online the best algorithm in the ensemble. Lastly, one interesting approach is the one proposed by Osugi et al. [20], they proposed a strategy that randomly chooses between exploitation and exploration at each round, then the system receives feedback on how effective the exploration phase was and, following a simple reinforcement learning algorithm, the active learner updates the probability of exploring in subsequent rounds.

Finally, it is important to remark that, as mentioned above, AL is an iterative but also an interactive process. When incorporating humans, it is necessary to take into account aspects related to Human-Computer Interaction (HCI). We have to consider not only the development of machine learning models but also the design of the interactions and behaviors that compose the human experience around the models [3]. Here HCI plays a key role in achieving intelligent systems that continuously improve through use [15].

## 4. Experiments

Following the philosophy outlined in the previous sections, we considered the need to perform several experiments in the field of iMT and AL to test the feasibility of the proposed approach. In the following sections we will explain these experiments, describe the datasets used in them, and comment on the results obtained.

### 4.1. Iterative Machine Teaching Experiment

The first experiment we carried out in this research was related to Iterative Machine Teaching (iMT). We propose and implement different types of machine teachers and use them firstly with example problems—in order to test their functioning—and finally with a real problem, namely, identifying and classifying vehicles by weight and size.

With iMT, what we do is create a smaller dataset in which most of the examples are of border regions between classes, eliminating those redundant instances that belong to categories that do not provide greater precision to the model. This results in using less data, performing fewer learning iterations and making the models more accurate.

The teacher's objective is to provide examples to the student so that the model converges as quickly as possible. The selection of these examples will depend on their *difficulty* and *usefulness*. To wit:

- **Difficulty of an example.** The difficulty is the amount of information that the example brings to the learning process. The more information the example has, the more difficult it is to learn.
- **Usefulness of an example.** Usefulness is a measure that correlates the difficulty of an example with the discrepancy in how the teacher and the student interpret that example. If the discrepancy is very large, it means that the example is very useful in the current iteration.

In the case of Iterative Machine Teaching, we can distinguish between different types of teachers, depending on the information the teacher has about the students [16]:

- **Omniscient teacher.** It has complete access to the student's characteristics: feature space, model, loss function and optimization algorithm. With this teacher, the difficulty of an example is calculated using the loss function of the student (as the norm of the gradient of the squared loss function) and the usefulness is calculated using the discrepancy between the student's weights and the teacher's weights while taking into account the difficulty of that example.
- **Surrogate teacher.** It only has access to the loss function. The difficulty of an example is calculated in the same way as the omniscient teacher, but the usefulness of an example must be calculated differently since only the loss function is available. In this case, the usefulness is calculated as the loss of the student minus the loss of the teacher (taking again into account the difficulty of the example).
- **Imitation teacher.** It does not have access to the student's learning performance. Therefore, the teacher needs to have a copy of the student that serves as a reference for the selection of examples. For this copy, the teacher would access the learning parameters as if it were an omniscient teacher.

The teaching algorithm aims to minimize the difficulty of the examples and maximize their usefulness. For each iteration, the teacher is tasked with choosing a particular example. In the early iterations, the teacher will look for instances that are not excessively complex since they would be counterproductive to advancing the learning. When the training is at a more advanced stage, the teacher will prefer more difficult examples. Here our objective is different, since at this point in learning we are already very close to the decision boundary and the simplest examples would not be relevant here.

The algorithm for comparing a passive training with an Iterative Machine Teaching training is detailed below (algorithm 1). As learners we use either a **linear classifier**—that bases its decisions on a linear combination of the characteristics of the objects—or a **convolutional neural network** (ConvNet/CNN)—a Deep Learning algorithm in which the neurons correspond to receptive fields in a similar manner to the neurons in the visual cortex of the brain.

**Algorithm 1** Comparison algorithm between passive training and the iML approaches (comments to the right)

---

```

procedure COMPARISON(numIter, dataset, algorithm, teacher)
  dataset ← preprocess(dataset)
  randomizedDataset ← splitAndRandomize(dataset)
  algorithm ← initAlgorithm(algorithm)
  for all example ∈ randomizedDataset do
    noTeachingResults ← train(algorithm, example)
  end for
  teacher ← initTeacher(algorithm)
  for all example ∈ randomizedDataset do
    train(teacher, example)
  end for
  student ← initStudent(algorithm)
  for all iteration ∈ numIter do
    example ← teacher.select(student, dataset)
    teachingResults ← student.train(example)
  end for
  compare(teachingResults, noTeachingResults)
end procedure

```

---

- Data are preprocessed for the learning algorithm.
- Data are split (train & test) and randomized.
- The algorithm (learner) is initialized.
- Passive learning loop.
- The teacher is initialized with the algorithm.
- The teacher is trained if necessary.
- The algorithm (learner) is initialized as a student.
- The iMT algorithm takes place.
- The teacher selects the following example.
- The algorithm is trained with that example.
- The loop continues until a halt criterion is met.
- Comparison of results.

#### 4.2. Active Learning Experiment

The second experiment performed in this work has to do with Active Learning and deals with how to incorporate new data once the first prototype of the model has been built. This data will probably come unlabeled, and we will need help in labeling it. It is at this point that the human is incorporated into the loop.

The three aspects that we consider fundamental for a successful AL experiment to be conducted are:

- **Determine the AL strategy:** Decide if you want to perform an exploitation—performing an uncertainty sampling and choosing those elements that are close to the decision boundary—or an exploration—including novel and unknown elements to better cover the problem space.
- **Make the model updateable:** The model is constantly trained with the newly labeled elements and its efficacy is compared against the previous model. The system keeps a history of the accuracy of each configuration of the algorithm.
- **Create an usable interface for manual annotation:** Since we include humans in the learning loop, it is necessary to take human factors into account when developing the system. HCI techniques must be used to avoid user boredom and frustration [4].

In the AL experiment we pose a simple problem, namely, the classification of images into categories, and we address the three aspects previously mentioned in the following way: First, the AL strategy chosen was the uncertainty sampling, we want to follow a *curriculum learning* strategy [6] focusing our efforts in the *know unknowns*, that is, data our model cannot confidently deal with. In particular, we use entropy-based sampling calculating the uncertainty as the difference between all predictions, as defined by information theory [19].

Second, to make the system more scalable and updateable and to allow for easy integration of new data once in production, we propose a distributed architecture in which the model resides on a central server but is propagated to simpler nodes. These nodes would be in charge of forwarding new examples or conflicting data that they do not know how to deal with to be annotated by humans.

Finally, to filter the data that reaches humans (and try not to deliver poor quality or unrepresentative data that would make them lose interest in the system), *Haar Cascades*—a machine learning approach for visual object detection which is capable of processing images extremely rapidly and achieving high detection rates [28]—are used. The final architecture of the system can be seen in Figure 1.

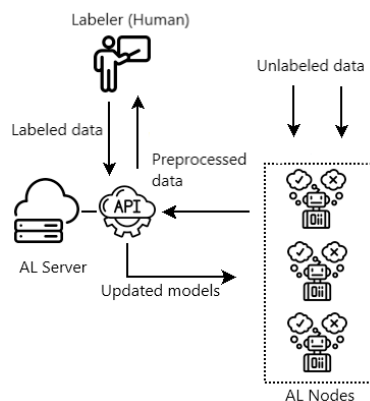


Fig. 1. Architecture of the AL approach.

#### 4.3. Datasets

The iMT experiment uses three datasets. For the sake of simplicity, all the datasets contain exactly two classes (i.e., we are not performing multi-class classifications).

Firstly, the **Gaussian** dataset contains data from a Gaussian function generated on the fly with the NumPy library (more specifically, with its ‘numpy.random.multivariate\_normal’ function). These data will be classified with a linear classifier. The input data corresponds to Gaussian functions generated with means of 0.5 and  $-0.5$ , respectively. The output is a binary classification of the generated Gaussian function, either belonging to the class 0.5 or  $-0.5$ .

Secondly, the **MNIST** dataset consists of handwritten digits, and is commonly used for training image processing systems. In this dataset, each digit belongs to a class. In our case, we selected different pairs of classes to be compared in each experiment. The inputs are the images of the handwritten digits, and the output is a binary classification as one of the two digits in the pair.

The third dataset is the **Vehicles** dataset [26], which consists of images taken by surveillance cameras from different angles. The images have been split into two classes, namely, light-duty vehicles and heavy-duty vehicles. Light vehicles include cars of different sizes, while heavy vehicles include vans and trucks. The inputs are the original images of the dataset after normalizing them to 32x32px grayscale pictures. The output is a classification of the images as either a light-duty vehicle or a heavy-duty one.

The AL experiment uses a different dataset, namely, the **Stanford Dogs** dataset. It consists of over 20,000 images of dogs from all over the world, corresponding to 120 different breeds.

#### 4.4. Results

The first two results correspond to the iML experiment with the Gaussian and MNIST datasets using the linear classifier. The results are shown in Figures 2 and 3. As can be seen, the algorithms trained with the teachers converge faster than the ones trained in a conventional way (passive learning).

The next result was again an iMT experiment using the CNN with the vehicles dataset. The parameters of the CNN were intentionally not optimized, as the purpose of the experiment was to check whether our model converged faster than another model that was not taught by any of the proposed teachers. The results (Figure 4) show that the algorithms trained by the proposed machine-teaching teachers converge faster than the ones that have not been trained this way. Particularly the one trained by the surrogate teacher.

The final result consisted in using AL to incorporate humans into the learning cycle. We used the Stanford dogs dataset, starting with a very small dataset and, on each iteration, adding ten unlabeled elements for which the model is not conclusive. The results are shown in Figure 5, in the first graph we ran our model trying to discriminate between two dog breeds: Chihuahua and Japanese Spaniel. We obtained high accuracy in the first few iterations because the dataset was very small. When we begin to add new examples, the accuracy of the system drops at first but soon recovers and keeps on improving thereafter. The second graph corresponds to training our model with another pair

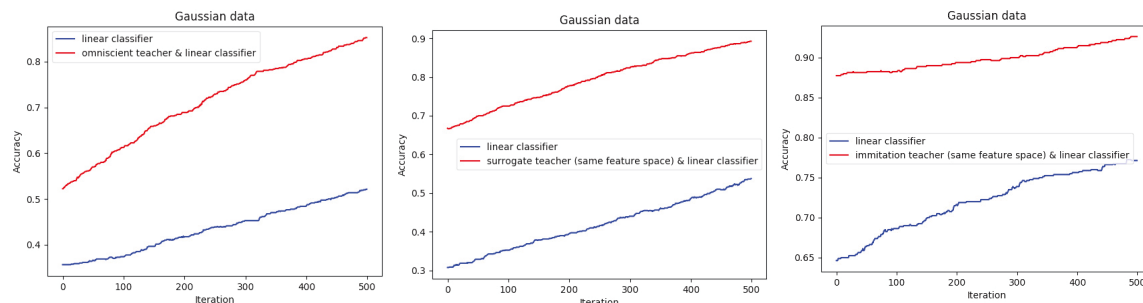


Fig. 2. Gaussian dataset results for a linear model with the three different teachers (red) and without teachers (blue).

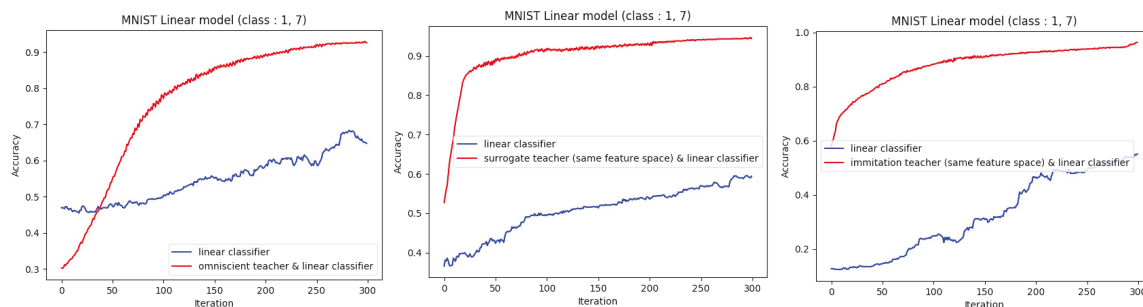


Fig. 3. MNIST dataset results for a linear model with the three different teachers (red) and without teachers (blue).

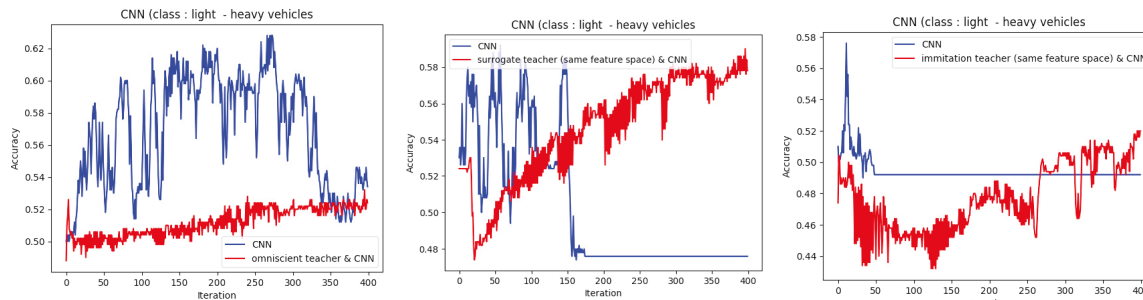


Fig. 4. Vehicles dataset results for a linear model with the three different teachers (red) and without teachers (blue).

of dog breeds: Bloodhound and Papillon. We obtained similar results, but without the initial drop in accuracy. A traditional model would be stuck in the first iteration, as accuracy would not improve with new elements. In the last graph, we compared our preferred way of uncertainty sampling, the entropy sampling, with a random sampling technique. We found that accuracy grew faster with entropy sampling.

### 5. Discussion, conclusions and future work

In this paper we have used two distinct approaches to enhancing Machine Learning, namely, Iterative Machine Teaching (iMT) and Active Learning (AL). First of all, we should mention that the two approaches are not mutually exclusive, as they correspond to different stages of the automatic learning process. iMT reduces the size of the dataset, while the inclusion of actual humans in the loop leads to capturing new data and updating the model in an iterative and incremental way. Both approaches help us to make the process more efficient by different means.

One important concept that is worth highlighting and researching is the *teaching dimension* [12]. When we train with iMT, what we do is create a smaller dataset where most of the examples belong to borderline regions, with the aim of differentiating between classes. That is, when an example is evaluated to train the algorithm, its utility is



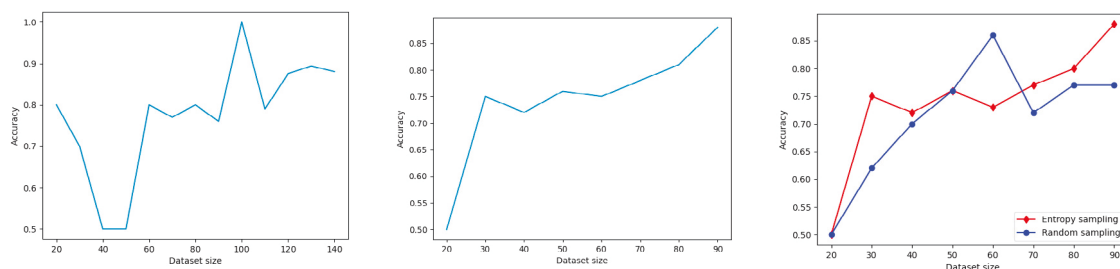


Fig. 5. Stanford dogs dataset results. Two experiments with different dog breeds and comparison between entropy sampling and random sampling.

checked. If this instance does not confer greater precision to the model, it will not be used for training. Therefore, we would generate a training set with very few redundant data and we can see that iMT can be useful as a data selection tool, since it helps to select the most representative examples of our dataset.

In addition, if our data is better our algorithm will generalize better. The results would improve, both in terms of training time (it would be faster) and precision (as shown in the previous examples). The models converge faster because the teachers select the most appropriate example for each moment of the learning process, which gives us a significant advantage over an algorithm that chooses examples randomly. Even if the learner is a black box and the teacher does not interact with it, it is demonstrated that the results are better than providing random examples [9].

In our iMT experiment, the results show—both in the example problems and in the real-world problem—that the algorithms trained by any of the proposed teachers obtain better results than those trained by randomly choosing the examples. In the example problems, accuracy is significantly better when using iMT than when not using it. In the real-world problem the difference is not as significant. This is due to the fact that the problem is more complex (the examples have noise, have not gone through an exhaustive preprocessing phase, etc.) and optimal parameters were not sought to improve the accuracy of the algorithm because it was not the objective of our experiment, which was more focused on the selection of examples and the improvement of convergence with a low number of examples. As a drawback of the iMT process we can say that training times are noticeable longer than the ones for conventional machine learning. This is because, on each machine teaching iteration, the dataset must be reevaluated to calculate the utility of the examples. This utility is then used by the teacher to select the next instance for the learner.

In our AL experiment, we find that the greatest advantage of this approach is in the continuous improvement of the model, which enhances resilience and prevents obsolescence. Human involvement is not free of cost, however. The human must make decisions, and the act of classifying can be a complex task in some cases. We must also take scalability into account—on the one hand, this approach allows the model to be scaled to a high number of nodes; on the other hand, too much conflicting data can lead to excessive demands on human experts and we have to take into account human factors and HCI issues.

As for future work, we would be interested in considering multi-class problems and applying iMT and AL jointly to new real-world problems. Another important thing to consider is the time consumed by the model, particularly in terms of training. This could be optimized by improving the heuristics for the selection of examples, or by discarding certain parts of the dataset in the earliest stages, as the complete reevaluation of the model can be costly.

## Acknowledgements

This work has been supported by the State Research Agency of the Spanish Government (grant PID2019-107194GB-I00 / AEI / 10.13039/501100011033) and by the Xunta de Galicia (grant ED431C 2018/34) with the European Union ERDF funds. We wish to acknowledge the support received from the Centro de Investigación de Galicia “CITIC”, funded by Xunta de Galicia and the European Union (European Regional Development Fund- Galicia 2014-2020 Program, grant ED431G 2019/01).

## References

- [1] Abdul, A., Vermeulen, J., Wang, D., Lim, B.Y., Kankanhalli, M., 2018. Trends and trajectories for explainable, accountable and intelligible systems: An HCI research agenda, in: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, Association for Computing Machinery, New York, NY, USA. pp. 1–18. doi:[10.1145/3173574.3174156](https://doi.org/10.1145/3173574.3174156).
- [2] Adadi, A., Berrada, M., 2018. Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access* 6, 52138–52160. doi:[10.1109/ACCESS.2018.2870052](https://doi.org/10.1109/ACCESS.2018.2870052).
- [3] van Allen, P., 2018. Prototyping ways of prototyping AI. *Interactions* 25, 46–51. doi:[10.1145/3274566](https://doi.org/10.1145/3274566).
- [4] Amershi, S., Cakmak, M., Knox, W.B., Kulesza, T., 2014. Power to the people: The role of humans in interactive machine learning. *AI Magazine* 35, 105–120. doi:[10.1609/aimag.v35i4.2513](https://doi.org/10.1609/aimag.v35i4.2513).
- [5] Baram, Y., Yaniv, R.E., Luz, K., 2004. Online choice of active learning algorithms. *Journal of Machine Learning Research* 5, 255–291.
- [6] Bengio, Y., Louradour, J., Collobert, R., Weston, J., 2009. Curriculum learning, in: Proceedings of the 26th Annual International Conference on Machine Learning, Association for Computing Machinery, New York, NY, USA. pp. 41–48. doi:[10.1145/1553374.1553380](https://doi.org/10.1145/1553374.1553380).
- [7] Bonwell, C.C., Eison, J.A., 1991. Active Learning: Creating Excitement in the Classroom. ASHE-ERIC Higher Education Reports.
- [8] Churchill, E.F., van Allen, P., Kuniavsky, M., 2018. Designing AI. *Interactions* 25, 34–37. doi:[10.1145/3281764](https://doi.org/10.1145/3281764).
- [9] Dasgupta, S., Hsu, D., Poulis, S., Zhu, X., 2019. Teaching a black-box learner, in: Chaudhuri, K., Salakhutdinov, R. (Eds.), Proceedings of the 36th International Conference on Machine Learning, PMLR, Long Beach, California, USA. pp. 1547–1555. URL: <http://proceedings.mlr.press/v97/dasgupta19a.html>.
- [10] Donmez, P., Carbonell, J.G., 2010. From active to proactive learning methods, in: *Advances in Machine Learning I*. Springer, pp. 97–120.
- [11] Fails, J.A., Olsen Jr, D.R., 2003. Interactive machine learning, in: Proceedings of the 8th international conference on Intelligent user interfaces, pp. 39–45.
- [12] Goldman, S.A., Kearns, M.J., 1995. On the complexity of teaching. *Journal of Computer and System Sciences* 50, 20–31.
- [13] Holmberg, L., Davidsson, P., Linde, P., 2020. A feature space focus in machine teaching, in: 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), pp. 1–2. doi:[10.1109/PerComWorkshops48775.2020.9156175](https://doi.org/10.1109/PerComWorkshops48775.2020.9156175).
- [14] Lewis, D.D., Gale, W.A., 1994. A sequential algorithm for training text classifiers, in: SIGIR'94, Springer. pp. 3–12.
- [15] Lindvall, M., Molin, J., Löwgren, J., 2018. From machine learning to machine teaching: The importance of UX. *Interactions* 25, 52–57. URL: <http://doi.acm.org/10.1145/3282860>, doi:[10.1145/3282860](https://doi.org/10.1145/3282860).
- [16] Liu, W., Dai, B., Humayun, A., Tay, C., Yu, C., Smith, L.B., Rehg, J.M., Song, L., 2017. Iterative machine teaching, in: Proceedings of the 34th International Conference on Machine Learning (ICML-2017) - Volume 70, pp. 2149–2158. [arXiv:1705.10470](https://arxiv.org/abs/1705.10470).
- [17] Liu, W., Dai, B., Li, X., Liu, Z., Rehg, J., Song, L., 2018. Towards black-box iterative machine teaching, in: Dy, J., Krause, A. (Eds.), Proc. of the 35th Int. Conf. on Machine Learning, PMLR. pp. 3141–3149. URL: <http://proceedings.mlr.press/v80/liu18b.html>.
- [18] Mei, S., Zhu, X., 2015. Using machine teaching to identify optimal training-set attacks on machine learners, in: Proc. of the 29th AAAI Conf. on Artificial Intelligence, pp. 2871–2877. URL: <https://ojs.aaai.org/index.php/AAAI/article/view/9569>.
- [19] Munro, R., 2020. Human-in-the-Loop Machine Learning. Manning Publications.
- [20] Osugi, T., Kim, D., Scott, S., 2005. Balancing exploration and exploitation: A new algorithm for active machine learning, in: Fifth IEEE International Conference on Data Mining (ICDM'05), IEEE. doi:[10.1109/ICDM.2005.33](https://doi.org/10.1109/ICDM.2005.33).
- [21] Ramos, G., Meek, C., Simard, P., Suh, J., Ghorashi, S., 2020. Interactive machine teaching: a human-centered approach to building machine-learned models. *Human-Computer Interaction* 35, 413–451. doi:[10.1080/07370024.2020.1734931](https://doi.org/10.1080/07370024.2020.1734931).
- [22] Roy, N., McCallum, A., 2001. Toward optimal active learning through Monte Carlo estimation of error reduction, in: Proceedings of the International Conference on Machine Learning (ICML), Morgan Kaufmann. pp. 441–448.
- [23] Settles, B., 2009. Active learning literature survey. Technical Report. University of Wisconsin-Madison Department of Computer Sciences. URL: <https://minds.wisconsin.edu/handle/1793/60660>.
- [24] Simard, P.Y., Amershi, S., Chickering, D.M., Pelton, A.E., Ghorashi, S., Meek, C., Ramos, G., Suh, J., Verwey, J., Wang, M., Wernsing, J., 2017. Machine teaching: A new paradigm for building machine learning systems. *arXiv preprint arXiv:1707.06742* URL: <http://arxiv.org/abs/1707.06742>.
- [25] Singla, A., Bogunovic, I., Bartók, G., Karbasi, A., Krause, A., 2014. Near-optimally teaching the crowd to classify, in: International Conference on Machine Learning, PMLR. pp. 154–162.
- [26] Sochor, J., Špaňhel, J., Herout, A., 2018. Boxcars: Improving fine-grained recognition of vehicles using 3-D bounding boxes in traffic surveillance. *IEEE Transactions on Intelligent Transportation Systems* 20, 97–108. doi:[10.1109/TITS.2018.2799228](https://doi.org/10.1109/TITS.2018.2799228).
- [27] Suh, J., Zhu, X., Amershi, S., 2016. The label complexity of mixed-initiative classifier training, in: Balcan, M.F., Weinberger, K.Q. (Eds.), Proceedings of The 33rd International Conference on Machine Learning, PMLR, New York, New York, USA. pp. 2800–2809. URL: <http://proceedings.mlr.press/v48/suh16.html>.
- [28] Viola, P., Jones, M., 2001. Rapid object detection using a boosted cascade of simple features, in: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, IEEE. pp. I-511–I-518. doi:[10.1109/CVPR.2001.990517](https://doi.org/10.1109/CVPR.2001.990517).
- [29] Yang, Q., Suh, J., Chen, N.C., Ramos, G., 2018. Grounding interactive machine learning tool design in how non-experts actually build models, in: Proceedings of the 2018 Designing Interactive Systems Conference, Association for Computing Machinery, New York, NY, USA. pp. 573–584. doi:[10.1145/3196709.3196729](https://doi.org/10.1145/3196709.3196729).
- [30] Zhu, X., 2015. Machine teaching: An inverse problem to machine learning and an approach toward optimal education, in: Proceedings of the 29th AAAI Conference on Artificial Intelligence, pp. 4083–4087. URL: <http://dl.acm.org/citation.cfm?id=2888116.2888288>.
- [31] Zhu, X., Singla, A., Zilles, S., Rafferty, A.N., 2018. An overview of machine teaching. *arXiv preprint arXiv:1801.05927* URL: <http://arxiv.org/abs/1801.05927>.