Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 225 (2023) 3086–3093
Procedia Computer Science
www.elsevier.com/locate/procedia
27th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2023)

Gamifying Machine Teaching: Human-in-the-Loop Approach for Diphthong and Hiatus Identification in Spanish Language

Eduardo Mosqueira-Rey[a,*], Sergio Fernández-Castaño[a], David Alonso-Ríos[a], Esteban Vázquez-Cano[b], Eloy López-Meneses[c]

[a] *Universidade da Coruña (CITIC), Campus de Elviña, 15071 A Coruña, Spain*
[b] *Universidad Nacional de Educación a Distancia, Madrid, Spain*
[c] *Universidad Pablo de Olavide, Sevilla, Spain*

Abstract

Human-in-the-Loop Machine Learning (HITL-ML) is a set of techniques that attempt to actively involve experts into the learning loop of machine learning (ML) models. One of these techniques is Machine Teaching (MT) which tries to apply techniques that come from the world of didactics within machine learning (ML), such as sorting the dataset according to its difficulty and presenting the cases to the model in incremental levels of complexity. In this work we propose a new twist to MT: since its foundation is to bring didactic techniques to ML, why not use this technique as a didactic method itself? In this case we propose the creation of an ML model for the identification of diphthongs and hiatuses in the Spanish language. The first step is to develop a deep learning model to identify diphthongs and hiatuses using Curriculum Learning (CL) and a sorted dataset that identifies simple and complex cases. The accuracy of this model identifies the upper limit of efficiency that we can obtain by training the model. The next step is to reset the weights of the model but retain its architecture and offer the model to the students for its training. The idea is that students use MT techniques to make the model learn again, but the ultimate goal is that students learn by teaching in an informal and gamified learning environment. The results show how a HITL strategy can make a model learn iteratively to identify diphthongs and hiatuses and a workflow is proposed to include this technique in the classroom.

Keywords: Orthography; Human-in-the-Loop Machine Learning; Machine Teaching; Curriculum Learning

* Corresponding author. Tel.: +34-881011343; fax: +34-981167160.
E-mail address: eduardo@udc.es

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 the 27th International Conference on Knowledge Based and Intelligent Information and Engineering Systems
10.1016/j.procs.2023.10.302

1. Introduction

1.1. Human-in-the-loop Machine Learning

Humans are needed at various points in the loop of the Machine Learning (ML) process, but the classical approach to this human-computer collaboration has a static structure: humans label and preprocess the data, decide the model to use, adjust its hyperparameters and the learning process is launched without any human intervention other than repeating the same process in a loop if the results are not the desired ones.

Conversely, Human-in-the-loop Machine Learning (HITL-ML) [8] has been proposed as a more interactive way to introduce humans into the ML model training process. The reasons for initiating a HITL-ML process are several, among which we can cite: to allow scaling models in the future with the appearance of new data, to introduce more easily expert knowledge into the model to improve its accuracy [11], to make the learning processes faster and less resource-consuming [13], to improve the explanatory capabilities of the models [4], to face data bottleneck problems [9], etc.

1.2. Machine Teaching

Depending on who is in control of the learning process, we can identify different approaches to HITL-ML [6][8]:

- **Active Learning (AL)** [12], in which the system remains in control treats humans as oracles to annotate unlabeled data.
- **Interactive Machine Learning (IML)** [1], in which there is a closer interaction between users and models, with people interactively supplying information in a focused, frequent, and incremental way.
- **Machine Teaching (MT)** [11], where human domain experts have control and delimit the knowledge that they intend to transfer to the ML model.

MT techniques can be separated in two groups, machines teaching machines [14][7] and humans teaching machines [11]. The main usefulness of the latter is to take advantage of the inherent abilities of humans for teaching, in order to allow people without a ML background to transfer knowledge to a computer system similarly to how they would teach another human. This would make machine learning methods accessible to subject-matter experts and would allow the creation of semantic and debuggable ML models.

1.3. Curriculum Learning

Curriculum Learning (CL) [2], is a technique that imposes some structure on the training set (sorting the dataset by the degrees of difficulty of the different cases) to accelerate and improve the learning.

There are two concepts to take into account in a CL process [5]:

- **Scoring function**: determines the difficulty of the examples and allow to classify the examples as easy or hard.
- **Pacing function**: determines how the data is presented to the network.

We can see that CL and MT are related, both has been inspired by didactics and by the way human beings learn. The ML model is the student that wants to learn, and CL and MT are the teachers organizing their curriculum in increasing levels of complexity to facilitate student learning.

The difference between CL and MT is that CL is not interactive, the curriculum is prepared prior the learning process, (automatically or by domain experts) and then the learning is started controlled by a pacing function. On the other hand, MT uses also an ordered curriculum, but in an interactive way in which the *teacher* (normally a human) responds to the results of the *student* (the model) by trying to reinforce the identified weak points of the student's learning.

1.4. Contribution of the paper

Since HITL techniques come primarily from the didactics we propose in this work a new twist to MT: **to use HITL techniques as a didactic method itself**. We propose to use MT as a "**learning by teaching**" approach [3]. Learning by teaching is a method in which students are the ones who teach and transmit knowledge to other people. This type of knowledge benefits very positively their learning and communication skills, since students learn concepts unconsciously, without realizing that they are facing a new lesson or receiving knowledge.

We will use CL to train the model previously to the MT experiment, with two ideas in mind: First, to decide the best architecture for the ML model and to prove that the model can learn following a curriculum strategy. Second, to provide the students a semi-trained model to avoid *cold starts* in the MT experiment with the learning progressing too slowly, which can lead to boredom and disinterest on the part of the student.

The paper is structured as follows: we describe the problem that we have chosen to test our idea in section 2, the ML model built for that purpose and the learning process followed using CL is described in section 3. The machine teaching experiment carried out is shown in section 4 and, finally, we end up with the conclusions and future work in section 5.

2. Problem description and dataset

In phonology, hiatus describes the occurrence of two separate vowel sounds in adjacent syllables with no intervening consonant. On the other hand, when two vowel sounds occur together as part of a single syllable, the result is called a diphthong.

Diphthongs and hiatuses are key elements of Spanish orthography since their understanding is necessary to separate a word into syllables correctly and, consequently, to be able to correctly apply accentuation rules.

Therefore, we propose the creation of an ML model for the identification of diphthongs and hiatuses in the Spanish language. This ML model will be, initially, fully-trained to identify the upper limit of efficiency that we can obtain by training the model. Because languages have multiple exceptions and inconsistencies one cannot expect that an ML model can generalize all the rules of a language that allow to distinguish diphthongs from hiatus.

2.1. Dataset description

We use a dataset with a total of 700 cases divided into three classes: diphthongs, hiatuses and none. For the first two classes, every word has been tagged with its estimated difficulty to learn (low, medium and high). Also included in the dataset is the type of diphthong/hiatus based on Spanish orthographic rules. In the case of the third class, the one with a set of words that do not contain any diphthong or hiatus, the estimated difficulty has been set to the lowest one, as these words do not present any further complexity that might confuse our model.

One problem with the datasets is that, since there are only 14 different diphthongs in Spanish, all other combination of vowels are hiatuses. For that reason the datasets are unevenly distributed, with more hiatuses (300) and other words (300) than diphthongs (100).

Furthermore, the amount of words for each difficulty level is also quite far apart, with around a 50% of words estimated to be of intermediate difficulty, that tried to be representative of what is actually happening in general with the Spanish language, but which will obviously have an impact on the performance of the CL algorithm.

2.2. Dataset encoding

It is known that ML models do not work with words or letters, but with numbers, so it is necessary to encode our dataset in a format understandable by the ML model. To encode the dataset, as the main indicator of a diphthong/hiatus appearance is two or more consecutive vowels in a word, we aim to give the most weight to these specific cases.

Our first approach was to encode our data on a character level, meaning each word is split in letters, and each of those is encoded using the *tf-idf* (term frequency–inverse document frequency) index [10]. This index allows most frequent letters in our dictionary to have a higher value, and the least frequent ones to be close to 0. This is very convenient, as accentuated vowels will most likely be the least frequent letters, and vowels, would be the most frequent ones, making big oscillations in case of a probable hiatus, and minor ones in case of a diphthong.

```python
def __get_index_for_token(char):
    nude_char = char.translate(str.maketrans('', '', string.punctuation))
    nude_switcher = {
        'a': 1,
        'e': 1,
        'i': 3,
        'o': 1,
        'u': 3,
    }
    value = nude_switcher.get(nude_char, 0)
    if value == 0:
        return value
    punctuation_switcher = {
        'á': 1,
        'é': 1,
        'í': 0,
        'ó': 1,
        'ú': 0,
    }
```

Fig. 1. Tokenization strategy for vowels.

But this approach, although it was able to obtain good results during training, could not learn any further because it was still taking into account consonants and wasn't measuring properly the weight that each vowel has and the variances between whether the vowel is accentuated or not.

With this in mind, we took a new approach simplifying the encoding part to a system where every consonant is set to weight 0, and vowels take a different weight, according to Spanish language diphthongs and hiatus rules (see Fig. 1).

The higher the weight of a vowel, the more it makes the machine believe it is processing a diphthong. The lower the weight, the more the tendency switches to hiatus. This encoding also makes it easier to tag the "none" group, as there will have no consecutive vowels thus no consecutive weights differing from zero. In addition to that, since this encoding is based on Spanish language rules, it can also estimate whether a made up word would be a diphthong or a hiatus.

3. Machine learning model and training

3.1. Model architecture

Our initial ML model was a simple artificial neural network model (ANN) composed by dense connected layers, but it tended quickly to overfit and has difficulties to generalize from the dataset. For this reason, we reduced the amount of dense layers and replaced the first ones by convolutional one dimensional layers, that try to establish relations from the input data to achieve a successful classification. Convolutional layers are interspersed with pooling layers to induce spatial-filter hierarchies by making successive convolution layers look at increasingly large windows (in terms of the fraction of the original input they cover).

We added also batch normalization layers (for normalizing the layers' inputs by re-centering and re-scaling), dropout layers (used to control overfitting by dropping a percentage of neurons randomly, at the cost of making the training process less accurate and more dependent on which neurons are dropped). The transition between the feature learning part and the classification part is done by a flatten layer that transforms the data into a one-dimensional array. Finally the classification part is a dense layer with a dropout layer and with a final layer that is a dense layer with

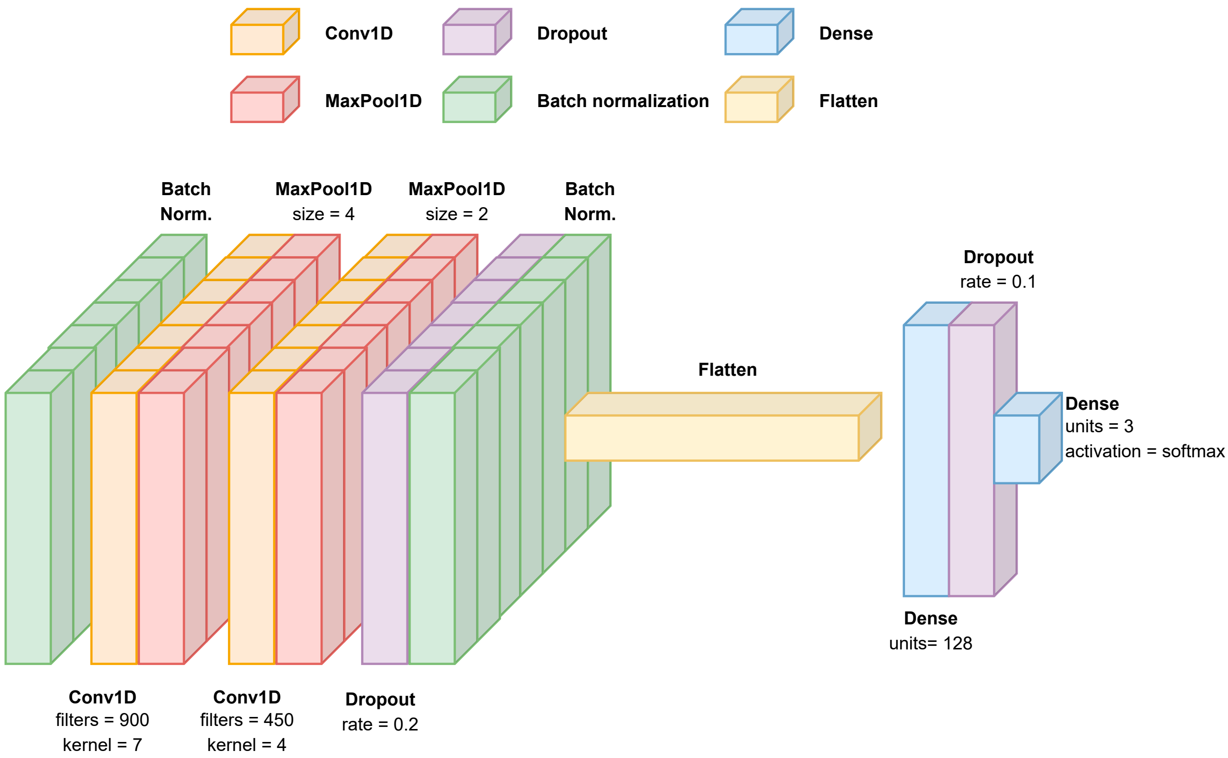

Fig. 2. Model architecture defined as a deep neural network.

three units, one for each class we aim to classify the data in, and a *softmax* activation function that assigns a given probability to each class. An schematic representation of the final model architecture can be seen in Fig. 2

3.2. Model training with Curriculum Learning

To train the model we used Curriculum Learning (CL), training from easier cases to harder cases. In a CL training we have to implement two elements: a *scoring function* and a *pacing function*.

In our case, the scoring function was a teacher with experience in teaching Spanish orthography, who labeled each of the cases with three levels of difficulty, so it would be considered a heuristic approach. The pacing function chosen was *Baby-Step*, that makes use of an incremental exposure to examples, segmenting them into groups by difficulty. That allowed us to track the learning process of the model while gradually incrementing the difficulty of samples.

The training results can be seen in Fig. 3, showing adequate training and validation behavior. The accuracy of the test data was 0.84 for a fully trained AI without using CL and of 0.89 using CL, which proves that a curriculum strategy is adequate for this dataset.

4. Machine Teaching experiment

The accuracy of the model developed identifies the upper limit of efficiency that we would expect to obtain in an MT experiment. The next phase is to reset the weights of the model but retain its architecture and offer the model to the students for its training. May be, to avoid a cold start, we can train the model with some cases so as not to make the training process so long.

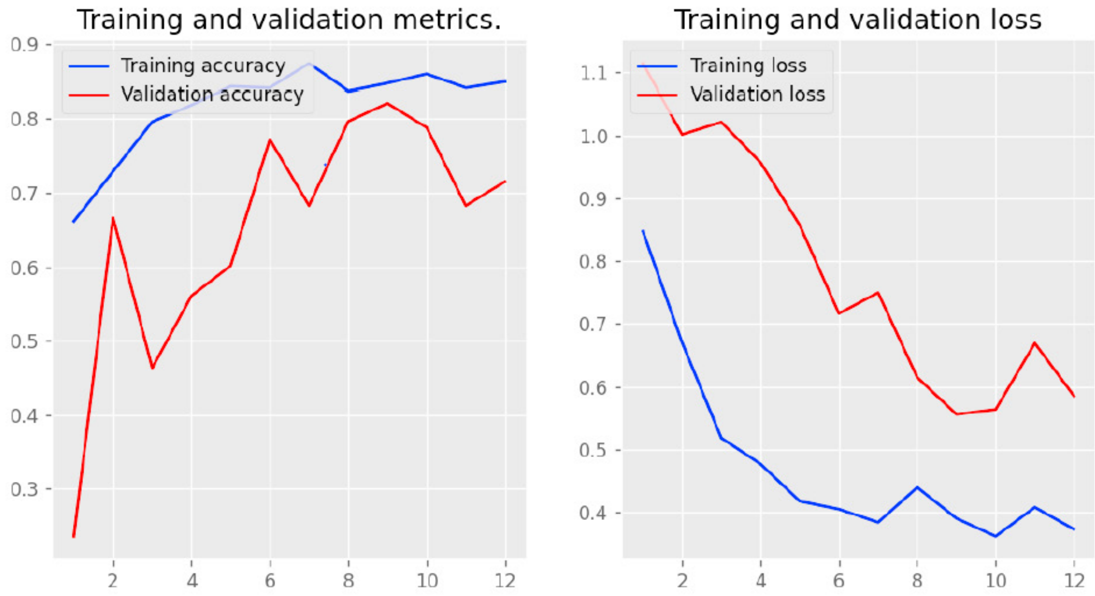

Fig. 3. Training and validation results of the fully-trained model.

The final phase and what, in our view, is the most novel aspect of this work, is that students now have to use MT techniques to make the model learn again, but the ultimate goal is that students learn by teaching in an informal and gamified learning environment. The workflow of this MT experiment is the following (see Figure 4):

1. **Initial training**: The teacher selects a small dataset with labeled examples that is used as the initial training of the ML model.
2. **Teacher teach examples**: The teacher uses the same small dataset of labeled examples to explain the linguistic criteria that define a diphthong and a hiatus and teach the students how to identify the easy ones.
3. **Teacher provides examples**: The teacher will provide to the students a large set of unlabeled words with diphthongs, hiatuses and words that are neither one nor the other.
4. **Students label the dataset**: The students are organized in groups and each group should label the unlabeled dataset words as hiatus or diphthongs (or neither) and identify their difficulty. It is not necessary to label the whole dataset at once, it can be done iteratively by identifying at least at the beginning some words in the three possible categories (diphthong, hiatus, none) and in the three levels of difficulty (easy, medium, hard).
5. **Students train the model**: This is the first step of an interactive MT process. The students will have to train the ML model supplying cases to them. Here, following the instructions of the teacher they should choose manually some pacing strategies (start with the easy cases, have a mixture of cases of each type, etc.).
6. **Feedback**: After the training of the ML model the students will receive feedback on how well it performs in the different categories. This information will serve to select the new cases for the next training cycle trying to solve the weaknesses of the model (they would act as a human teacher teaching his students).
7. **Rate model**: After several cycles of MT, the final model is obtained and the teacher rates each ML model using a test dataset with cases not present neither in the small labeled dataset nor in the larga unlabeled dataset.

Before testing the approach in a real classroom, in this work we have simulated the MT experiment, preparing the system to learn in different iterations and with the authors playing the role of children, feeding the system with different cases in incremental levels of difficulty and following various pacing strategies.

It could be seen that after each iteration the system improved its accuracy, which is important to motivate students to keep feeding their model with more cases so that it learns more. It was also possible to verify that, after the last iteration, the model showed an accuracy of 0.88, very similar to that obtained during the learning process using CL. It

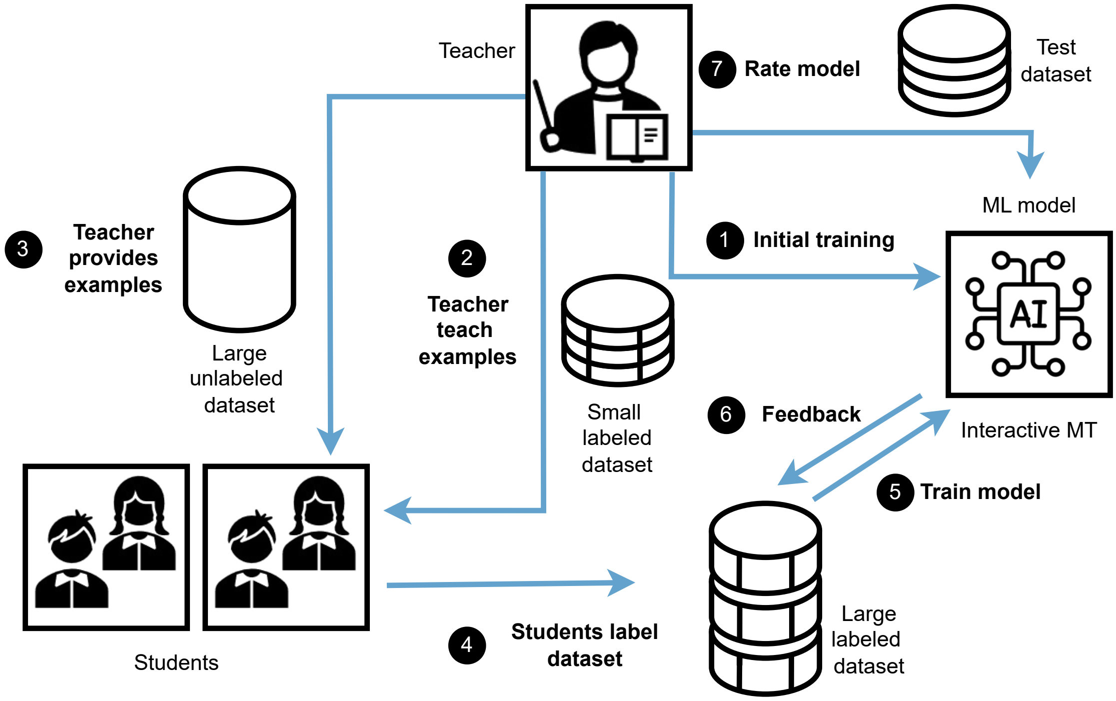

Fig. 4. MT experiment workflow.

is important that the final result is a high value, so that the model does not make too many errors. However, as already mentioned, due to the exceptions and inconsistencies of human languages, some kind of error in the classification is always to be expected.

5. Conclusions and future work

As a conclusion of this work we believe that, given that HITL techniques such as CL and MT were inspired by didactics, we can also use these techniques as didactic resource themselves. A simpler result could be obtained with a system that simulates learning, but we believe that its functioning would be more predictable and less realistic. Using our approach the model is actually learning, so the results will be more in line with the actions taken by the students (in this case acting as teachers of the model). If students mislabel cases, choose the wrong case difficulty, do not follow a consistent strategy for feeding cases to the model, and do not react correctly to their mistakes, the model will not learn.

As future work, we plan to test the model in a real learning environment, installing it in an app and using it with real children, control groups, etc. In addition, the HITL experiment could be completed with an Active Learning (AL) experiment in which not only the children feed the system, but the system can ask for help with new words obtained from a pool of unlabeled words. The system would choose a new word to be labeled following an exploration or exploitation strategy and the children would have to give the appropriate label to that word. We are also considering applying it to other types of problems where the ML model may be able to learn faster and more efficiently (image recognition, for example).

Acknowledgements

This work has been supported by the Spanish Government (PID2019-107194GB-I00/AEI/10.13039/501100011033) the Xunta de Galicia (ED431C 2022/44), the Centro de Investigación "CITIC" (ED431G 2019/01) with European Union ERDF funds and the "Cátedra de Educación y Tecnologías Emergentes, Gamificación e IA (EduEmer)" of the Universidad Pablo de Olavide.

References

[1] 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. URL: https://www.aaai.org/ojs/index.php/aimagazine/article/view/2513, doi:10.1609/aimag.v35i4.2513.

[2] 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. p. 41–48. URL: https://dl.acm.org/doi/10.1145/1553374.1553380, doi:10.1145/1553374.1553380.

[3] Cortese, C.G., 2005. Learning through teaching. Management Learning 36, 87–115.

[4] Guillot Suarez, C., 2022. Human-in-the-loop Hyperparameter Tuning of Deep Nets to Improve Explainability of Classifications. Master's thesis. Aalto University. School of Electrical Engineering. URL: http://urn.fi/URN:NBN:fi:aalto-202205223354.

[5] Hacohen, G., Weinshall, D., 2019. On the power of curriculum learning in training deep networks. arXiv:1904.03626.

[6] 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. URL: http://mau.diva-portal.org/smash/get/diva2:1428195/FULLTEXT01.pdf, doi:10.1109/PerComWorkshops48775.2020.9156175.

[7] Mosqueira-Rey, E., Alonso-Ríos, D., Baamonde-Lozano, A., 2021. Integrating iterative machine teaching and active learning into the machine learning loop. Procedia Computer Science 192, 553–562. URL: https://www.sciencedirect.com/science/article/pii/S1877050921015441, doi:10.1016/j.procs.2021.08.057. knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 25th International Conference KES2021.

[8] Mosqueira-Rey, E., Hernández-Pereira, E., Alonso-Ríos, D., Bobes-Bascarán, J., Fernández-Leal, A., 2023a. Human-in-the-loop machine learning: A state of the art. Artificial Intelligence Review 56, 3005–3054. URL: https://doi.org/10.1007/s10462-022-10246-w, doi:10.1007/s10462-022-10246-w.

[9] Mosqueira-Rey, E., Pérez-Sánchez, A., Hernández-Pereira, E., Alonso-Ríos, D., Bobes-Bascarán, J., Ángel Fernández-Leal, Moret-Bonillo, V., Vidal-Ínsua, Y., Vázquez-Rivera, F., 2023b. Human-In-The-Loop machine learning for the treatment of pancreatic cancer, in: 2023 International Joint Conference on Neural Networks (IJCNN) (IJCNN 2023), Queensland, Australia.

[10] Rajaraman, A., Ullman, J.D., 2011. Data Mining. Cambridge University Press. p. 1–17. doi:10.1017/CBO9781139058452.002.

[11] 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. URL: https://doi.org/10.1080/07370024.2020.1734931, doi:10.1080/07370024.2020.1734931.

[12] 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.

[13] 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:1707.06742.

[14] Zhu, X., 2015. Machine teaching: An inverse problem to machine learning and an approach toward optimal education, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI Press. pp. 4083–4087. URL: https://ojs.aaai.org/index.php/AAAI/article/view/9761.