

# Contextual machine teaching

Lars Holmberg, Paul Davidsson, Carl Magnus Olsson  
*Internet of Things and People Research Center*  
*Department of Computer Science and Media Technology*  
*Malmö University*  
Malmö, Sweden

lars.holmberg@mau.se, paul.davidsson@mau.se, carl.magnus.olsson@mau.se

Per Linde  
*Internet of Things and People Research Center*  
*School of Arts and Communication*  
*Malmö University*  
Malmö, Sweden  
per.linde@mau.se

**Abstract**—Machine learning research today is dominated by a technocentric perspective and in many cases disconnected from the users of the technology. The machine teaching paradigm instead shifts the focus from machine learning experts towards the domain experts and users of machine learning technology. This shift opens up for new perspectives on the current use of machine learning as well as new usage areas to explore. In this study, we apply and map existing machine teaching principles onto a contextual machine teaching implementation in a commuting setting. The aim is to highlight areas in machine teaching theory that requires more attention. The main contribution of this work is an increased focus on available features, the features space and the potential to transfer some of the domain expert’s explanatory powers to the machine learning system.

**Index Terms**—Machine learning, Machine Teaching, Human in the loop

## I. INTRODUCTION

Contemporary Machine Learning (ML) research is currently dominated by a technocentric perspective that primarily solves and addresses technical and functional goals but risks not placing enough focus on serving human values and ethical principles [1]. As an alternative, moving the agency towards domain experts and end users is an emerging and promising development in the field [2], [3]. A move in this direction can help democratize the knowledge and thus mitigate some of the risks of the problematic knowledge concentration of this disruptive technology [4]–[6]. An agency move in the direction of non-machine learning experts also offers the possibility to find and explore novel application areas that build on subjective knowledge production.

In Machine Teaching (MT) [7], [8], a human domain expert acts as a teacher in relation to a machine-based learner and consequently, domain knowledge becomes a natural part of the resulting deployment of an ML system. Previous studies mainly involve theoretical examples [8]–[11] in the area of MT and the majority of the concrete studies are in the medical [12], [13] and robotics field [14]–[16]. Also, to our knowledge, no studies currently exist that explore MT in a personal knowledge domain that includes contextual data. Focusing an MT approach in this direction could open up for a new type of individualization and personalization in areas such

This work was partially financed by the Knowledge Foundation through the Internet of Things and People research profile.

as Personal Informatics, Assistive Technology, and Intelligent Personal Assistants.

Simard et al. [7] see and treat MT as a discipline separate from traditional machine learning and have developed principles intended to be useful for MT implementations.

Our study is guided by a desire to understand what aspects of the MT paradigm [7] are the most important when it is applied on a contextual application in a personal knowledge domain. In this study we focus on answering the following sub-questions using a prototype we have developed for a commuting context:

- 1) What are the consequences, both in relation to the application and the MT paradigm, of applying the principles developed by Simard et al.?
- 2) What does a personal knowledge domain imply for the MT paradigm?
- 3) What does context dependence imply for the MT paradigm?

To answer the questions, we used the principles developed by Simard et al. [7] as an analytic lens, a way of understanding the area and potentially strengthen the MT paradigm, in conjunction with a user study. The study spanned eight weeks, involved eight participants and one meeting every week.

In the remaining part of the work, we present the project background, outline our framing, our selection of MT principles and the methodological approach we applied. Thereafter we describe our prototype and the study setup. In the analysis and discussion, we relate MT principles to our prototype and identify areas where our work suggests that more research can bring the area forward. We end with a conclusion section where we elaborate on additional perspectives that needs further research.

## II. RELATED WORK

Machine teaching differs from Active Learning (AL) [17] and Interactive Machine Learning (IML) [18] in that the teacher knows that, at some point in time, the extent of the knowledge they want to transfer to the learner [7]. The focus in MT is on that predictions made by the machine learner has to be interpretable, for the teacher, in relation to the performed teaching. If the system is constructed with a specific domain in focus, the role of the ML expert can be reduced so a domain expert can own the process of refining the model’s

knowledge. The benefit of IML/MT systems in relation to more traditional ML systems "is observable in situations where the precise design objectives of the user are unclear and/or data labels cannot be obtained a priori" [19]. MT and IML systems often facilitate short iteration loops where the model can be incrementally updated in small steps. Simard et al. states the role of the teacher as the person responsible for transferring knowledge using a teaching tool to the machine learner, in order to produce a model that can approximate a *concept*. Central to the teaching is the process when the teacher *selects* and *labels examples* that the teacher believes represents a *concept*.

Simard et al. [7] propose the following principles for the domain specific language and the process of MT.

- The language should be *universal* over a domain and easy to learn. If they have the required domain knowledge, this results in the domain expert being interchangeable. The ML method becomes exchangeable, given that the MT language is independent of the ML method used.
- The teaching tool should be implemented it is *feature complete*, implying that there are enough features for the teacher to separate all the desired target *concepts*. *Feature completeness* is seen as a responsibility of the tool, not the teacher.

According to Simard et al. [7] the teaching process places demands on the *sampling distribution*, *training distribution*, and the *deployment distribution*. The *sampling distribution*, from which the *examples* are selected, must be rich and diverse so that the teacher can select a *training distribution* that generalize well on the *deployment distribution*.

### III. METHODOLOGY

To explore and challenge contextual MT we applied the principles developed by Simard et al. [7] as an analytical lens on a prototype. We chose Participatory Design [20] as a frame for the research, as it allows for adaptations during the study in order to handle new insights. We recruited eight participants for a study that lasted eight weeks. The evaluation is an application grounded evaluation with real humans and real tasks [21].

#### A. Research setting

We selected commuting as our application area since it is accessible and easy to relate to. Commute patterns are individual and predictions are contextually dependent. We built a prototype giving a commuter control over the transfer of their commute pattern to the ML model. The MT interface of the prototype can be seen in Fig. 1 and the prediction part in Fig. 2.

We used the capabilities of modern mobile phones to collect contextual features [22], such as time, activity and location, used as context cues in order to make journey predictions and collect training data. The model is deployed without any initial data about the user's commute pattern and thus cold started in order to be trained using the teaching interface (Fig 1). In this work, we distinguish between different usage modes: *in*

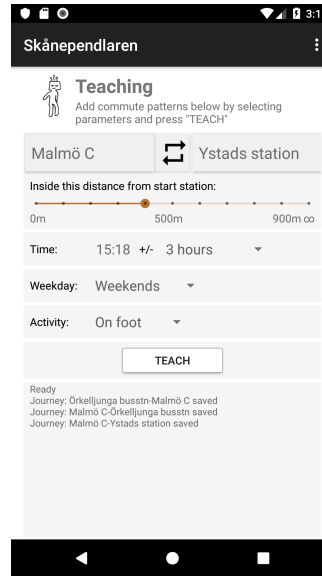


Fig. 1. Teaching interface.

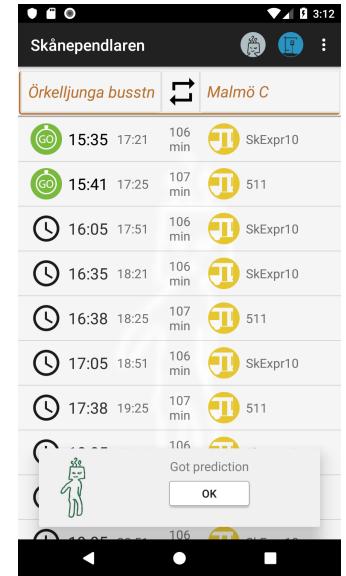


Fig. 2. Prediction interface.

*context teaching*, *out of context teaching*, *in context predicting* and *out of context predicting* (We did not implement out of context prediction at this stage). In context teaching implies that at least one feature from the context cue is used. Out of context teaching implies that none of the features maps to the user's current context cue and therefore all desired features have to be manually selected. In context, prediction implies that all the features in the context cue are sent to the ML model so a prediction can be made and out of context prediction imply that parts of the context cue are simulated. The contextual features we selected as relevant for this prototype were location, day of week, time of day and the commuter's current activity (walking, still, running and in vehicle). A prediction is done when the app is started and the contextual features are sent to the trained model. The resulting prediction is then used to collect and present information about upcoming departures for the predicted journey (Fig. 2).

We used one individually trained Neural Network (NN) for each user as ML method. The choice of NN was primarily made since the selection simplifies prototyping in that NN handles raw input data and can handle multiple levels of representation [23]. For the implementation of the neural network, we used the framework of fast.ai [24] on a cloud server. A real-time database (Firestore), a NodeJS server and a Flask server was used for the machine learning pipeline and synchronization. Prediction turn around time is on average under one second and retraining time on average around 15 seconds. We used a network with two hidden layers that we feed with the contextual feature values: latitude (double), longitude (double), minute of day (integer minutes from midnight), weekday (integer one to seven) and activity (integer that represents walking, still, running and in vehicle). Weekday and activity are handled as categorical features and the other as continuous.

To handle the cold start situation the commuter selects a sub-space in the feature space consisting of feature ranges, for example, a time-span and one or more days like weekend using the teaching interface (Fig 1). We then generate synthetic examples inside the selected feature sub-space. To find reasonable prediction accuracy we performed a functionally grounded evaluation [21] with data created for three personas. We iterated to find an appropriate size of the network (200|100 neurons), number of epochs (7) and a reasonable number of generated examples for a concept (40) to fulfill our prototype needs. We used an automatic learning rate finder [25] implemented in fast.ai to optimize the learning rate. The examples we generate are spread randomly using a uniform distribution over the feature sub-space that the user selects. For example, if you set weekday to "Any day" the examples are randomly spread over the days of the week. If you select "Friday" all examples have the value of 5 as training data. This is, of course, a simplification of the machine learning problem at hand but the goal at this stage was to build a working prototype for this study. Once we got the prediction performance acceptable, the back-end implementation (machine learning pipeline and model configuration) was frozen. This was done so we could get to a point where we could study and tweak our contextual MT implementation separated from an invariable machine learning back-end.

### B. Recruiting participants for the study

For the participatory design phase, we recruited users with a background in graphic design, interaction design, user experience design, programming or agile development. Our goal with this selection was the possibility to discuss and elaborate aspects of MT during the study with users who are accustomed to these types of discussions from both a technical perspective as well as a usage perspective. To recruit we mailed information about the project to some selected student groups and companies. Ten persons showed interest and we selected eight of them for the study. All the participants were under the age of 40, four were female and four were male. Six out of eight participants were university students ranging from first-year bachelor to master's level in the range of between 20 and 30 years. One participant works in the industry as primarily UX designer and product owner (age 36) and one as a project manager (age 39). The two participants at master's level were international students. One requirement when recruiting users was that the users had to commute and own an Android phone. The users were rewarded with one month of free commuting during the study.

### C. Participatory generated data

The study was conducted during eight weeks with weekly scripted recorded discussions with the participants. Initially, one to one meetings were held with the students to get to know them better. We then went to groups of two from a similar field and ended with groups of three or four with mixed competences. The intention with this was to create an atmosphere of creative discussion climate and interesting

group constellations. The meetings started with a follow up on previous week's experiences where metrics and redesign proposals were presented and discussed. In addition, the following week's focus and tasks were discussed. Initially, the participants were introduced to the MT paradigm, and during the study, they evaluated different teaching strategies and gradually became accustomed to the paradigm and our prototype. In the later sessions, the participants were able to, in varying degrees, abstract the MT paradigm from the commuting context and thus give input that can be generalized to other domains. Using content analysis [26], interviews and workshops were broken down into codes originating from the MT vocabulary used by Simard et al. [7].

## IV. ANALYSIS AND RESULT

In this section, we use the MT principles developed by Simard et al. [7] to better understand our contextual prototype and lift out aspects that come into focus. Those aspects in focus are further elaborated on their own or in conjunction with findings from our user study using the same prototype. The prototypes capabilities was summarized as follows by one study participant:

The first week we had this app I was quite impressed. [...] So I had only taught it that on "Mondays at this time and place I would like to travel from Malmö C to Svedala". Then on Tuesday when I opened the app it knew that this is the journey you would like to take. So it can know things that you didn't teach in detail, but only approximately.

### A. Teaching algorithm

Our MT algorithm (Algorithm 1) differs at some key points from the process proposed by Simard et al.

- The user can teach a concept both in and out of the target context (the two while loops). This means that you can, for example, teach your complete commute pattern from home by manually selecting all features in Fig. 1 or you can, teach just before you take a transport by using the context cue.
- When teaching, examples are not selected from a sampling set but rather a sub-space in the feature space is defined in the teaching interface (Fig. 1) as belonging to a desired concept. The machine learning back-end then generates synthetic examples within this sub-space.
- The user can only test towards a quality criterion when in the target context, in this case, this means to evaluate that the correct journey is predicted (the 'if-elseif-else' statement).
- Missing features cannot be added so feature blindness cannot be handled (the 'else-if' part) Fig. 1.
- If prediction errors occur due to inconsistent teaching, labeling errors, overlapping concepts or because the teacher distrust the predictions, the only alternative is to delete all training data and add all the desired concepts again (the 'else' part).

```

repeat
  //Teaching out of context
  while concept is realizable do
    if desired concepts added then
      | exit;
    end
    Add concept (generated examples) to training
    set;
  end
  //Fix prediction error (user in context)
  if caused by concept not added then
    //Teaching in context
    while concept is realizable do
      if quality criteria is met (correct journey
      predicted) then
        | exit;
      end
      Add concept (generated examples) to
      training set;
      Retrain;
    end
  else if caused by feature incompleteness or concept
  intentionally not added then
    | Search journey manually;
  else
    Delete training data (currently all examples in
    training set);
  end
until forever;

```

**Algorithm 1:** Contextual MT process

## B. Teaching strategies

During week two of the study, the participants were instructed to do out of context teaching only once at the beginning of the week. During the third week, they did in context teaching when the prediction made was wrong compared to their intended journey. The prototype was reset prior to these tests. During the following weeks, the participants decided their own teaching strategies. Comment from one of the users on in context teaching compared to previous weeks out of context teaching:

It was a lot messier than last week. Eh.. harder to add things and you were more stressed and not that particular.. I had a lot less control over the things I added compared to last week.

In out of context teaching, the users tended to teach general concepts directly (using, for example, the selection "weekdays" instead of individual days). During in context teaching most participants changed some of the feature ranges to make the teaching more general, especially regarding time span, but only one made all teaching as general as possible. All users found the predictions interpretable in relation to what they had taught the artifact, even if the predictions as such were wrong in relation to their current context. It was harder to tell if the predictions were logical when they did in

context teaching since the users found it difficult to remember if the whole concept had been taught or not. One of the users had complex commute patterns and experienced feature blindness (she works at two places and travels there depending on conditions that cannot be represented using the available features).

One user argued for why she, in theory, would prefer in context teaching, even if she, in this case, preferred out of context teaching:

In theory, I think I like in context teaching better since it is real events that occur and I only save them. But with the experience at the moment in the app, it feels better to add those teachings once. But that behavior could probably change if the app is changed.

A comment on combining the teaching strategies:

Can't you combine both? The first time you open the app as a new user you get an out of context teaching interface and add your most common journeys. And then when you done, you could teach it via a fast-teach-button, you get thumbs up/down in relation to a prediction.

## C. Prediction in and out of context

In our implementation, the commuter can only receive predictions when in context. We are primarily interested if the teacher finds the in context predictions interpretable in relation to the performed teaching. The model's prediction accuracy becomes less interesting from a system level, as long as the prediction is correct, or as one participant expressed it:

It's more interesting for the developer, for you maybe. But not really for the user. I mean I just need the bus.

The users with simple commuting patterns, for example to and from work and a few other non conflicting regular activities, found that the predictions were interpretable. When the teaching became complex or fragmented over time it was harder to conclude, from the commuter's perspective, if the predictions were logical in relation to the previously performed teaching.

Our study found that it is difficult to tell whether this was due to the choice of ML method, the examples created by the back-end, that the teacher was inconsistent in his/her teaching or whether the teacher forgot if a concept was completely taught. As one participant expressed it "I was more impressed in the beginning, but after a while, I think it got confused by all the teaching". Another comment was "there should be a way to check what it knows". In our implementation, when there is a mismatch between the expected predictions and the learned concepts, the only option is to completely delete the training distribution and retrain the complete commute concept. Our approach works well for the participants in the study who had simple commute patterns, so the complete commute pattern could be retrained out of context quickly and easily within a few minutes time or as an alternative retrained in context

for each journey concept that appears as a context cue. The usual reaction from the participants is to reteach the intended journey and check the prediction. This situation is important because if the prediction is not corrected it is easy to mistrust the app. In any case, the ability to visualize the teaching done or probe predictions would be an important addition. As one participant expressed this:

But from the teacher’s perspective when I teach him, can you show me what I teach him? ... So I know I am the teacher.

#### D. Cold start

In a personal knowledge domain, as we defined it in this study, no labeled or unlabeled examples exists initially, only an empty feature space limited by the available features and their ranges. Similar cold start challenges will likely appear for MT targeting the areas of Personal Informatics, Intelligent Personal Assistants and Assistive Technology. In these areas, it will likely be challenging to collect enough rich and diverse data in the background so a teacher can search and select examples representing all envisioned concepts. In context teaching and evaluation, as our study indicates, in an iterative fashion is then an interesting alternative approach. This does not rule out using previously collected data or using models transferred from similar domains or situations as a starting point.

#### E. Transfer concept to learner

The teacher’s complete commute patterns are, on a top level, the concept they want to transfer to the learner. This top concept is for our context broken down by the teacher to mutually exclusive sub-concepts in a tree structure that, depending on the teacher, could be, for example, on the lowest level, “Traveling from home to work” or “Going home from Sara”. In our case, the teacher can readily express those concepts in terms of the available features. This decomposition and translation to feature ranges is, in our case, done using an implicit schema in the teacher’s head and is not represented or saved in the application. If the knowledge domain were public or shared, many concepts would be part of the domain specific vocabulary for that domain. In our personal case there could be pre-created concept names like “Going to work” etc. and/or a possibility to name mundane personal concepts. The possibility to name and use concepts during teaching was not requested by the study participants, but it stands out as a result of this study that would be interesting to study further and as an important part of a teaching language.

#### F. Teaching Language

In this study, we created a domain specific MT language that can aspire to be universal across the commuting domain. By using our language, a commuter can teach a learner his/her commute patterns as long as an application is connected to a relevant service from a transport provider.

Limitations in our language implementation stem from the fact that our implementation is not feature complete and we cannot add missing features. For our prototype, we adopted a

weaker alternative approach to feature completeness in that we allow users to ignore existing features, to make it clear that, for example, a journey prediction is only dependent on location and not time, day and activity. To make this more explicit for our study participants in order for, feature completeness to be discussed in our study, we made this option clearer in a later iteration of the prototype. All our study participants found location, day of the week, and time to be relevant and useful features but none of the participants found activity (still, walking, running or in vehicle) to be a useful feature. The teachers in our case have access to the complete feature space and this gives them the possibility to teach a concept in any part of this space. In comparison, the principles by Simard et al. [7] only gives access to the feature space through existing examples in the sampling distribution, this then has to be rich and diverse in order to offer a possibility to teach all intended concepts.

In relation to those principles, we have pre-created the composed features weekdays and weekends so the user can select them directly. In contrast to the principles developed by Simard et al. [7] we do not give the users a possibility to define, name and compose new features for example “Mornings”, “Mornings on weekends at Helens’s place” or “Night”. Instead, teaching must be carried out a number of times for one concept instead of reusing created composed features. Additional features suggested and discussed by our study participants were personal calendar, travel time, preferred means of transport and weather data.

## V. DISCUSSION

In this section, we outline and compare our results and analysis, with related MT theory and practice.

At the heart of MT lays the teacher’s domain knowledge and consequently the capability to map an example with a label representing a concept. In our case, when the app is first started, no labeled or unlabeled examples exist and the commuter has no knowledge of the ML method the learner’s uses. In the work by Simard et al. [7], there are examples representing the concepts to choose from in a sampling distribution. In our case, the lack of examples is compensated by the teacher using their domain knowledge to select a sub-space in the feature space that represents a concept. Our approach is a direct result of treating the domain as a personal knowledge domain that uses a limited number of human interpretable features and that no examples have been collected by the user prior to the initial teaching of a concept. Our approach emphasizes the construction of a learner that, in relation to the concepts taught and the commuter’s current context, includes an ML method that gives predictions that are interpretable. This in order for the teacher to be able to, over time, create a cognitive model over how to teach the model.

MT systems must take into account that the concepts and the schemes that connect them are not stable over time in relation to the teacher and in relation to the world. These are challenges that we have not addressed in this work. We did also not save any training data during use, making

use of this data could be one possible route, from an ML perspective, to fine-tune the system and address concept- and covariant- shift [27]. Combing a domain specific taxonomy and a possibility to name mundane concept with the possibility to rearrange concept data seems like a useful approach, from a teacher's perspective, to address dataset shift. This possibility to rearrange, add and delete concept also opens up for a domain expert to address their knowledge development and shifting usage needs within the domain.

Traditional programming languages often aim to be oblivious about the usage domain and more specifically in relation to the intended technical platform or type of problem it can solve. In contrast, teaching languages are specific to the usage domain and closely connected to the vocabulary of that domain. Some features, like time and location, will be represented and needed in many contextual teaching languages given that they are used in many domains. Central to a teaching language for any domain is the available features that the language gives access to. Here, as Simard et al. [7] and our work points out, it is central that features can be created, added, removed, composed and that sub-spaces in the feature space can be ignored. In our case, when the knowledge domain is personal, the need for a flexible teaching language is central, but will probably not cover all individual cases. From our case, the weather situation could be relatively straight-forward to add, at least for in context teaching and in context predictions, but integration with a personal calendar is a complex feature expansion that needs special attention.

## VI. CONCLUSIONS

We set out to apply principles for MT [7] onto a prototype with the goal of gaining a better practical understanding of the MT paradigm in a contextually dependent setting. We found, in relation to our first research question, that our selected MT principles make it possible to separate and discuss MT as a discipline in its own right and that a domain expert could apply, teach, discuss and evaluate a learner independent of the machine learning expert. In relation to our second research question, we found that MT can be a fruitful and promising approach to make machine learning useful in supporting personal and individual needs, useful for example in the area of intelligent personal assistants. Regarding the third research question, we identified areas of the MT paradigm that needs further research to be applicable in a contextually dependent domain; in context teaching, teaching over time, possibilities to probe the learner for learned concepts and interpretations of predictions given. In terms of future research, we suggest a direction and a focus on integrating the explanatory powers of humans as a natural part in machine learning systems via an MT approach with an aim towards building the systems we want and not solely the one we can.

## VII. ACKNOWLEDGMENT

I would like to thank Jonas Löwgren for invaluable support during the work.

## REFERENCES

- [1] IEEE. Ethically aligned design. Technical report, 2019.
- [2] Paul Dourish. Algorithms and their others: Algorithmic culture in context. *Big Data & Society*, 3(2):205395171666512, 2016.
- [3] Wiebe E Bijker. Do Not Despair: There Is Life after Constructivism. *Science Technology, & Human Values Sage Periodicals Press*, 18(114):113–138, 1993.
- [4] Shoshana Zuboff. Big other: Surveillance capitalism and the prospects of an information civilization. *Journal of Information Technology*, 30(1):75–89, 2015.
- [5] Cathy O'Neil. *Weapons of math destruction: How big data increases inequality and threatens democracy*. Broadway Books, 2016.
- [6] Nick Couldry and Ulises A. Mejias. Data Colonialism: Rethinking Big Data's Relation to the Contemporary Subject. *Television & New Media*, 20(4):336–349, 5 2019.
- [7] Patrice Y. Simard, Saleema Amershi, David M. Chickering, Alicia Edelman Pelton, Soroush Ghorashi, Christopher Meek, Gonzalo Ramos, Jina Suh, Johan Verwey, Mo Wang, and John Wernsing. Machine Teaching: A New Paradigm for Building Machine Learning Systems. Technical report, Microsoft Research, 2017.
- [8] Xiaojin Zhu, Adish Singla, Sandra Zilles, and Anna N. Rafferty. An Overview of Machine Teaching. *arXiv preprint arXiv:1801.05927*, 2018.
- [9] Yuxin Chen, Adish Singla, Oisín Mac Aodha, Pietro Perona, and Yisong Yue. Understanding the Role of Adaptivity in Machine Teaching: The Case of Version Space Learners. *Advances in Neural Information Processing Systems*, pages 1476–1486, 2 2018.
- [10] Weiyang Liu, Bo Dai, Ahmad Humayun, Charlene Tay, Chen Yu, Linda B. Smith, James M. Rehg, and Le Song. Iterative Machine Teaching. *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 2149–2158, 2017.
- [11] Shike Mei and Xiaojin Zhu. Using Machine Teaching to Identify Optimal Training-Set Attacks on Machine Learners. *Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2871–2877, 2015.
- [12] Martin Lindvall, Jesper Molin, and Jonas Löwgren. From machine learning to machine teaching. *Interactions*, 25(6):52–57, 10 2018.
- [13] Andreas Holzinger. Interactive machine learning for health informatics: when do we need the human-in-the-loop? *Brain Informatics*, 3(2):119–131, 6 2016.
- [14] Maya Cakmak, Crystal Chao, and Andrea L Thomaz. Designing Interactions for Robot Active Learners. *Computing in Science & Engineering*, 2(2):108–118, 2010.
- [15] Guanglong Du, Mingxuan Chen, Caibing Liu, Bo Zhang, and Ping Zhang. Online robot teaching with natural human-robot interaction. *IEEE Transactions on Industrial Electronics*, 65(12):9571–9581, 2018.
- [16] Luka Peternel, Tadej Petrič, and Jan Babič. Human-in-the-loop approach for teaching robot assembly tasks using impedance control interface. *Proceedings - IEEE International Conference on Robotics and Automation*, 2015-June(June):1497–1502, 2015.
- [17] Burr Settles. Active Learning Literature Survey. Technical Report January, University of Wisconsin-Madison, 2010.
- [18] Jerry Alan Fails and Dan R. Olsen. Interactive machine learning. In *Proceedings of the 8th international conference on Intelligent user interfaces - IUI '03*, page 39, 2003.
- [19] John J. Dudley and Per Ola Kristensson. A review of user interface design for interactive machine learning, 6 2018.
- [20] Robertson, Toni, Simonsen, and Jesper. *Routledge International Handbook of Participatory Design*. Routledge, 2012.
- [21] Finale Doshi-Velez and Been Kim. Considerations for Evaluation and Generalization in Interpretable Machine Learning. pages 3–17. 2018.
- [22] Albrecht Schmidt. Advanced Interaction in Context. *First Symposium on Handheld and Ubiquitous Computing (HUC '99)*, pages 89–101, 1999.
- [23] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
- [24] Fast.ai. fast.ai, 2019.
- [25] Leslie N Smith. Cyclical Learning Rates for Training Neural Networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472. IEEE, 2017.
- [26] U.H Graneheim and B Lundman. Qualitative content analysis in nursing research: concepts, procedures and measures to achieve trustworthiness. *Nurse Education Today*, 24(2):105–112, 2 2004.
- [27] Jose G. Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V. Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521–530, 2012.