A Taxonomy of Interactive Online Machine Learning Strategies

Agnes Tegen [✉], Paul Davidsson, and Jan A. Persson

Internet of Things and People Research Center, Department of Computer Science and Media Technology, Malmö University, Malmö, Sweden agnes.tegen@mau.se

Abstract. In interactive machine learning, human users and learning algorithms work together in order to solve challenging learning problems, e.g. with limited or no annotated data or trust issues. As annotating data can be costly, it is important to minimize the amount of annotated data needed for training while still getting a high classification accuracy. This is done by attempting to select the most informative data instances for training, where the amount of instances is limited by a labelling budget. In an online learning setting, the decision of whether or not to select an instance for labelling has to be done on-the-fly, as the data arrives in a sequential order and is only valid for a limited time period. We present a taxonomy of interactive online machine learning strategies. An interactive learning strategy determines which instances to label in an unlabelled dataset. In the taxonomy we differentiate between interactive learning strategies when the computer controls the learning process (active learning) and those when human users control the learning process (machine teaching). We then make a distinction between what triggers the learning: active learning could be triggered by uncertainty, time, or randomly, whereas machine teaching could be triggered by errors, state changes, time, or factors related to the user. We also illustrate the taxonomy by implementing versions of the different strategies and performing experiments on a benchmark dataset as well as on a synthetically generated dataset. The results show that the choice of interactive learning strategy affects performance, especially in the beginning of the online learning process, when there is a limited amount of labelled data.

Keywords: interactive machine learning · online learning · active learning.

1 Introduction

The performance of a machine learning method is dependent on the labelled data it is trained on. While the amount of data in the world is increasing at an accelerating rate, the act of labelling that data is often costly. In active learning, a limited number of data instances are chosen for labelling, as compared to all of them [14]. The aim is to achieve an equivalent performance to when all instances are labelled, by choosing the limited number of instances wisely. The active learning strategy selects which instances should be included and an oracle is then queried to provide correct labels for those instances. In machine teaching, the goal is also to achieve a high performance with a limited number of labelled samples, but with the teacher selecting the labelled instances to train the learner on [18, 19]. Often the oracle, in the case of active learning, or the teacher, in the case of machine teaching, is a human interacting with the system and we will henceforth adopt the term user to describe both.

In settings where data arrive as streams, the selection has to be done differently compared to a pool-based setting. In an online learning scenario, the data arrives in a specific sequential order and the current data instance is the only one that can be labelled at a given point in time. Furthermore, if the machine learning method is supposed to produce estimations in real-time, the computations might have to be done at the edge where algorithms often have restrictions on computational complexity. Most work where interactive machine learning strategies are employed focus on active learning, especially strategies based on the uncertainty of the learner. Several works compare different strategies, but do often include only one type of interactive machine learning strategies. Particularly, machine teaching in a setting with streaming data is an area which needs further exploration.

In this work we present a taxonomy of interactive online machine learning strategies. We include active learning strategies, where the learner queries the user, and machine teaching strategies, where the user provides labels by their own initiative according to a specific strategy, as well as hybrid versions. Some of the strategy types included in the taxonomy are abstractions of previously proposed strategies, while others are novel, at least in the given problem setting. We implement versions of the presented strategies and compare them through experiments on a benchmark dataset and a synthetically created dataset.

2 Related work

Fu et al. surveyed and organized existing work into two main categories based on whether the active learning strategy solely is based on the uncertainty of independent and identically distributed instances or if it also takes into account instance correlations [5]. Experiments comparing time complexity of the strategies discussed were carried out. While active learning for streaming data was mentioned as an emerging application and the related challenges discussed, the strategies are presented in a pool-based setting, where all unlabelled data is available at the same time. The strategies categorized as only being based on uncertainty can apply to an online learning setting, but the strategies presented where the instance correlations are included tend to be less suitable. First, they generally have a higher time complexity, which often is not suitable when estimations have to be produced in real-time. Second, they take into account a data instance's correlation to other instances, which means that a batch of data has to be collected before the strategy can be properly employed.

Active learning within a streaming data setting was explored by Miu et al. [13]. The authors present an online active learning framework, where user-provided annotations can be collected in real-time and used for Human Activity Recognition tasks. The framework is compared to more naïve annotation methods and evaluated on benchmark datasets. User studies were also carried out, where the framework was implemented in a mobile app through which the participants could provide labels. The user cannot provide labels by their own initiative however, as in machine teaching, but only when queried by the active learning strategy which is based on uncertainty of the learner.

Lughofer investigated active learning for data streams using evolving fuzzy classifiers [10]. Two different strategies of active learning are presented and tested, based on *conflict* and *ignorance*. *Conflict* means that the new data sample is on the border of previously defined classes, while *ignorance* means that the new data sample is far outside the previously defined borders. Both *conflict* and *ignorance* based learning can be considered strategies that are based on the uncertainty of the learning model. The paper does not explore a user providing labels by their own initiative.

A user with proactive capacities was studied by Chen et al., as they explored the role of adaptivity in algorithmic machine teaching [3]. In the work, the authors studied the teaching of version space learners in an interactive setting through experiments. An adaptive teacher, where the learner's hypothesis at every time step can be observed, was compared to a non-adaptive teacher, where only the initial hypothesis of the learner is known i.e. no feedback is received during teaching. In the first one, the teacher can proactively adapt the information provided to the learner on-the-fly, based on the performance. This can be seen as an interactive learning strategy where the user provides a label when the estimation of the learner is incorrect. The latter one can be seen as a pool-based setting where all examples are constructed beforehand. In the given setup, the learner did not have the possibility to query the teacher for labels however, and can only train on the instances provided by the teacher.

3 Interactive online machine learning strategies

The different categories of sampling strategies presented below are all interactive online learning strategies. Interactive learning includes both active learning, where the learner queries the user, who in turn responds with a label and machine teaching, where the user proactively provides labels based on a selection strategy. Online learning means that the data stream is received in a single-pass manner, i.e. the strategy is processing one data instance at a time. Special cases of sampling strategies that are meant to be combined with specific machine learning methods (e.g. evolving models [11]) are not included. In Fig. 1, a visualization of the taxonomy can be seen.

In interactive online machine learning, several issues have to be taken into consideration that might not be as relevant for interactive machine learning in a pool-based setting. The most prominent issue is the one-pass manner in which the data instances arrive. Within an online setting a decision must be made for each new instance as it appears, whether or not to query for (in the case of active learning) or provide (in the case of machine teaching) the label. Streaming data also needs another approach for dealing with labelling expenses compared to a pool-based setting, since the incoming data instances in theory could be infinite. In the experiments the labelling expenses, unless otherwise stated, are calculated based on the the labelling status of recent instances. Other suggestions to handle the calculation over of budget with streaming data are presented for instance by Kottke et al. [8] and Žliobaitė et al. [20].

For each type of strategy, an algorithm illustrates the structure in pseudo code. It is assumed in the following descriptions that the user never is fallible or reluctant, i.e. that the user always will provide a correct label in accordance with the interactive machine learning strategy in question. Each algorithm has a data stream $X = \{x_0, x_1, ...\}$, a labelling budget B and classifier Ψ as input. In the case of additional input, it is further described for that particular strategy. The data stream contains a possibly infinite sequence of data instances, where x_0 is the first instance received by the learning algorithm. The labelling budget B determines the maximum ratio of incoming samples that can be labelled, with the restriction $0 < B \leq 1$, where $B = 1$ means that all incoming instances can be labelled. The classifier Ψ specifies which machine learning method is used along with any parameter settings. The classifier in its initial state can be pre-trained, if applicable data exists, or it might not have been previously trained, in the case of a cold-start scenario. The aim for the classifier is to accurately classify a state y. The classifier produces an estimate of the state $\hat{y}_i$ for each data instance x_i. The user provides a label y_i in accordance with the interactive learning strategy.

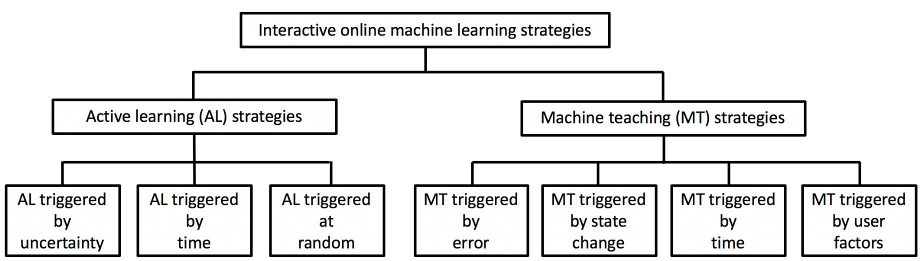


Fig. 1: Taxonomy of the interactive online machine learning strategies.

AL triggered by uncertainty *AL triggered by uncertainty* is the most commonly used interactive learning strategy. The learner queries the user for labels for the data instances where it is least certain how to label [14]. Different implementations of the strategy exist and a lot of work in the area focus on comparing and developing strategies where the querying is triggered by uncertainty. As algorithm 1 shows, the decision whether to query or not is based on the output from an uncertainty measurement function $U(\hat{Y}_i, x_i)$, where $\hat{Y}_i$ is the possible classes and x_i is the current data sample, which is compared to a threshold value θ.

Depending on which strategy is chosen for the function U and which classifier Ψ is used, $\hat{Y}_i$ might contain everything from only the most probable class $\hat{y}_i$ to all the possible classes ranked by probability. The implementation is especially straight-forward for probabilistic classifiers, but can be used with any classifier where an appropriate uncertainty measurement can be defined. Different versions of *AL triggered by uncertainty* include for example *least confident*, *margin sampling*, *entropy* and others where a region of uncertainty is explored [4]. In *least confident*, the learner queries when the probability of the prediction is low, *margin sampling* when the margin between the most probable and the second most probable class is low and in *entropy* when the combined probability of all possible classes is low.

Algorithm 1 AL triggered by uncertainty

Input: data stream $X = \{x_0, x_1, ...\}$, labelling budget B, classifier Ψ, threshold θ, step size s
$i \leftarrow 0$
repeat
  receive next data sample x_i
  $\hat{y}_i \leftarrow \Psi(x_i)$
  if $U(\hat{Y}_i, x_i) < \theta$ **and** $\hat{b} < B$ **then**
    ask user about label y_i for x_i
    $\Psi \leftarrow$ incremental learning procedure$((x_i, y_i))$
    $\theta \leftarrow \theta - s$
  else
    $\theta \leftarrow \theta + s$
  end if
  update labelling expenses $\hat{b}$
  $i \leftarrow i + 1$
until end of data stream

In the experiments described in section 4 we include *least confident*, where the uncertainty function becomes $U(\hat{Y}_i, x_i) = 1 - P(\hat{y}_i \mid x_i)$. The inverted probability is compared to a threshold θ which determines whether the classifier is uncertain regarding its prediction. The value of the threshold is set initially, but can be time-variable [20], meaning that it can be increased or decreased with the step size s depending on whether or not a query was posed. Note that if $s = 0$, the threshold is constant.

A sliding window is used to calculate the current labelling expenses $\hat{b}$. The window contains the labelling status of the most recent data instances. Each element of the window is either a 0, if there was no label provided by the user, or 1, if the user did provide a label. The labelling expenses are calculated by computing the ratio of how many instances in the window has had an accompanying label provided by the user. This value is compared to the labelling budget

B to decide whether the strategy can query the user. The labelling expenses are updated by shifting all elements of the window in accordance with a queue system, i.e. when a new element is entered the oldest one is discarded.

AL triggered by time In *AL triggered by time*, the algorithm queries at a given time interval. The sampling rate is calculated based on the labelling budget B, as can be seen in algorithm 2. A counter, c, is updated for every new data instance that arrives to keep track of when it is time to query. This strategy can be useful for instance as an option if another strategy is not using up the allowed budget or in a cold start scenario. In the latter case, the classifier has no access to any labelled data in the beginning, which means that *AL triggered by uncertainty* might not work, depending on how the uncertainty measurement is defined. Another scenario where this strategy would be useful is if concept drift is present [6]. Concept drift means that the underlying statistical properties of the data is changing over time. *AL triggered by uncertainty* might be inadequate in this case, since the measurement of uncertainty is based on older data.

Algorithm 2 AL triggered by time

Input: data stream $X = \{x_0, x_1, ...\}$, labelling budget B, classifier Ψ
$c \leftarrow 1$
$i \leftarrow 0$
repeat
  receive next data sample x_i
  $\hat{y}_i \leftarrow \Psi(x_i)$
  if $\frac{1}{B} \leq c$ **then**
    ask user about label y_i for x_i
    $\Psi \leftarrow$ incremental learning procedure$((x_i, y_i))$
    $c \leftarrow c - \frac{1}{B}$
  end if
  $c \leftarrow c + 1$
  $i \leftarrow i + 1$
until end of datastream

AL triggered at random Similar to *AL triggered by time*, *AL triggered at random* does not take the data instance itself or the classifier into account when deciding whether or not to query. Unlike *AL triggered by time* however, the queries are not made with a set time interval, instead the time for querying is randomly chosen, as algorithm 3 shows. First, a variable ζ is randomly generated from the uniform distribution $[0, 1]$. The value of ζ is then compared to the value of the labelling budget B, to decide whether to query or not. This type of strategy is often used as a baseline when evaluating performance of other interactive learning strategies [10, 13].

Algorithm 3 AL triggered at random

Input: data stream $X = \{x_0, x_1, ...\}$, labelling budget B, classifier Ψ
$i \leftarrow 0$
repeat
  receive next data sample x_i
  $\hat{y}_i \leftarrow \Psi(x_i)$
  generate a random variable $\zeta \in [0, 1]$
  if $\zeta \leq B$ **then**
    ask user about label y_i for x_i
    $\Psi \leftarrow$ incremental learning procedure((x_i, y_i))
  end if
  $i \leftarrow i + 1$
until end of datastream

MT triggered by error In *MT triggered by error*, the decision of whether or not to provide a label is based on the estimation from the classifier. The user has access to the classifier's estimation of the current state. Whenever the estimation made by the classifier is incorrect, the user provides the correct label, given that the labelling expenses $\hat{b}$ does not exceed the labelling budget B. The labelling expenses are calculated and updated the same way as in *AL triggered by uncertainty*. Algorithm 4 showcases the process. There are previous strategies proposed where the user provides labels based on the output from the classifier [3], but they are typically not presented in the context of single-pass streaming data.

Algorithm 4 MT triggered by error

Input: data stream $X = \{x_0, x_1, ...\}$, labelling budget B, classifier Ψ
$\hat{b} \leftarrow 0$
$i \leftarrow 0$
repeat
  receive next data sample x_i
  $\hat{y}_i \leftarrow \Psi(x_i)$
  if $\hat{y}_i \neq y_i$ **and** $\hat{b} < B$ **then**
    label y_i for x_i is provided by user
    $\Psi \leftarrow$ incremental learning procedure((x_i, y_i))
  end if
  update labelling expenses $\hat{b}$
  $i \leftarrow i + 1$
until end of datastream

MT triggered by state change In *MT triggered by state change*, the user will provide a label when the state y changes, given that the labelling expenses

$\hat{b}$ does not exceed the labelling budget B. The labelling expenses are calculated and updated the same way as in *AL triggered by uncertainty.* This strategy can be useful if there is a concern that all possible classes might not be properly represented in the labelled data used for training, e.g. if the dataset is unbalanced. Unbalanced datasets can result in a high performance overall, even though the performance related to the less frequent classes is poor. For instance, if *AL triggered by time* is used, the learner will have many labelled instances from the frequently occurring classes, while much less from the rare classes. A user could counteract this by providing labels based on the changing status of the state. As illustrated in algorithm 5, in *MT triggered by state change*, the user provides a label when the class of the current label changes. Depending on the scenario, the state might not change often, resulting in a low total number of labelled instances. However, as long as $\hat{b} < B$ holds, the user will provide a new label if the state changes. This means that while $\hat{b} < B$ and no new label is provided, the learner can assume the incoming instances have the same label as the last one provided by the user y_l and also add these to the collection of labelled data.

Algorithm 5 MT triggered by state change

```
Input: data stream X = {x_0, x_1, ...}, labelling budget B, classifier Ψ
b̂ ← 0
i ← 0
repeat
   receive next data sample x_i
   ŷ_i ← Ψ(x_i)
   if b̂ < B then
      if y_i ≠ y_{i−1} or i = 0 then
         label y_i for x_i is provided by user
         y_l ← y_i
         Ψ ← incremental learning procedure((x_i, y_i))
      else
         Ψ ← incremental learning procedure((x_i, y_l))
      end if
   end if
   update labelling expenses b̂
   i ← i + 1
until end of data stream
```

MT triggered by time The user can by their own initiative also provide labels with a given frequency. The algorithm for this strategy will almost be identical to the one for *AL triggered by time*, presented in algorithm 2. The only difference is that the user is not queried by an active learning strategy and instead keeps track of time themselves. Depending on the scenario, the user might be less exact compared to an active learning strategy that keeps track of time. In our setting however we assume that the user does always provide a label in accordance with the strategy, in which case it does not affect the implementation or the performance of the strategy. Therefore, the results from *AL triggered by time* are representing both *AL triggered by time* and *MT triggered by time*.

MT triggered by user factors The user can be triggered by factors related to themselves. These factors include for instance the user's internal state, e.g. their current stress level, and characteristics, e.g. the knowledge level on what they are supposed to provide labels for. A user might provide fewer labels if they are stressed compared to when they are not. If a user only wants to provide labels when they are certain of the label, a knowledgeable user could provide labels for more data instances compared to a less knowledgeable user. The decision function of when a user will provide a label will vary depending on which factor is triggering the user. Because of the broad spectrum of possible strategies and the additional information needed, this strategy was not included in the experiments.

Hybrid strategies The strategies presented above do not necessarily have to be employed separately, but can be merged in different types of combinations to create hybrid strategies. There can be several reasons for creating a hybrid strategy. For instance, if an *MT triggered by error* does not use up it's budget, it can be combined with *AL triggered by time* which will guarantee an influx of new labelled instances. Another reason can be to benefit from the strength of two separate strategies. For example, Žliobaitė et al. present the Split strategy as a combination of *AL triggered by uncertainty* and *AL triggered at random* which can be useful especially if concept drift is occurring [20].

4 Experimental setup

To illustrate how the different types of interactive machine learning strategies compare, we evaluated implementations of them in experiments on a benchmark dataset and a synthetically generated dataset with the user interaction simulated.[1] In all of the following experiments the learner started with no labelled data and had to incrementally learn as the labelled data was gradually accumulated over time. The evaluation was done in a test-then-train fashion. First, each data instance is used for testing, then the instance is added to the training data of the machine learning algorithm if a label is provided. The results show an average of 20 separate runs.

4.1 Datasets

The mHealth dataset The mHealth dataset contains recordings of activities within a health application [1, 2]. The total length of the recordings are 98304-161280 data instances, but the unlabelled data was excluded, resulting in data

[1] See the following link for synthetic dataset and code: https://github.com/ategen

recordings of 32205-35532 instances. Wearable sensors, accelerometer, gyroscope, magnetometer and electrocardiogram sensor, were used for the recordings. Ten different subjects each perform a sequence of 12 different physical exercises in one recording instance. The exercises are performed one at a time, which means that when the interval of one type of exercise is over, it is not repeated in the same recording. In the experiments, the recordings were concatenated to create one longer sequence and the different physical exercises were repeated 10 times. The order of the recordings was randomly generated for each run.

The synthetic dataset The synthetic dataset contains 5 classes where the mean values for each class were randomly generated in a 2D space. Ten thousand samples were then drawn for each class from a normal distribution with the given mean value and standard deviation. The total of 50000 generated samples constitutes the synthetic dataset. In all experiments, the values are the same but the order of the values in the data stream is shuffled for each run. How long the interval is for each class is also generated by sampling from a normal distribution. Fig. 2 displays a visualisation of the distributions of the classes of the dataset.

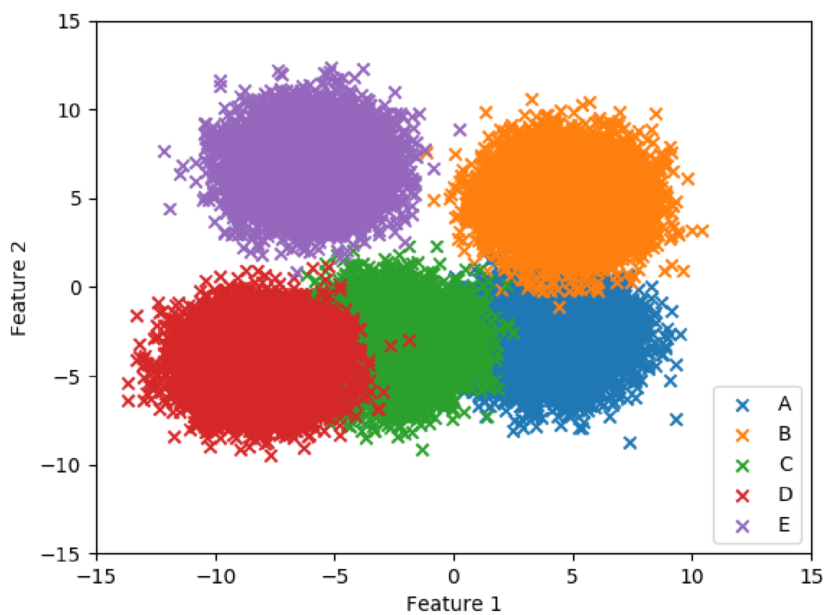


Fig. 2: A visualisation of the distributions of the classes of the synthetic dataset.

4.2 Machine learning algorithms

To compare the interactive learning strategies they were tested in combination with three different machine learning algorithms, Support Vector Machine with a polynomial kernel (SVM), k-Nearest Neighbor with $k = 3$ (k-NN) and Naïve Bayes classifier with assumption of Gaussian distributions of the features. The machine learning algorithms included were chosen because they are suitable for online learning, real-time classification and frequently employed in similar settings [7, 9, 12].

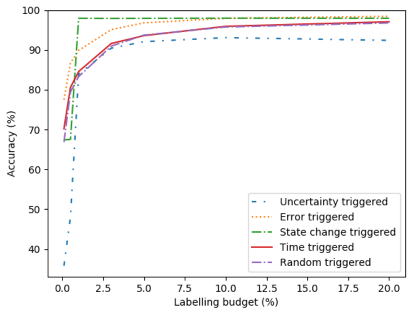
5 Results and Discussion

Figs. 3a, 3c and 3e display the accumulated accuracy given the labelling budget for the mHealth dataset for the Naïve Bayes classifier, SVM and k-NN, respectively. In the Figs. 3b, 3d and 3f the accumulated accuracy over the amount of samples estimated, with a labelling budget of 3%. Figs. 4a, 4c and 4e show the accumulated accuracy given the labelling budget for the synthetic dataset for the Naïve Bayes classifier, SVM and k-NN respectively. Similarly as for the mHealth dataset, Figs. 4b, 4d and 4f display the accumulated accuracy over the amount of estimated samples, with a labelling budget of 0.5%.

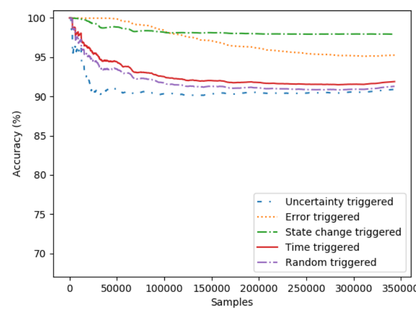
The left column of Fig. 3 (panels 3a, 3c and 3e) shows the final accumulated accuracy given the labelling budget. The figures show that when starting from a very low labelling budget, the performance substantially improves by only increasing the budget slightly. After a while however, there is only a small increase in performance, if any, even though the budget continues to increase. In the right column of the figure (panels 3b, 3d and 3f) the accumulated accuracy over number of samples can be found for a labelling budget of 3%. The best performing strategies, especially at the start, are *MT triggered by error* and *MT triggered by state change*.

In Fig. 4, the results from the experiments on the synthetic dataset are presented. The left column of Fig. 4 (panels 4a, 4c and 4e) displays the accumulated accuracy over labelling budget. Compared to Fig. 3, it does not display the final accumulated accuracy, but the accumulated accuracy after 5000 samples, or a tenth of the total amount of samples, has been processed. Since this dataset does not contain any concept drift, the performance of the different strategies will approach each other as the number of samples increases. After a while, the amount of labelled data is enough regardless of strategy to perform at a similar level. Consequently, it is interesting to analyze when the amount of samples is still relatively low. As the experiments where done using a cold start setup, the performance at the start of the data stream gives an indication of how the strategies perform with a limited amount of labelled data. Also here, the overall best performing strategies are *MT triggered by error* and *MT triggered by state change*.

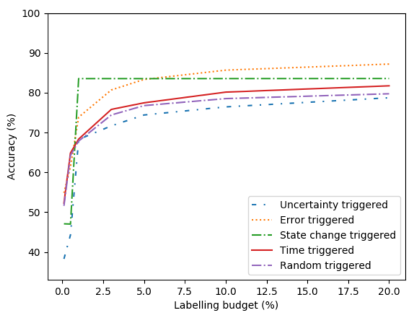
It is the two machine teaching strategies *MT triggered by error* and *MT triggered by state change* that generally perform best in the experiments. The



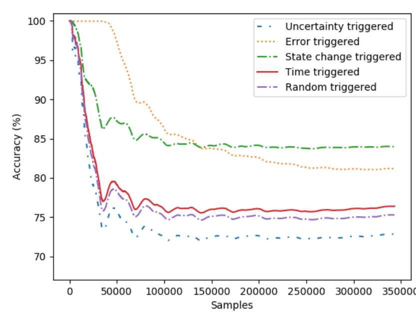
(a) NB, accuracy vs labelling budget.



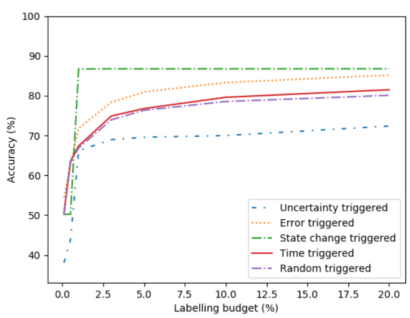
(b) NB, accuracy vs samples.



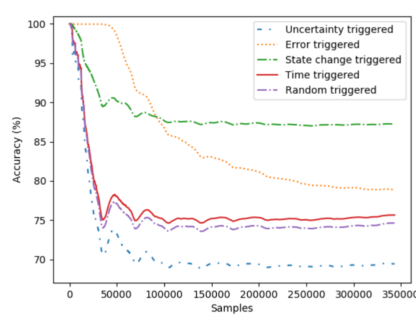
(c) SVM, accuracy vs labelling budget.



(d) SVM, accuracy vs samples.

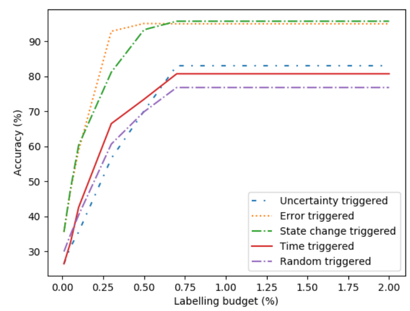


(e) kNN, accuracy vs labelling budget.

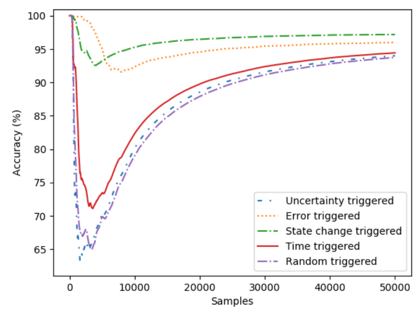


(f) kNN, accuracy vs samples.

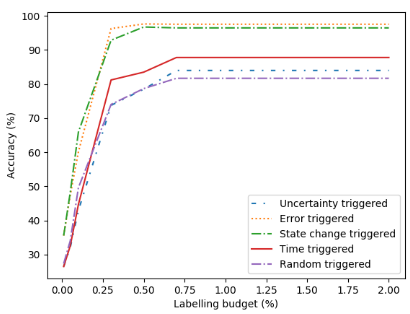
Fig. 3: The results from the experiments on the mHealth dataset. The left column (a, c and e) displays the final accumulated accuracy over labelling budget for Naïve Bayes (NB), Support Vector Machine (SVM) and k-Nearest Neighbor (kNN) respectively. The right column (b, d and f) displays accumulated accuracy over number of samples for a labelling budget of 3% for Naïve Bayes (NB), Support Vector Machine (SVM) and k-Nearest Neighbor (kNN) respectively.



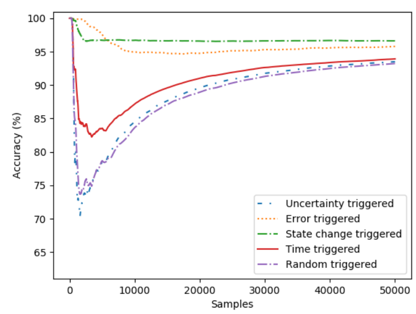
(a) NB, accuracy vs labelling budget.



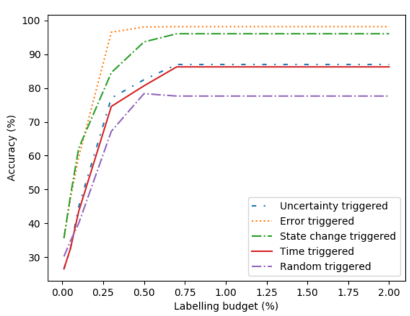
(b) NB, accuracy vs samples.



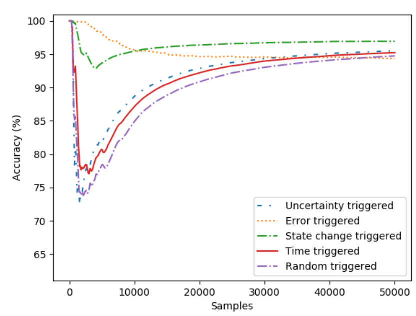
(c) SVM, accuracy vs labelling budget.



(d) SVM, accuracy vs samples.



(e) kNN, accuracy vs labelling budget.



(f) kNN, accuracy vs samples.

Fig. 4: The results from the experiments on the synthetic dataset. The left column (a, c and e) displays accumulated accuracy after 5000 samples over labelling budget for Naïve Bayes (NB), Support Vector Machine (SVM) and k-Nearest Neighbor (kNN) respectively. The right column (b, d and f) displays accumulated accuracy over number of samples for a labelling budget of 0.5% for Naïve Bayes (NB), Support Vector Machine (SVM) and k-Nearest Neighbor (kNN) respectively.

results are in line with our previous work [16, 17] and give an indication that letting the user be more proactive can be beneficial on performance. This is noteworthy since most interactive online machine learning strategies employed in the literature fall under the active learning subcategory. Online machine teaching is an area that currently deserves further exploration.

The reason for the high performance at the start in the right-hand side column of Figs. 3 and 4, contrasting the accuracy over samples, is the cold start scenario. The figures are displaying the accumulated accuracy for each new incoming sample. For the very first estimation, there will only be one labelled data instance collected. As the learner only has been introduced to one label at this point in time, it will continue to estimate this label until another option is presented. In many types of data streams it is probable that data instances with the same label follow consecutively for a period of time (e.g. data streams collected in an activity recognition setting). At the very beginning, for a short period of time, this leads to an accumulated accuracy of 100%. When other labels are introduced to the learner, the performance goes down. As more labelled data instances are gathered for each class, the performance reaches a level that is more representative of the learner in the long run.

We chose to include experiments on both a dataset consisting of actual recordings, the mHealth dataset, and a synthetically generated dataset. The mHealth dataset gives an example of how the strategies perform in a real-life setting, where for instance data can be noisy and there might be concept drift. The synthetic dataset on the other hand, is not meant to represent a realistic scenario for data collection, but it provides a framework for comparing the different strategies. While we have included experiments on two different datasets, further experiments on a variety of datasets would be useful to get an exhaustive comparison of the strategies.

The high performance of *MT triggered by state change* is because it can collect a larger amount of labelled data to use as training data, while not having to involve the user all the time. The strategy expects the user to provide a label when the state is changing and if the labelling expenses still are below the allowed labelling budget, the learner assumes that the state is the same until the user provides a new label. The learner can thus continue to collect labelled data instances, without any effort from the user, as long as the labelling expenses are below the labelling budget. The current labelling expenses are calculated based on how many times the user has provided labelled data. The labelling expenses can thus be kept low while the learner continues to gather labelled data

While the overall performance is high for *MT triggered by state change*, it relies on the assumptions that the user will always provide a correct label in accordance with the strategy as long as there is labelling budget. These assumptions are found in most work on interactive machine learning but whether they are realistic is rarely discussed. While there exists scenarios where these are reasonable assumptions (e.g. a medical doctor labelling data within their expertise), in many scenarios the assumptions are simplifications of the actual setting. An interactive machine learning strategy that performs well given the assumptions on the user might not necessarily perform as well if the user did not always provide a label or sometimes provided an incorrect label. We plan to explore how relaxing these assumptions on the user can affect performance in future work. Even though we kept the assumptions in the experiments, the taxonomy also covers scenarios were they are relaxed.

We assume in the experiments that there is only one user, or possibly multiple users, but with the same behavior, but there exists several settings with multiple users that all have individual characteristics. For instance in crowdsourcing, there are multiple users with the possibility to provide labels, but they might not always be available or willing to provide a label [15]. When they do provide a label, it might be incorrect. We also assume that only one label can be provided for each instance. In many scenarios where the classes are disjoint this is a reasonable assumption (e.g. is the room empty or not), but in other scenarios there might be a value in allowing multiple labels for one instance.

MT triggered by user factors was not included in the experiments because of the lack of necessary data. These types of factors are highly dependent on the user and strategies based upon them are difficult to model. The pattern of when the user is providing labels might not be known to an outside observer or even to the user themselves. Nevertheless, in many real-life scenarios they do have an impact on the result. For instance, a user might be more probable to provide a label when he or she is in a good mood and less likely to provide labels when in a bad mood or when they are stressed. Our aim is to explore these factors further in future work.

6 Conclusion and Future Work

In this work we have presented a taxonomy of interactive online machine learning strategies. We have also done implementations of the strategies presented in the taxonomy and performed experiments on one benchmark dataset and one synthetically generated dataset. The experiments show that the strategies where the user is triggered to provide labels when the state is changing or when an estimation is incorrect are overall better performing than when an active learning strategy queries the user. The difference in performance is especially noticeable when there is a lower amount of labelled data. The results gives an indication that giving the user a more proactive role in labelling, unlike typical active learning, can be beneficial on performance.

In future work we plan to further validate our conclusions and further explore the taxonomy through experiments. Our aim is to test the taxonomy on a variety of datasets and other machine learning algorithms. We also aim to explore machine teaching that is triggered by user related factors further. We plan to test the robustness of interactive online machine learning strategies if the user does not always respond with a correct label.

References

1. Banos, O., Garcia, R., Holgado-Terriza, J.A., Damas, M., Pomares, H., Rojas, I.,Saez, A., Villalonga, C.: mhealthdroid: a novel framework for agile development of mobile health applications. In: International workshop on ambient assisted living.pp. 91–98. Springer (2014)
2. Banos, O., Villalonga, C., Garcia, R., Saez, A., Damas, M., Holgado-Terriza, J.A.,Lee, S., Pomares, H., Rojas, I.: Design, implementation and validation of a novel open framework for agile development of mobile health applications. Biomedical engineering online **14**(2), S6 (2015)
3. Chen, Y., Singla, A., Mac Aodha, O., Perona, P., Yue, Y.: Understanding the role of adaptivity in machine teaching: The case of version space learners. In: Advances in Neural Information Processing Systems. pp. 1476–1486 (2018)
4. Cohn, D., Atlas, L., Ladner, R.: Improving generalization with active learning. Machine learning **15**(2), 201–221 (1994)
5. Fu, Y., Zhu, X., Li, B.: A survey on instance selection for active learning. Knowledge and information systems **35**(2), 249–283 (2013)
6. Gama, J.,Žliobaitėe, I., Bifet, A., Pechenizkiy, M., Bouchachia, A.: A survey on concept drift adaptation. ACM computing surveys (CSUR) **46**(4), 1–37 (2014)
7. Khan, Z.A., Samad, A.: A study of machine learning in wireless sensor network.Int. J. Comput. Netw. Appl4, 105–112 (2017)
8. Kottke, D., Krempl, G., Spiliopoulou, M.: Probabilistic active learning in datas-treams. In: International Symposium on Intelligent Data Analysis. pp. 145–157.Springer (2015)
9. Krawczyk, B.: Active and adaptive ensemble learning for online activity recognitionfrom data streams. Knowledge-Based Systems 138, 69–78 (2017)
10. Lughofer, E.: Single-pass active learning with conflict and ignorance. Evolving Systems **3**(4), 251–271 (2012)
11. Lughofer, E.: On-line active learning: a new paradigm to improve practical useability of data stream modeling methods. Information Sciences 415, 356–376 (2017)
12. Mahdavinejad, M.S., Rezvan, M., Barekatain, M., Adibi, P., Barnaghi, P., Sheth,A.P.: Machine learning for internet of things data analysis: A survey. Digital Communications and Networks **4**(3), 161–175 (2018)
13. Miu, T., Missier, P., Plötz, T.: Bootstrapping personalised human activity recognition models using online active learning. In: 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing. pp. 1138–1147. IEEE (2015)
14. Settles, B.: Active learning literature survey. Tech. rep., University of Wisconsin-Madison Department of Computer Sciences (2009)
15. Shickel, B., Rashidi, P.: Art: an availability-aware active learning framework for data streams. In: The Twenty-Ninth International Flairs Conference (2016)
16. Tegen, A., Davidsson, P., Mihailescu, R.C., Persson, J.A.: Collaborative sensing with interactive learning using dynamic intelligent virtual sensors. Sensors **19**(3),477 (2019)
17. Tegen, A., Davidsson, P., Persson, J.A.: Activity recognition through interactivemachine learning in a dynamic sensor setting. Personal and Ubiquitous Computing(2020). https://doi.org/10.1007/s00779-020-01414-2
18. Zhu, X.: Machine teaching: An inverse problem to machine learning and an approach toward optimal education. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)

19. Zhu, X., Singla, A., Zilles, S., Rafferty, A.N.: An overview of machine teaching. arXiv preprint arXiv:1801.05927 (2018)
20. Žliobaitė, I., Bifet, A., Pfahringer, B., Holmes, G.: Active learning with drifting streaming data. IEEE transactions on neural networks and learning systems **25**(1),27–39 (2013)