The Effects of Reluctant and Fallible Users in Interactive Online Machine Learning

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 it is important to select the most informative data instances to label in order to minimize the effort of the human user. There are basically two categories of interactive machine learning. In the first category, active learning, it is the computational learner that selects which data to be labelled by the human user, whereas in the second one, machine teaching, the selection is done by the human teacher. It is often assumed that the human user is a perfect oracle, i.e., a label will always be provided in accordance with the interactive learning strategy and that this label will always be correct. In real-world scenarios however, these assumptions typically do not hold. In this work, we investigate how the reliability of the user providing labels affects the performance of online machine learning. Specifically, we study reluctance, i.e., to what extent the user does not provide labels in accordance with the strategy, and fallibility, i.e., to what extent the provided labels are incorrect. We show results of experiments on a benchmark dataset as well as a synthetically created dataset. By varying the degree of reluctance and fallibility of the user, the robustness of the different interactive learning strategies and machine learning algorithms is explored. The experiments show that there is a varying robustness of the strategies and algorithms. Moreover, certain machine learning algorithms are more robust towards reluctance compared to fallibility, while the opposite is true for others.

1 Introduction

Active learning [13] and machine teaching [17,18] are two different categories of interactive machine learning strategies that can be used to decrease the amount of labelled data needed to train a machine learning algorithm, while still preserving a high performance. Labelling data is often costly and demands a lot of work from a human user that is meant to provide the labels. In interactive machine learning, a smaller selection of the instances are instead chosen for labelling, where the size of the selection is decided by a labelling budget. The aim of the interactive learning strategy is to pick the instances that will provide most information to the machine learning algorithm.

Interactive online learning is a special case of interactive learning, where the data arrives in a single-pass streaming fashion and each data instance can only

be processed by the interactive learning strategy when it arrives. Thus, a decision has to be made at each point in time whether a label should be provided or not for the current data instance.

Generally in interactive learning, the assumptions of the user are that they will always provide a label when queried by an active learning strategy or in accordance with an machine teaching strategy. Furthermore, it is typically assumed that the label provided by the user always is correct. In some settings this can be reasonable assumptions, for instance a medical doctor labelling patient data within their expertise, but in many scenarios they do not hold. While the intent of the assumptions might be to create simplifications in experiments based on complex real-world settings, they may result in conclusions that are not valid. For instance, in an idealised setting, where the user always responds with a correct label, one approach might give the best performance compared to an alternative approach. This does not necessarily mean that the approach will still be the highest performing one if not all labels are correct. If the idealised setting is a simplification of the real setting and the user sometimes does provide an incorrect label, the alternative approach might be the better choice.

In this work we explore how the reliability of a user providing labels affects the performance of online machine learning. We look at the aspects of reluctance, how probable is it that a user will not provide a label in accordance with a given interactive learning strategy, and fallibility, how probable is it that the label provided by a user is incorrect. By varying the degree of reluctance and fallibility of the user we study how this influence the performance of different interactive online machine learning strategies.

2 Related work

In most work on interactive machine learning, the assumptions are that the user will always provide a correct label when queried by an active learning strategy or in accordance with a machine teaching strategy. Furthermore, when the assumptions are made, it is rarely discussed whether they are realistic for the given scenario. In previous work that explore settings where the assumptions are relaxed, this type of user is often referred to as an imperfect oracle. The term is in contrast to the standard definition of oracle in active learning, which is always assumed to respond to a query with a correct label.

Yan et al. explores an active learning setting where the user might return incorrect labels, but might also abstain from labelling [15]. The results show that learning with a user that might abstain is easier than a user that might provide incorrect labels, as an abstention response never mislead the learning algorithm, unlike incorrect labels. However, in this setting the learner can request the label of any data point in the instance space. In our setting the instances are presented in a single-pass streaming manner.

Bouguelia et al. introduces an active learning strategy that handles incorrectly labelled instances, without relying on crowdsourcing [3]. Experiments compare the strategy to multiple benchmark strategies and showcase that the

proposed strategy achieves better performance than several of them. The experimental setup is not single-pass and does not address the cold start problem.

The effect of feature noise in an active learning setting is studied by Ramdas et al. [12]. They conclude that active learning results in better performance compared to passive learning even with the presence of feature noise.

Miu et al. present an online active learning framework [10]. Annotations provided by a user is collected in real-time and used for Human Activity Recognition tasks. Apart from testing the proposed framework on benchmark datasets, it was also tested in user studies, by implementing it in a mobile app through which participants could provide labels. In the user study, the replies from the user were simulated to be incorrect 10 % of the time. Apart from baselines, only one interactive learning strategy and one machine learning algorithm was used in the experiments.

Shickel et al. also introduces a framework for active learning in an online setting with multiple imperfect oracles [14]. The framework can query multiple different oracles, based on when they are available, which is useful for instance when generating data from crowdsourcing. While different active learning strategies might work for the framework, the only strategy used in the work is active learning triggered by uncertainty.

The effects of an imperfect oracle are explored by Donmez et al., both with regard to not always being correct and to not always being avaliable [4]. To obtain labels, an active learning strategy based on uncertainty is used in the experiments. Unlike to the typical active learning setting, the oracle can be both fallible (i.e. provide incorrect labels) and reluctant (i.e. might not always respond when queried). The scenario discussed in the work is for batch learning however, and not streaming data.

Zeni et al. perform experiments where students are asked to provide information regarding their behaviour via a mobile application [16]. The information provided from the students is compared to information gathered from the phone, e.g. location, to test the correctness. The experiments show that the students do sometimes provide incorrect labels and that there was a variation among the individuals in the amount of incorrect labels provided.

Machine teaching where the user providing labels can have varying degree of reliability is an area that needs further investigation, as well as for single-pass streaming data in cold-start settings. In our work we take a step towards exploring how the reliability of a user affects performance of interactive online machine learning, including both machine teaching and active learning strategies.

3 Experimental setup

The aim of the experiments is to explore how varying reliability of a user providing labels affects performance of different online machine learning methods and different interactive learning strategies. The online learning setting means that the data arrives in a streaming fashion, where each instance is presented and processed once by the learning algorithm. This can be compared to pool-based

settings where all unlabelled data instances typically are available at any given time.

In the experiments we have a cold start scenario, which means that there is no labelled data for the machine learning to train on at the start of the data stream. Labelled data has to be collected gradually over time to incrementally train the machine learning model.

Streaming data means that the total amount of data is not necessarily known and might even be infinite. This creates issues that has to be taken into account for the interactive learning strategies and the machine learning algorithms. For instance, the labelled data that is gradually collected cannot be stored indefinitely, as the amount of labelled data theoretically could approach infinity. To counteract this, there is a limit of how many labelled data instances are stored for each class. If the maximum limit is reached for a given class and a new instance with the same label arrives, the oldest one from the collection is discarded.

The evaluation was done in a test-then-train fashion [5], i.e. where the model first attempts to estimate the incoming data instance and then, if a label is provided, uses the new labelled instance to incrementally train the model. The result thus becomes an accumulative accuracy that showcases performance of the model over time. The results displayed in the next section are all average values of several runs of each experiments. How a run of the experiments was constructed is described in more detail in section 3.4.

3.1 Machine Learning algorithms

Three machine learning algorithms were implemented in the experiments, Naïve Bayes classifier, Support Vector Machine (SVM) and k-Nearest Neighbor (k-NN). The aim was to study if there is a discernible difference in the effect of a fallible or reluctant user. The algorithms were chosen to be well-known off-the-shelf machine learning algorithms but also suitable for the setting at hand, i.e. online learning with a cold start scenario. Even though the experiments presented here are from simulations done on previously recorded or created datasets, the intention is that the experiment should be able to run in real-time, which means that the complexity of the machine learning algorithm also has to be considered.

Gaussian Naïve Bayes classifier was included in the experiments because it works well for online learning and has low computational complexity [7]. It also needs a relatively small amount of training data before it can start to produce estimations, which makes it suitable for a cold start scenario.

SVMs are effective in dealing with high-dimensional data, which is of importance in many settings with streaming data, and have efficient memory usage [9]. SMVs aim to find hyperplanes that divides the classes with a margin that is maximized. As the number of fallible instances increases this will become increasingly difficult and the method will spend more time trying to optimize the classifier. To counteract that training time becomes a concern, a maximum number of iterations to optimize the hyperplanes is set to 1000. A polynomial kernel is used in the experiments, based on initial testing. These results are not included in this work due to space restrictions.

k-NN is a fitting classifier for our scenario because of its simplicity, it is suitable for online learning and the computational work needed can be limited through the amount of data temporarily stored [6]. The method looks at the labels of the k nearest, previously collected, data instances to classify a new data instance. This means that only k instances needs to be collected before the method can be used, which is good for a cold start scenario. The value of k was set to 3, based on initial experiments that are not included in this work due to space restrictions.

3.2 Interactive Learning strategies

Two different interactive learning strategies were used in the experiments, one active learning strategy, where the learning model is deciding when to query the user for a label, and one machine teaching strategy, where the user decides which instances to provide labels for. All type of interactive learning has to accommodate for a labelling budget. The labelling budget is set beforehand and decides how big portion of the total amount of data can be labelled by the user. In a pool-based setting, the use of the labelling budget is straightforward. In contrast, in a setting with streaming data, only one data instance is processed at a time and the choice of whether or not to ask for a label has to be done as soon as it appears in the data stream. Since the total number of data instances is unknown at the start of processing the streaming data, the labelling budget can not be calculated the same way as it is done in a pool-based setting. Instead, a sliding window containing information on which of the latest processed instances the user has provided a label for is used to calculate the current labelling expenses. The labelling expenses are compared to the labelling budget, to determine if it is currently possible to query for more labels. The window size is set to 200 instances in the experiments.

Active learning triggered by uncertainty Along with each estimation produced by the machine learning algorithm is also a measurement of how certain the model is of its own estimation. The active learning strategy compares the produced uncertainty measurement to a set threshold. If the measurement is below the threshold, the estimation is considered uncertain and the user is queried, given that there is enough labelling budget. This is the most common type of active learning used and is sometimes referred to as Uncertainty sampling [13] or Uncertainty-based sampling methods [8]. To implement this strategy, the measurement of uncertainty needs to be defined and this is dependent on the machine learning algorithm used. As three different machine learning algorithms are employed in our experiments, each one needs their own uncertainty measurement.

The Naïve Bayes classifier produces a probability for each class and then picks the class with the highest probability for its estimation. The probability of the chosen class is then compared to a threshold. An initial value is set for the threshold, but the value can be lowered or increased over time, depending on whether a query is made or not. If many queries are made the threshold is

gradually lowered, as it might indicate that the threshold is too high, or the opposite if few queries are made. The implementation is based on the Variable Uncertainty Strategy presented by Žliobaitė et al. [19].

The measurement of uncertainty used for SVM is the distance from the new data instance to the hyperplanes. If the distance is short it means that the new instance is in close proximity to another class and thus might have a higher probability belonging to the other class compared to an instance further away. The distance is compared to a threshold that, like in the case of Naïve Bayes can be altered depending on how many queries are made.

For the k-NN the measurement is based on how many of the k instances nearest to the new instance, i.e. the instances deciding which class to estimate, have the same label. If more than two-thirds of the instances have the same label the estimation is considered certain otherwise not. The strategies used for SVM and k-NN are further described by Pohl et al. [11].

Machine teaching triggered by error In machine teaching it is the user that employs a strategy of when to provide labels to the learning algorithm. In the strategy included in the experiments, the user is aware of the current estimation produced by the model and whenever this estimation is incorrect the user is triggered to provide a label, given that there is enough labelling budget to do so. In this way the user can aid the model by correcting it when it makes a mistake.

3.3 Simulation of reliability in the user

In the experiments the aim was to explore how a varying degree of reliability in the user providing labels affects performance. The two aspects of reliability studied in the experiments were reluctance and fallibility, both are further explained below. By simulating reluctance and fallibility in the user, the level of reliability could be controlled in each experiment.

Reluctance A reluctant user is a user that does not always provide a label in accordance with the given learning strategy. In the case of the active learning strategy, the user will not always reply to a query and in the case of the machine teaching strategy, the user will sometimes not provide a label, even though the estimation is incorrect and there is enough labelling budget. In a real-world scenario the reluctant behavior can be explained by a user that i.e. is distracted by something else, unwilling to provide labels or uncertain of which label to provide. The level of reluctance is varied between 0% and 50% in the experiments, during which fallibility is kept at 0%. The level of reluctance informs how big portion of the queries posed that the user will not respond to. For each new query posed, a random number between 0 and 1 is generated and if the generated number is lower than the level of reluctance, the user will not reply.

Since the window used to calculate the current labelling expenses only includes the cases when a label has been received, the expenses are not increased when a query does not get a reply or when a user does not provide a label.

Theoretically this could mean that for the very next instance the active learning strategy could pose a new query or the user could provide a label in the case of machine teaching. This would not be very realistic however, as a user that for instance is distracted at one point in time will likely still be distracted the moment afterwards. Instead, if a label that should have been provided was not due to reluctance, the algorithm had to wait the length of the labelling window before another data instance could be considered for labelling.

Fallibility A fallible user does always provide a label when queried or triggered, but the label is not always correct. This could for example correspond to a user that does not know or is uncertain about the correct label or that makes a mistake. The experiments are constructed in a similar way to the experiments explained above for a reluctant user. The level of fallibility decides the probability of a label being incorrect. In the experiments the level is varied between 0% and 50%, while the level of reluctance was kept at 0%. When an incorrect label is provided, the false label to be attached to the data instance is chosen randomly from all the incorrect labels. As one of the datasets employed in the experiments contains real-world recordings (the mHealth dataset, described further below) there is a risk concerning the correctness of the labels provided. In the experiments however, the assumption is made that the labels in the dataset are correct.

3.4 Datasets

To study the effects on performance of reluctance and fallibility in the user, experiments were performed on two separate datasets. The first is an activity recognition dataset and consists of recordings from a real-world scenario. The second is a synthetically constructed dataset.

mHealth dataset The mHealth dataset consists of recordings of 10 subjects with wearable sensors performing a specific routine of physical exercises [1, 2]. The set of wearable sensors include gyroscope magnetometer, accelerometer and electrocardiogram sensor. Each recording contains between 98304 to 161280 data instances from one subject, resulting in 10 recordings in total. The data contained unlabelled data which was excluded for the experiments, resulting in recordings of a length 32205-35532 instances. The specific routine consists of 12 different physical exercises that the subject is meant to perform in a specific sequence. The routine is constructed so that one exercise follows the other, but is never repeated. Because of the test-then-train evaluation, the different classes that are to be estimated, i.e. the psychical exercises in this dataset, should appear more than once to result in any proper conclusion with regards to performance evaluation. To create a sequence where all exercises appear more than once, the recordings are all put after one another to create one longer data sequence. The order of which the different recordings are placed is randomly generated for each run. The result produced is the average of 20 separate runs.

Synthetic dataset The synthetically generated dataset[1] contains 50000 data instances in total. The dataset has two features and five classes with 10000 instances belonging to each class. For each class, a mean value was created for the two features. The instances was then generated by sampling from a 2D normal distribution with the given mean value of the given class and a set standard deviation. In Fig. 1 a visualisation of the dataset is displayed. For each run of the experiments, an order of all the data instances had to be established. This was done by first randomly choosing one of the classes, then a random sample from a normal distribution was drawn to decide how many samples of this class should be in the interval. If for instance 20 instances of class A was set for one interval, 20 random data instances belonging to class A were chosen and put after one another in the sequence. One interval was put after another until all the data instances were arranged in the sequence. The ordering of the instances was redone for each run, but the data instances themselves were the same for all of them. The result presented is the mean value of 100 separate runs.

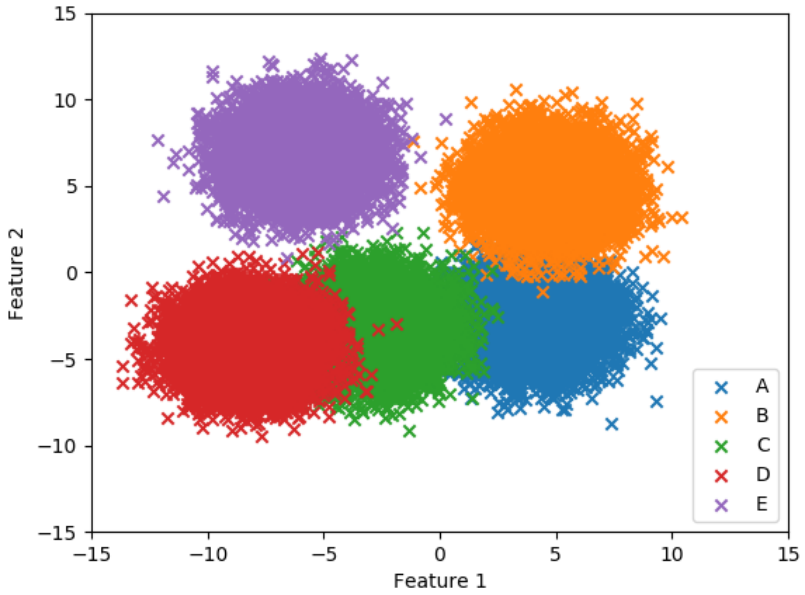


Fig. 1: A visualisation of the distributions of the classes of the synthetic dataset.

[1] The dataset can be found via the link: https://github.com/ategen/synthetic-dataset

3.5 Limitations

There are several aspects of the typical assumptions made in interactive learning that does not always hold in real-world scenarios. In this work we have chosen to focus on two aspects of reliability of the user, reluctance and fallibility, but there are other that could be of interest to study, depending on the application.

In the experiments it is assumed that there is one user, or multiple users with the same levels of reliability, providing labels. In certain settings however, there might be multiple users with different characteristics, all with the possibility to provide labels at least at some point in time. An example where this is highly relevant is in cases when crowdsourcing is used to collect labelled data.

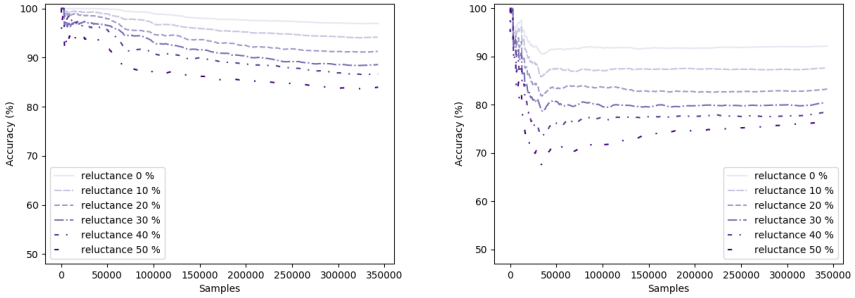
Another assumption made in the experiments that could be challenged is that the cost of providing a label is uniform for all possible labels. In some settings this might not be the case. For instance, if a scenario has classes that are similar, there might be data instances that need a more thorough examination to find out which class they belong to, while other instances can be classified by the user at a glance. If the cost of labelling varies, this could also connect to reluctance and fallibility of the user. A more difficult, or costly, label can lead to a higher probability that the user does not provide a label. There could also be a case where there is an option to have more thorough and costly labelling by the user, resulting in a lower fallibility, or a quicker and cheaper labelling, but with a higher risk of being fallible. Depending on the application setting, one approach might be preferable over the other.

One relevant issue when discussing learning from streaming data is concept drift, which means that the statistical properties of the streaming data changes over time. It is a phenomenon that is present in many streaming data settings and there exists works that discusses it in more detail [5]. Concept drift is not the main focus of this work, but is passively handled by continuously updating the machine learning model with new incoming data instances and discarding old ones. With a limited amount of data for training however, there is a risk of overfitting. When dealing with streaming data, this is an important trade-off to be aware of.

4 Results and Discussion

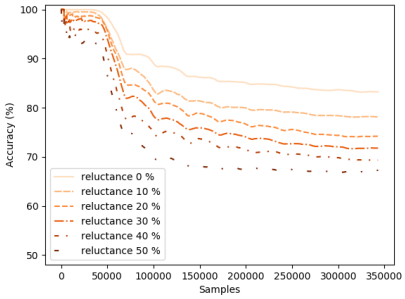
In Figs. 2 and 3 the results from experiments on the mHealth dataset when varying the degree of reluctance and fallibility of the user can be seen. Figs. 4 and 5 show results from the corresponding experiments on the synthetic dataset. The performance is displayed as accumulated accuracy over the number of samples that have been estimated so far. This means that when the number of samples is lower it represents performance early, i.e. when fewer estimations have been done and fewer labelled data instances have been gathered.

The results confirms that performance gets worse with an increased level of reluctance or fallibility of the user in all of the experiments. How big the decrease in performance is depends on the dataset, interactive learning strategy and machine learning method however.

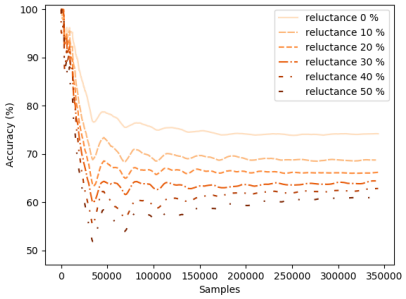


(a) Machine teaching, NB

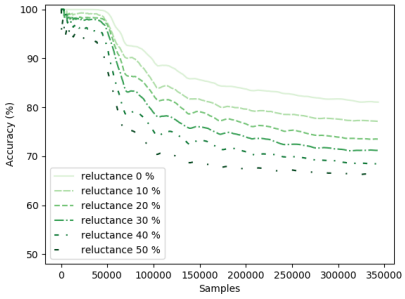
(b) Active learning, NB



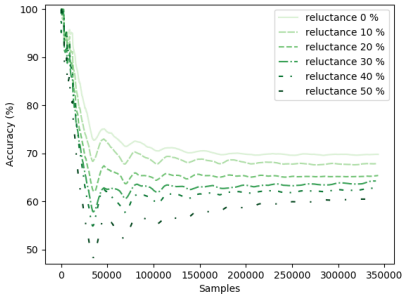
(c) Machine teaching, SVM



(d) Active learning, SVM

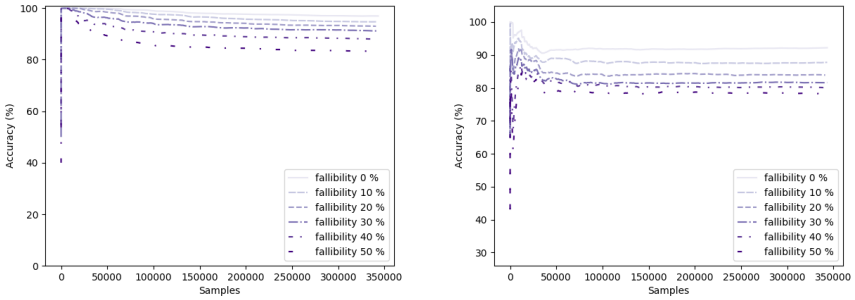


(e) Machine teaching, kNN



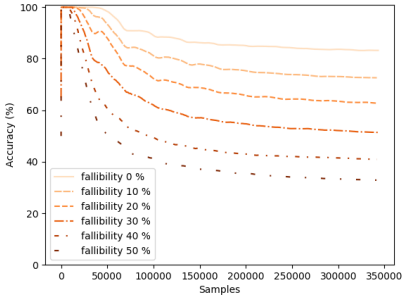
(f) Active learning, kNN

Fig. 2: The results from the experiments on the mHealth dataset when the level of reluctance is varied. The left column (a, c and e) displays the result for the machine teaching strategy triggered by uncertainty and the right column (b, d and f) for the active learning strategy triggered by error for Naïve Bayes (NB), Support Vector Machine (SVM) and k-Nearest Neighbor (kNN) respectively.

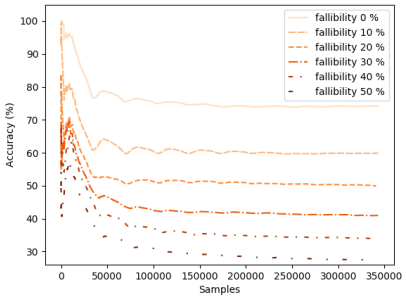


(a) Machine teaching, NB

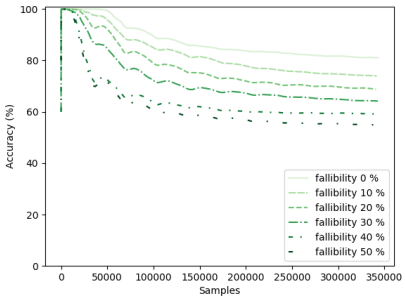
(b) Active learning, NB



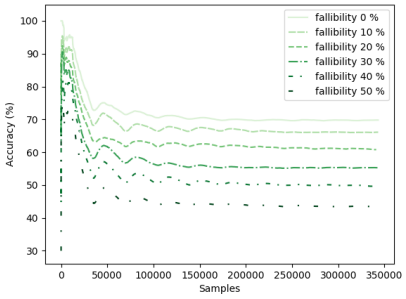
(c) Machine teaching, SVM



(d) Active learning, SVM

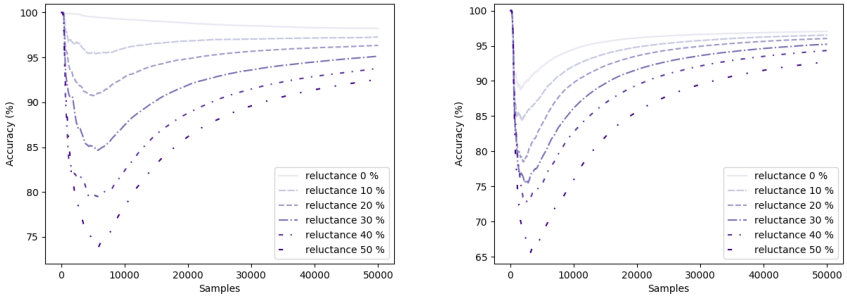


(e) Machine teaching, kNN



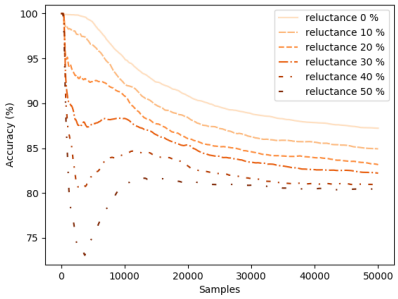
(f) Active learning, kNN

Fig. 3: The results from the experiments on the mHealth dataset when the level of fallibility is varied. The left column (a, c and e) displays the result for the machine teaching strategy triggered by uncertainty and the right column (b, d and f) for the active learning strategy triggered by error for Naïve Bayes (NB), Support Vector Machine (SVM) and k-Nearest Neighbor (kNN) respectively.

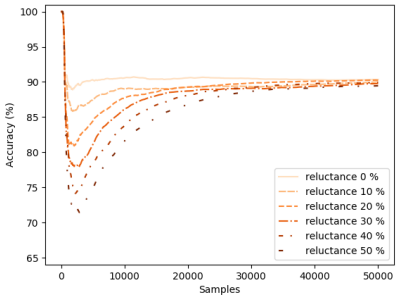


(a) Machine teaching, NB

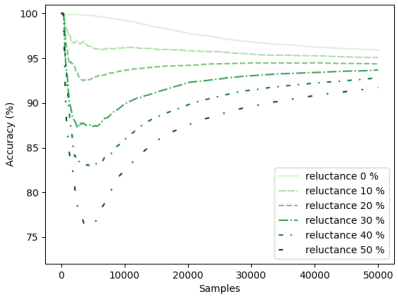
(b) Active learning, NB



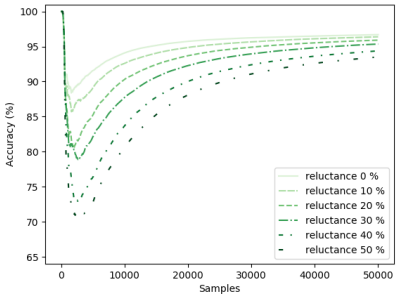
(c) Machine teaching, SVM



(d) Active learning, SVM

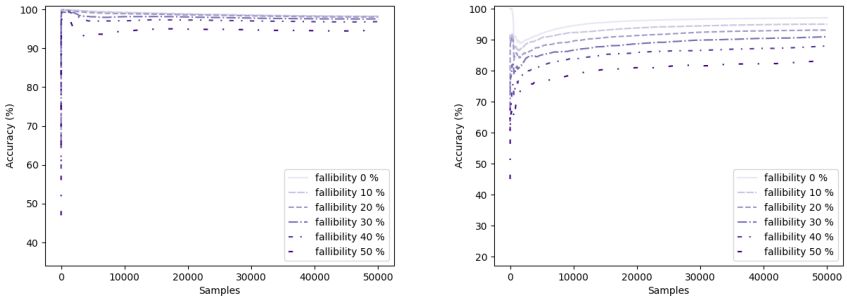


(e) Machine teaching, kNN



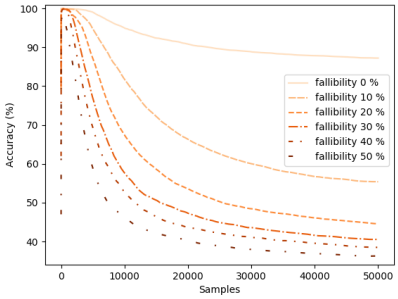
(f) Active learning, kNN

Fig. 4: The results from the experiments on the synthetic dataset when the level of reluctance is varied. The left column (a, c and e) displays the result for the machine teaching strategy triggered by uncertainty and the right column (b, d and f) for the active learning strategy triggered by error for Naïve Bayes (NB), Support Vector Machine (SVM) and k-Nearest Neighbor (kNN) respectively.

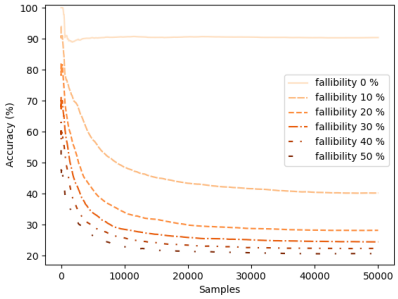


(a) Machine teaching, NB

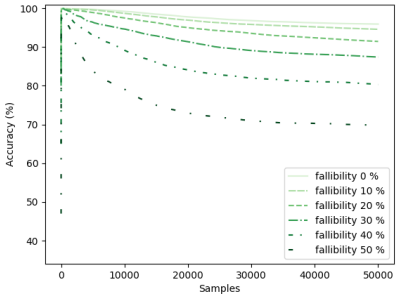
(b) Active learning, NB



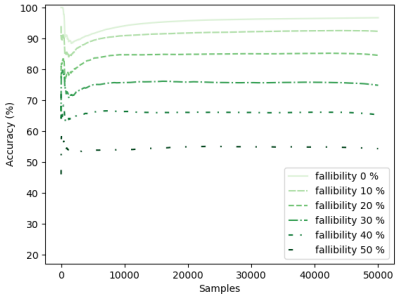
(c) Machine teaching, SVM



(d) Active learning, SVM



(e) Machine teaching, kNN



(f) Active learning, kNN

Fig. 5: The results from the experiments on the synthetic dataset when the level of fallibility is varied. The left column (a, c and e) displays the result for the machine teaching strategy triggered by uncertainty and the right column (b, d and f) for the active learning strategy triggered by error for Naïve Bayes (NB), Support Vector Machine (SVM) and k-Nearest Neighbor (kNN) respectively.

In the experiments on the mHealth dataset, Naïve Bayes classifier and the machine teaching strategy triggered by error has the highest performance. If the level of reluctance is increased (Fig. 2) or if the level of fallibility is increased (Fig. 3) the combination of Naïve Bayes and the machine teaching strategy is still giving the best performance. In Fig. 2a, showing the results for this combination with an increasing reluctance level however, the trend still seems to be downwards towards the end of all samples. This can be compared to Naïve Bayes with active learning triggered by uncertainty, Fig. 2b, where the performance overall is lower, but stabilizes after a while. In the experiments with a varying degree of fallibility, SVM and k-NN appear to be more affected compared to Naïve Bayes classifier. As the level of fallibility increases, the performance of SVM and k-NN drops faster, especially compared to Naïve Bayes in combination with the active learning strategy.

Figs. 4 and 5 contain the results from the experiments on the synthetic dataset. In the experiments where the reluctance of the user is varied, Fig. 4, the biggest difference in performance can be found early, when the number of samples is low. The main reason for this is likely that a higher level of reluctance in the user leads to a longer time before enough labelled data instances have been gathered to result in a performance on the same level as a user with 0% reluctance to provide labels. Towards the end of the average run, the performance of the different levels of reluctance approaches each other. When looking at the final accumulated accuracy of Fig. 4, Naïve Bayes classifier and k-NN performs better than SVM. The figure also shows that at the start the machine teaching strategy performs better than the active learning strategy. This is consistent over all machine learning algorithms tested, but the difference decreases as the level of reluctance increases. For SVM at the highest level of reluctance, there is no significant difference at the start between the active learning strategy and the machine teaching strategy. Furthermore, after a while the active learning strategy outperforms the machine teaching strategy.

In the results a drop in performance can be seen in several of the figures. One reason for this is the cold start scenario and the patterns in the data streamed. At the very start of the data stream, a labelled data instance from the first class is provided to the machine learning algorithm. In accordance with the nature of the data, for a period, the same class will continue and more labelled instances from this class can be collected. At this point in time the task of classifying is easier, or even trivial in the case of no fallibility. As more classes are introduced over time however, the difficulty of classification increases, which in turn can lead to a lower performance.

The experiments where the effects of fallibility of the user was tested on the synthetic dataset are displayed in Fig. 5. Here, the effect that the choice of interactive learning strategy and machine learning method can have on performance is clear. The best performing and most robust combination is Naïve Bayes classifier with the machine teaching strategy triggered by error. When the level of fallibility is at 0%, there is not a significant difference between the different interactive learning strategies and machine learning algorithms. When the fallibility

level is higher than 0% however, the difference becomes noticeable. The biggest decrease in performance can be seen in the experiments using SVM, displayed in Figs. 5c and 5d. Since SVM tries to optimize the positioning of hyperplanes to as much as possible separate the different classes, the task will get increasingly more difficult as the number of incorrect labelled instances increases in the dataset used for training. The steepest drop in performance of these two can be seen in Fig. 5d, where the active learning strategy triggered by uncertainty is employed. The measurement of uncertainty implemented for SVM is based on the distance from the new data instance to be tested, to the hyperplanes. If the SVM classifier has trouble positioning the hyperplanes correctly due to incorrect training data, the uncertainty measurement which is dependent on this position will also be inadequate. While less extreme, the effect of an increasingly larger portion of training data being incorrect is visible for the Naïve Bayes classifier and k-Nearest Neighbor as well.

An interesting observation from the experiments is that the Naïve Bayes classifier appears more robust towards fallibility compared to reluctance while the opposite is true for SVM and k-Nearest Neighbor. The possible explanation for the poor results of SVM when fallibility is introduced is discussed above. For k-Nearest Neighbor, the higher the level of fallibility, the bigger the risk that the k closest instances are incorrect, which in turn leads to a faulty classification. For SVM and k-Nearest Neighbor the experiments show that it is better with a user that might not provide as many labels, but when they do they are correct. Naïve Bayes classifier on the other hand is a generative model which classifies by used mean values generated from the labelled data obtained. Depending on the nature of data, the averaging can smooth possible noise in the data and still create useful mean values. For Naïve Bayes classifier, the experiments indicate that a user who provides more data, even though some instances have incorrect labels, is preferable to a user that is more restrictive but always correct.

The experiments with a fallible user are meant to simulate a user that is not always correct in assessing what label currently is representative of the state to be classified. In a real-world scenario, a user that is sometimes incorrect in this assessment might not always recognize when the estimation of the machine learning algorithm is incorrect either. This is not included in the experiments where the machine teaching strategy is employed and might therefore not portray the entire spectrum of possible effects of a fallible user.

Another factor of the experimental setup that might affect the results is the choice of which data instances that are affected by fallibility or reluctance. As explained in section 3.3, the data instances that are either not provided, in the case of a reluctant user, or provided with incorrect labels, in the case of a fallible user, are chosen at random. In certain scenarios it might be reasonable to assume that the probability of all the instances to be chosen are evenly distributed. For instance, if the user is distracted by another task they are performing, they might sometimes, i.e. in a random pattern, miss to provide a label in accordance with the given learning strategy. For a user that is attentive but not as knowledgeable of what the correct label is on the other hand, the probability of which labels

are not provided or given an incorrect label might be correlated to the data instance itself. For example, a data instance belonging to one label, but that is close to the boundary of another, might be more difficult for the user than a data instance that is a typical example of the same class.

5 Conclusion and Future Work

In this work we explored how the reliability of the user providing labels affects the performance of online machine learning in a cold start scenario. We also studied the robustness of different interactive learning strategies and different machine learning algorithms with regards to a user that can be fallible and reluctant respectively. The results show that the choice of interactive learning strategy and machine learning algorithm has an effect on performance in the experiments, where the combination of Naïve Bayes classifier and the machine teaching strategy triggered by error overall resulted the highest performance. This combination is also most robust towards increased levels of fallibility and reluctance of the user. The overall least robust machine learning algorithm was SVM, especially for a fallible user.

In future work we plan to further validate our conclusions by testing on other datasets and more machine learning algorithms. We also aim to further explore how varying the level of reliability of a user can affect performance.

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. Bouguelia, M.R., Nowaczyk, S., Santosh, K., Verikas, A.: Agreeing to disagree: active learning with noisy labels without crowdsourcing. International Journal of Machine Learning and Cybernetics **9**(8), 1307–1319 (2018)
4. Donmez, P., Carbonell, J.G.: Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In: Proceedings of the 17th ACM conference on Information and knowledge management. pp. 619–628. ACM (2008)
5. Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., Bouchachia, A.: A survey on concept drift adaptation. ACM computing surveys (CSUR) **46**(4), 1–37 (2014)
6. Khan, Z.A., Samad, A.: A study of machine learning in wireless sensor network. Int. J. Comput. Netw. Appl **4**, 105–112 (2017)
7. Krawczyk, B.: Active and adaptive ensemble learning for online activity recognition from data streams. Knowledge-Based Systems **138**, 69–78 (2017)
8. Lughofer, E.: On-line active learning: a new paradigm to improve practical useability of data stream modeling methods. Information Sciences **415**, 356–376 (2017)

9. 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)
10. 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)
11. Pohl, D., Bouchachia, A., Hellwagner, H.: Batch-based active learning: Application to social media data for crisis management. Expert Systems with Applications **93**, 232–244 (2018)
12. Ramdas, A., Poczos, B., Singh, A., Wasserman, L.: An analysis of active learning with uniform feature noise. In: Artificial Intelligence and Statistics. pp. 805–813 (2014)
13. Settles, B.: Active learning literature survey. Tech. rep., University of Wisconsin-Madison Department of Computer Sciences (2009)
14. Shickel, B., Rashidi, P.: Art: an availability-aware active learning framework for data streams. In: The Twenty-Ninth International Flairs Conference (2016)
15. Yan, S., Chaudhuri, K., Javidi, T.: Active learning from imperfect labelers. In: Advances in Neural Information Processing Systems. pp. 2128–2136 (2016)
16. Zeni, M., Zhang, W., Bignotti, E., Passerini, A., Giunchiglia, F.: Fixing mislabeling by human annotators leveraging conflict resolution and prior knowledge. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies **3**(1), 1–23 (2019)
17. 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)
18. Zhu, X., Singla, A., Zilles, S., Rafferty, A.N.: An overview of machine teaching. arXiv preprint arXiv:1801.05927 (2018)
19. 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)