Personal and Ubiquitous Computing (2024) 28:273–286
https://doi.org/10.1007/s00779-020-01414-2

ORIGINAL ARTICLE

Activity recognition through interactive machine learning in a dynamic sensor setting

Agnes Tegen[1] · Paul Davidsson[1] · Jan A. Persson[1]

Received: 12 November 2019 / Accepted: 8 May 2020 / Published online: 9 June 2020

Abstract
The advances in Internet of things lead to an increased number of devices generating and streaming data. These devices can be useful data sources for activity recognition by using machine learning. However, the set of available sensors may vary over time, e.g. due to mobility of the sensors and technical failures. Since the machine learning model uses the data streams from the sensors as input, it must be able to handle a varying number of input variables, i.e. that the feature space might change over time. Moreover, the labelled data necessary for the training is often costly to acquire. In active learning, the model is given a budget for requesting labels from an oracle, and aims to maximize accuracy by careful selection of what data instances to label. It is generally assumed that the role of the oracle only is to respond to queries and that it will always do so. In many real-world scenarios however, the oracle is a human user and the assumptions are simplifications that might not give a proper depiction of the setting. In this work we investigate different interactive machine learning strategies, out of which active learning is one, which explore the effects of an oracle that can be more proactive and factors that might influence a user to provide or withhold labels. We implement five interactive machine learning strategies as well as hybrid versions of them and evaluate them on two datasets. The results show that a more proactive user can improve the performance, especially when the user is influenced by the accuracy of earlier predictions. The experiments also highlight challenges related to evaluating performance when the set of classes is changing over time.

Keywords Interactive machine learning · Activity recognition · Internet of things · Active learning · Machine learning

1 Introduction

Ongoing advances in Internet of things technology lead to new possibilities within the application area of smart environment and activity recognition [1, 21, 26]. With an increasing number of devices in our surroundings streaming data, the opportunities to collect information about those surroundings increase. In sensor rich environments, the set of sensors that is streaming data might not be constant over time. Furthermore, as labelled data typically is costly to acquire, there might not exist any to train the machine learning model with. In this case the model can be incrementally trained on labelled data that is repeatedly provided by a user. It is unrealistic in most scenarios to assume that that the user can label all incoming data. In most cases only a limited ratio of the data can be labelled. Because of the restrictions on labelling data, it is of importance to choose the data instances to label such that the information for the machine learning model is maximized. The focus of our work is to classify the state of a specified environment with a dynamic set of sensors and limited labelled data provided by users.

By a dynamic set of sensors, we mean that the set of sensors streaming data is varying over time. The reasons for the dynamicity may vary, e.g. the sensors might be mobile and can enter or leave the environment at different points in time, they might stop streaming due to sensor malfunction, or there might be network problems.

The estimation of the activity or environmental state is done at each point in time by gathering and fusing data collected from the sensors currently available in

✉ Agnes Tegen
agnes.tegen@mau.se

Paul Davidsson
paul.davidsson@mau.se

Jan A. Persson
jan.a.persson@mau.se

[1] Internet of Things and People Research Center, Malmö University, Malmö, Sweden

the environment. By including mobile sensors, e.g. in smartphones, the number of possible data sources increases compared with a static set of sensors, but it also introduces challenges. For instance, by using data from a set of sensors that is changing over time, a machine learning model used for learning and data fusion must adapt to use the sensors that currently are streaming data in an optimal way.

When using supervised or semi-supervised machine learning techniques, not only data, but also labelled data is needed, e.g. the activity corresponding to the current sensor data. While the amount of generated data grows with an increased number of devices streaming data, the annotation of the data is still costly and often difficult to acquire. When dealing with streaming data in real world applications, there might not be any labelled dataset available to train the model with at the start. In this case, often referred to as the cold start problem, the model needs to be incrementally trained on data that is annotated gradually over time. Furthermore, in the case of streaming data and real-time estimation, the statistical properties of the data might change over time. This means that the information gathered at one point in time might not correctly represent the scenario later. To resolve this issue, referred to as concept drift, the model needs to be updated with new labelled data [16].

The aim in active learning is that only a subset of the entire dataset has to be labelled and used for training while still receiving the same performance. The size of the chosen subset is constrained by a labelling budget. Typically, there is a trade-off between performance of the machine learning model and the amount of labelled data needed. In active learning, the subset is chosen with the goal to optimize performance given the budget available. To obtain the labels, an oracle is queried, which can be a human (expert or non-expert) or another system. Generally, it is expected that the oracle will always reply with a correct label when queried. In many real-life scenarios these assumptions are unrealistic, especially if the oracle is a human expected to reply in real-time. We introduce interactive learning as an extension of active learning. To distinguish interactive learning from active learning, we adapt the term “user”, instead of “oracle”. The term “oracle” implies an all-knowing entity, while the term “user” is better suited for a situation where a person present in the environment provides labelled data. With interactive learning we include the possibility for the user to provide a label without being queried but instead by their own initiative, as in machine teaching [34, 35]. By relaxing the assumptions of active learning and giving the user possibility to be proactive in the learning process, our goal is to investigate how different interactive machine learning strategies affect the performance.

In this paper, we build upon previous work [29–31] and examine how different labelling strategies affect the performance of the estimations from the learning strategy. We use datasets simulating an environment of a user and dynamic sensors streaming data. We present a first version of a taxonomy for different strategies a user might have, or in other words, what might influence a user whether or not to provide labels.

2 Related work

Activity recognition is a popular research field, the use of which has been acknowledged within many application areas related to smart cities, such as, safety, security, medicine and smart buildings. Recently, deep learning methods applied to activity recognition problems have in many cases outperformed other methods. Among the most popular methods are Convolutional Neural Network, Recurrent Neural Network and Restricted Boltzmann Machine [32]. Ordóñez et al. introduce a framework for activity recognition based on convolutional and LSTM recurrent units and test it on two benchmark datasets [20]. Ronao et al. propose a deep Convolutional Neural Network for human activity recognition using sensors in a smart phone [24]. Hasan et al. use deep hybrid feature models that are incrementally trained through active learning [12]. All three works have divided up the data set into two, one used in a initial train phase, and the other used for testing, i.e. none addresses the cold start problem and require very large amounts of data to be available before any actvity recognition can be performed. While there are several advantages to using deep learning for activity recognition tasks, for instance the features can be constructed by the algorithm without the need of human experience or domain knowledge, it also comes with challenges such as being used in real-time and reliance on training data [32]. Deep learning is dependant on an adequate amount of data being available to train the model on before it can be used to produce estimations. Since we are focusing on a cold start scenario within this work, training the model before prediction starts is not possible.

The challenge of lacking labelled data for activity recognition tasks has also been addressed by using unsupervised learning [3, 9, 17, 23, 33]. Kwon et al. propose unsupervised learning methods that use data from smartphone sensors for activity recognition [17]. Ye et al. introduce USMART, a technique based on unsupervised learning that combines knowledge- and data-driven techniques [33]. Azkune et al. introduce a system that combines unsupervised learning and knowledge-based activity models [3]. They receive comparable results with supervised learning approaches in the results presented. The setup of the experiments in these works are all done in an offline manner, i.e. not in the single-pass streaming fashion that is the case in many real world activity recognition applications, and that we employ in our experiments. Using unsupervised learning has the advantage of not needing labelled training data [23, 33]. However, the models are dependent on a domain expert to provide knowledge to the model, a work effort that amounts to roughly one days work for the expert in the systems proposed by Riboni et al. and Azkune et al. [3, 23]. The presented strategies can be useful in many settings, but in the cold start setting of this work, it is assumed that there is no prior training or knowledge provided.

Stikic et al. explore different methods that could reduce the required amount of labelled data, including active learning approaches [27, 28]. Data from wearable sensors are used to aid psychiatrists in providing insight into their patients behavioural patterns in a work by Dietrich et al. [8]. The authors explore how different visualizations can be used as feedback to the psychiatrists. Compared with these, in our problem setting the feedback has to be provided in real-time, starting from a cold start scenario.

Active learning within a static setting without streaming data is a well-studied problem [16, 25]. The subset of data to be labelled by the oracle is selected at one point in time as all data is available from the start, based on the active learning strategy and the labelling budget. With streaming data however, the set of samples cannot be chosen beforehand, because they arrive one sample at a time. Instead, the choice of whether to query or not has to be made with each arriving sample, still adhering to the labelling budget over time. There are some work that have studied active learning with streaming data, e.g. Lughofer presents an overview of concepts, techniques, applications, challenges and more related to online active learning [18]. Online learning is a specific case of learning from streaming data, where one instance of data is processed at a time. Cheng et al. present a systematic framework with a feedback-driven active learning approach for streaming data with multiple classes and evaluate it on various datasets [7]. The approach aims to balance exploration and exploitation during the learning process.

Regardless of whether the data is streaming or not, active learning typically assumes that if and when queried, the oracle will always provide a correct label. However, in many real world applications this assumption is not realistic. Rather, the oracle is a human that is not necessarily an expert and is asked to provide labels in real-time. Furthermore, the person expected to answer the queries might not be the same person over time, as people might come and go through the environment, and sometimes no label at all is provided.

Previous work that takes into account the human factor of the oracle in active learning have mostly focused on interaction design or the user experience [2, 13]. Donmez et al. present a work where the assumptions of the oracle in active learning are relaxed [10]. The oracle can be reluctant (i.e. might not reply to a query) and unreliable (i.e. might provide an incorrect label). The authors suggest that by introducing these parameters to the oracle, the model is better suited for real-world applications. Cakmak et al. perform experiments where different levels of involvement and guidance from the oracle are compared with regular active learning [4]. It is pointed out that performance can be improved for active learning if the oracle also is allowed to have a more active role.

3 Experimental setup

In this section, the different machine learning methods and the different interactive learning strategies are explained. The setup of the experiments performed, along with the datasets used, are also presented.

Several popular evaluation methods for batch learning, e.g. cross-validation, are not directly applicable in a setting with streaming data. An evaluation method used in this type of scenario needs to work for the cold start problem, as well as respect the temporal order of data. The evaluation procedure used for all experiments was test-then-train [11]. When a new data instance is received through the incoming stream of data, the system first produces an estimate of the given instance. After this, the interactive learning strategy decides whether or not the instance should be labelled. If a label is provided, the machine learning model is incrementally trained with the new labelled data instance. The performance is then measured by calculating the accumulated accuracy over time.

3.1 Machine learning approaches

Three different machine learning approaches were used in the experiments: Support Vector Machine, k-Nearest Neighbor and Naïve Bayes classifier. Different parameter settings were tested for all machine learning approaches, but here we will only present results from using the ones best suited for the given problem. The classifiers also had to be adapted to be able to handle the dynamic sensor setting. The adaption is described in the section for each respective method.

Because of the regular influx of new data, all data cannot be stored indefinitely. Instead, an upper limit is set of how many labelled instances can be stored per class. If the maximum number of instances are reached and a new sample arrives, the oldest sample is discarded. After different limits of sample size were tested, the limit was set to 50 labelled instances of each class. This was big enough to represent possible variation within a given class, while still small enough to be updated in the case of concept drift.

3.1.1 Naïve Bayes classifier

The Naïve Bayes classifier with Gaussian distributions assumed for the variables was included in the experiments. The classifier has several advantages that makes it a suitable choice for our scenario [15]. Compared with many other machine learning approaches, it requires less training data. It is not computationally complex, which is important when estimations are produced in real-time and is also suitable for online learning. The Naïve Bayes classifier handles the different features separately, which makes it easier to adapt to the dynamic sensor setting. If a sensor stops streaming, the corresponding feature is simply discarded from the model, as the other features are not affected by it. If a new sensor starts streaming, the labelled data instances including this new feature are stored in parallel with a dataset with the old set of features. When enough data is collected of the new instance it can be added to the model used for estimation.

3.1.2 k-Nearest Neighbor

The k-Nearest Neighbor approach is a good choice in our scenario because of its simplicity [14]. Basically, the method classifies a new sample based on the labels of the k instances in the dataset that most resembles the sample given a distance metric. The label which is most frequent among these k neighbors is assigned to the new label. In our experiments we had $k = 3$. The simplicity of the method makes it suitable for online learning and its computational work can be limited through the number of labels being saved. Furthermore, only k labelled data instances are needed to start classifying, which is useful in a cold start scenario.

The adaption of k-Nearest Neighbor to the dynamic sensor setting was similar to that of the Naïve Bayes classifier, but with some changes. As the features are not dealt with independently, the addition of a new sensor is less flexible for these methods. A larger number of labelled samples are needed before including a new sensor in the model. In this case, the samples that were added before the arrival of the new sensor does not contain a value for the new feature, but were filled with an average of the so far collected samples of the new sensor.

3.1.3 Support Vector Machines

Support Vector Machines with a polynomial kernel was also chosen for the experiments. Support Vector Machines are, apart from also being suitable for online classification, efficient regarding memory usage and dealing with high-dimensional datasets [19]. Efficient memory usage is important for real-time estimations. Being able to efficiently deal with high-dimensional data is useful in a dynamic sensor setting which can result in many features. The adaption of Support Vector Machines to the dynamic sensor setting was the same as for k-Nearest Neighbor.

3.2 Interactive learning strategies

In this work we explored what can influence a user to provide feedback in a real-time learning scenario and implemented interactive learning strategies based on this. Active learning can be seen as a special case of interactive learning, where the user is triggered to provide a label by being queried. However, other factors might prompt the user to provide, or stop the user from providing, a label. Below, the different factors detected that could influence a user are described together with the implemented interactive learning strategy used in the experiments.

– *Uncertainty*: In active learning, the user will provide a label when queried by the active learning strategy. One of the most popular active learning strategies is to query when the model is uncertain regarding the estimation. To decide when the model is uncertain an uncertainty measurement has to be defined. There exists several different versions of uncertainty measurement, but they all depend on a boundary, such as a threshold for the uncertainty measure, labelling budget and the implementation of the strategy. Since we have three different machine learning approaches, each one needs their own implementation of an uncertainty measure.

  For the Naïve Bayes classifier, we use the Variable Uncertainty Strategy presented by Žliobaitė et al. [36]. The classifier produces probabilities for all classes and the class with the highest probability is then chosen for the classification. The probability is compared with a given threshold to decide if the estimation is uncertain or not (i.e. if the model should query). In this strategy the threshold is not static over time. If there are few queries, it might indicate that the threshold is set too high and so the threshold is gradually lowered and vice versa.

  For the Support Vector Machine and k-Nearest Neighbor, we chose active learning strategies proposed by Pohl et al. [22]. For Support Vector Machine, the distance from the given data point to the hyperplane was chosen as the measurement of how certain the classifier is. Like with the Naïve Bayes classifier, this measurement is compared with a threshold that can be altered over time. In the active learning strategy for k-Nearest Neighbor, more than two-thirds of the neighbors must have the same label for the estimation to not be considered uncertain.

– *Error*: A user can be influenced by the output from the machine learning model. The model produces estimations of the current state of the environment, and in some cases also the confidence of the given prediction or, in the case of multi-class classification, the probability of other candidates. If a user has access to this type of information, it could influence them on whether to provide or withhold labels. In our experiments, the strategy has been implemented so that the user will provide a label when the previous prediction was erroneous.

- *State change*: Since the users are assumed to have knowledge of the current state of the environment, as they themselves are present, this awareness can trigger users to provide labels. For instance, assume the machine learning model is supposed to estimate the currently ongoing activity in an office setting. If the user first sits and works by themselves, but then several other people enter and a meeting starts, the status has changed from silent work to meeting, which will trigger the user to provide a label. In our implementation, the user provides labels when there is a change in the state of the environment, otherwise not.
- *Time*: The user may provide labels at certain points in time, regardless of queries, the correctness of estimations, and the state of the environment. In this strategy, labels are provided with a given frequency which is calculated based on the labelling budget. The user can either by their own initiative provide labels with a given time interval, or they can be queried by a labelling strategy systematically.
- *Random*: Labels can also be provided at random. In this case there can either be a labelling strategy that queries the user at random or the user themselves might provide labels at random. The user might have reasons related to themselves regarding why they would provide labels or not. For instance, a user might provide fewer labels when they are more stressed. These labelling patterns might seem random, but they are not necessarily that. While they might appear random with regards to the sensor data, estimations, time etc, it might only be because the dataset does not include the relevant features such as the stress level of the user over time. The lack of appropriate data is the reason these types of strategies are not included here. However, it could be interesting to explore such strategies further in future work.

While the factors are listed separately above, they could be combined to different degrees. For example, the decision to provide a label or not could be based both on being queried by the model and what is happening around the user in the environment. To examine how such hybrid strategies performs compared with the separate strategies, a number of hybrid strategies were implemented as well. The different implementations and combinations of strategies leads to a large amount of possible hybrid strategies. In this work, we decided to focus on the combinations of the strategy *Uncertainty*, with the other interactive learning strategies listed above. We chose these combinations, as the decision on whether a label will be provided would not only be with the learning model nor with the user, but based on strategies of both. The different hybrid approaches are described below.

- *Uncertainty + Error*: In this strategy, the first step is the *Uncertainty* strategy (described further above). The model queries the user when uncertain regarding its prediction, but the user will only respond to the query with a label if the latest prediction was incorrect.
- *Uncertainty + State change*: Similarly to the previous strategy, the first step is *Uncertainty*, querying whenever the uncertainty of prediction is considered big enough. However, the user will only respond if the state of the environment has changed since the last label was given.
- *Uncertainty + Random*: Also this strategy consists of two steps, where the first is *Uncertainty*. In this case however, it is randomly generated whether the user will provide a label as a response to the query. This strategy is similar to a setting where the *Uncertainty* strategy is employed, but the user does not always reply to a query.
- *Uncertainty + Time*: This strategy is also *Uncertainty* combined with another interactive learning strategy, but differs from the other hybrid strategies. Instead of one strategy following the other in two separate steps, the two strategies are run in parallel. The interactive learning strategy, *Time*, has a separate counter that keeps track of how many instances has appeared since the last sample was labelled. Given that there is enough budget, a label is provided at a certain frequency, which is calculated from the labelling budget. At each point in time, the frequency and the uncertainty of the prediction is tested in parallel and if either one decides to provide a label or query, a label for the sample will be given. However, even though the strategies are run in parallel, they share the same labelling budget.

3.3 Datasets

We used two separate benchmark datasets within Activity Recognition and smart environment to evaluate the proposed methods. Both contain collections of sequential data, i.e. the data was recorded in sessions, from multiple heterogeneous sensors. Simulations were done on the datasets to create the dynamic set of sensors that is changing over time.

3.3.1 Opportunity dataset

The Opportunity dataset is a collection of recordings of different daily living activities performed separately by 4 subjects in a sensor-rich environment [6]. The subjects performed a scripted scenario in five different runs, giving a total of 20 recorded data sequences. The sequences are between 12.35 minutes (22230 instances) and 28.4 minutes (51116 instances) long. In total 72 different sensors, resulting in 242 features, were used for the recordings, some sensors were worn by the subject, others placed on objects in the room of the recording. For the experiments, 19 of the separate recordings were used (one was excluded, as it contained too many missing values). All tests were run on the data sequences separately and then averaged.

The data is annotated in several abstraction levels. For the experiments in this work, the highest abstraction level was used which includes four labels and a null category (which represents unlabelled data instances). The annotations at this level describe the mode of locomotion for the subject (“Stand”, “Walk”, “Sit” or “Lie”).

The labelling budget was set to 5% for the experiments on the Opportunity dataset based on tests on the different machine learning approaches and the interactive learning strategies. Figure 1 displays that 5% leads to a good balance between keeping the budget low, while still trying to maximize performance. The labelling budget is in reference to the number of incoming samples, not the timestamp. With this dataset a 5% labelling budget would equal 90 labels being provided each minute on average, which might not be realistic. While an ideal scenario would be recorded over longer periods of time, we consider the results to still be representative for our problem scenario, as the order in the sequences are maintained and the learning approaches does not use the timestamp as a feature.

3.3.2 Occupancy dataset

The Occupancy dataset contains recordings of whether or not a room in an office setting is occupied [5]. The dataset contains 3 separate sequences of recorded data, varying in length from 1.8 days (2664 instances) to 6.8 days (9752 instances) (6.8 days) and five different features (light, temperature, humidity, CO2-level and humidity ratio). The pattern of when the room is occupied follows regular working hours during weekdays, but the room was empty the entire day on weekends.

With the Occupancy dataset, there was not as many separate recordings as with the Opportunity dataset. Instead, the data was divided by days (from midnight to midnight) and randomly shuffled for each simulation, to obtain variation in the sequence of incoming data. In total, 100 simulations were run and then averaged to produce the presented results. The labelling budget for the experiments on the dataset was set to 1%, based on findings in earlier work [29].

3.3.3 Simulation of dynamic sensor set

To simulate a dynamic set of sensors streaming data, the access to some sensors were restricted during periods of time. The choice of sensors to be dynamic, and during which part of the sequence their data streams would be restricted, were randomly generated for each simulation. A sensor can be present at first, but later drop off, it can be absent from the start, but start streaming data at some point in time or a combination of the two. In case of a combination, the sensor could be streaming at the start, later drop off, and even later reappear again or vice versa. If a sensor was streaming at the start, then stopped and later start streaming again, the data collected from the first period of streaming was not stored

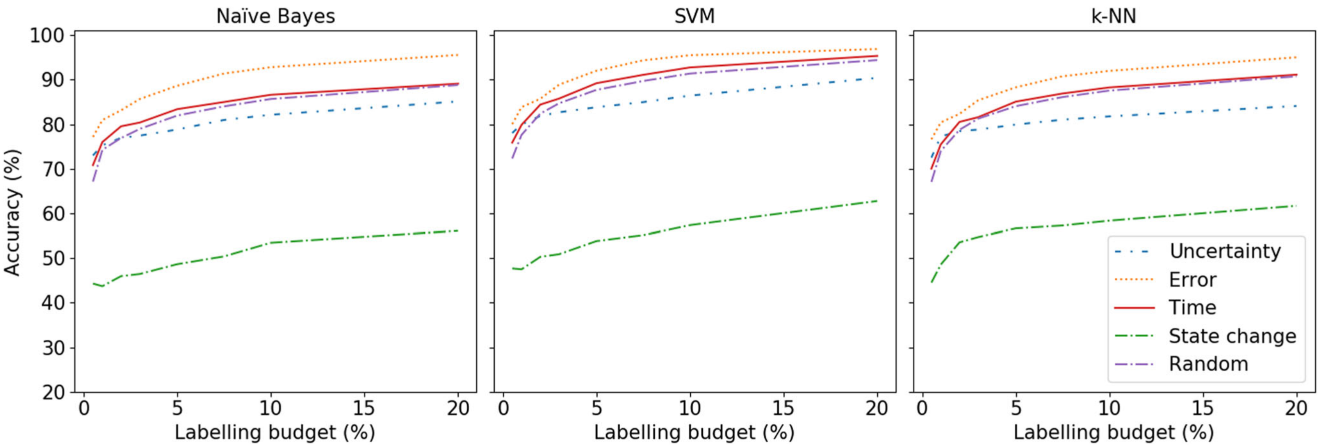

Fig. 1 The accumulated accuracy for different labelling budgets and interactive machine learning strategies for the Opportunity dataset

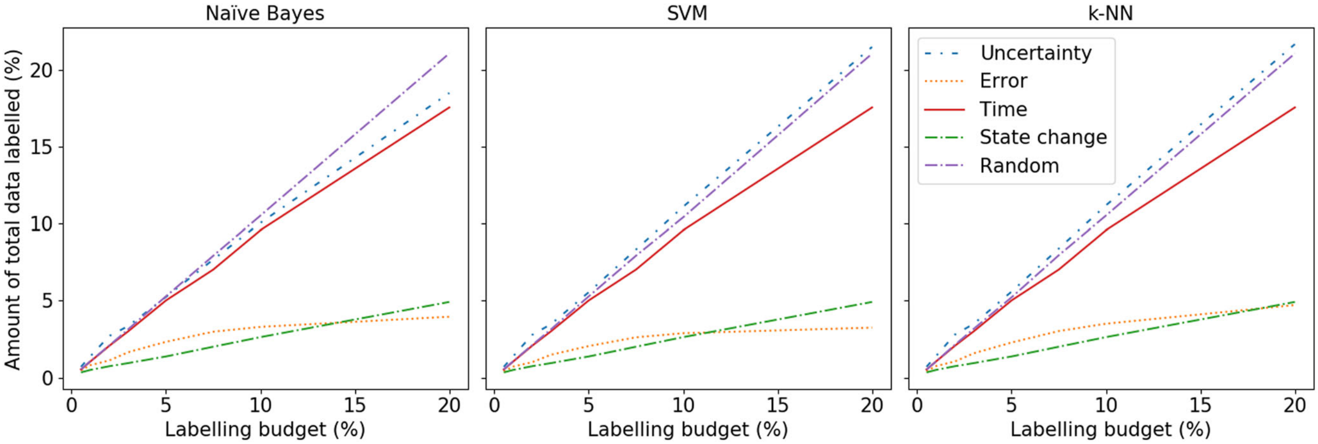

Fig. 2 The actual amount of data labelled for different labelling budgets and interactive machine learning strategies for the Opportunity dataset

in the model for the reappearance. In a sensor-intensive environment there might otherwise be a risk of an ever increasing amount of data from sensors that might never reappear. An alternative could have been to store the data for a limited time for the possibility of the sensor reappearing within this time period. While this was not implemented in the following experiments, it is planned to be investigated in future work.

First, experiments were done where the number of dynamic sensors was set to a fixed number. When each dynamic sensor would be streaming data and which sensors they would be, was still randomly generated. In the second part of the experiments, the number of sensors with a dynamic presence may vary between simulations, but are bounded by a minimum and maximum value. For the Opportunity dataset, 20–80% of the sensors had a dynamic presence, while it was 20–60% of the sensors for the Occupancy dataset. The reason for the different intervals is because the two datasets have different number of features. While the Opportunity dataset has 242 features, the Occupancy dataset only has five features. For each dynamicity value, the average number of streaming sensors over time has been calculated and it is this number that is displayed in the results.

4 Results

Figure 1 displays how different labelling budgets affects the accumulated accuracy for the three machine learning methods and the interactive learning strategies on the Opportunity dataset. The figure shows that an increased budget leads to increased performance, with decreasing relative improvements in performance for larger budgets. When choosing a labelling budget, the trade-off between a lower budget and a higher performance has to be considered. While the optimal labelling budget might differ slightly for the different strategies, due to consistency in the

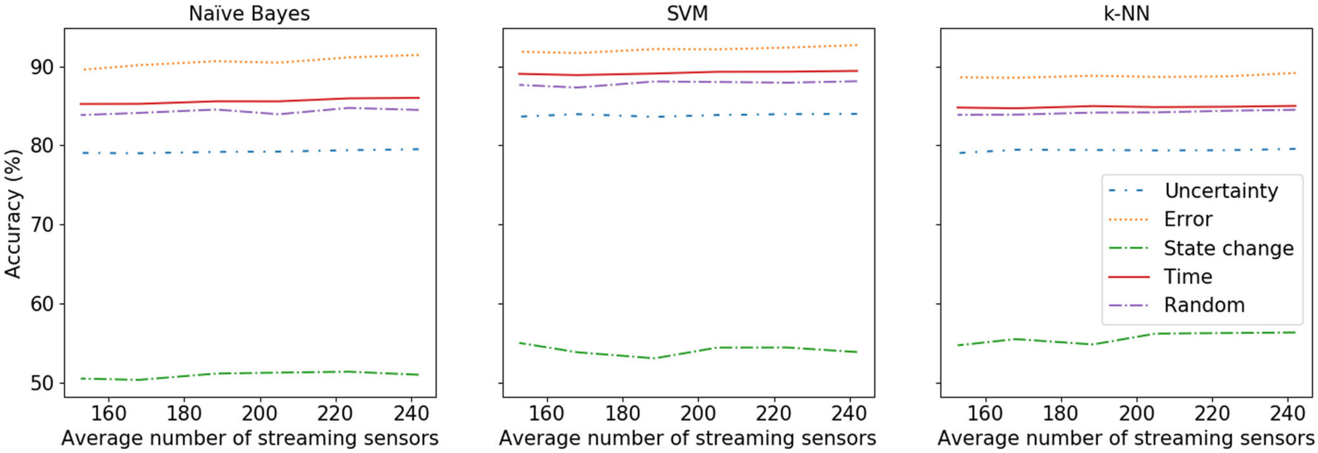

Fig. 3 The accumulated accuracy over the average number of streaming sensors for the separate interactive learning strategies on the Opportunity dataset

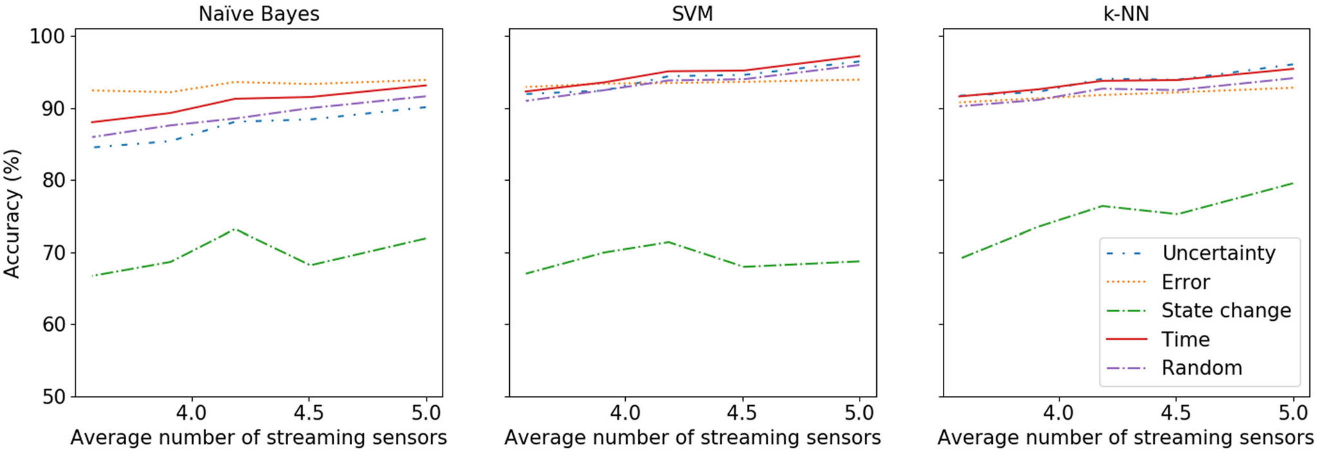

Fig. 4 The accumulated accuracy over the average number of streaming sensors for the separate interactive learning strategies on the Occupancy dataset

experiments one budget was chosen for all strategies. When taking the results of all strategies into account, the labelling budget was set to 5%.

As Fig. 2 illustrates, even though the labelling budget is increased, the actual amount of labelled samples is not necessarily increased with an equal amount. While *Uncertainty* and *Time* in most cases uses up their labelling budget, *Error* never use up the allowed budget, but still manages to perform better then the other two strategies.

In Figs. 3 and 4 the results from the experiments when varying the dynamicity of the streaming sensors are showcased. The figures show the final value of accumulated accuracy given an average number of streaming sensors. For the Opportunity dataset, the results are the average of simulations done on 19 separate recordings. For the Occupancy dataset, the results are the average of 100 simulations.

The results from the experiments on the separate interactive learning strategies and machine learning approaches can be found in Figs. 5 and 6, for the Opportunity dataset and the Occupancy dataset, respectively. Both figures display the accumulated accuracy over time, starting from the first prediction. The results in Fig. 5 are the average of 19 simulations, each from a separate recording and a varying dynamic set of sensors. Figure 6 displays the average of 100 simulations, also with a separate and varying dynamic sensor setting.

Figures 7 and 8 contain the results from the hybrid versions of the interactive learning strategies evaluated on the Opportunity dataset and the Occupancy dataset, respectively. The setup is the same as for the tests on the separate interactive learning strategies, regarding number of simulations, the dynamic sensor setting, etc.

The final Fig. 9, contains several plots, all displaying the accumulated accuracy for the separate interactive learning approaches over time. These tests were done on one of the recordings of the Opportunity dataset to give an example of how the learning curves differ when the sequence of data from the recording is preserved compared with when all data points are shuffled. The upper row contains the experiments

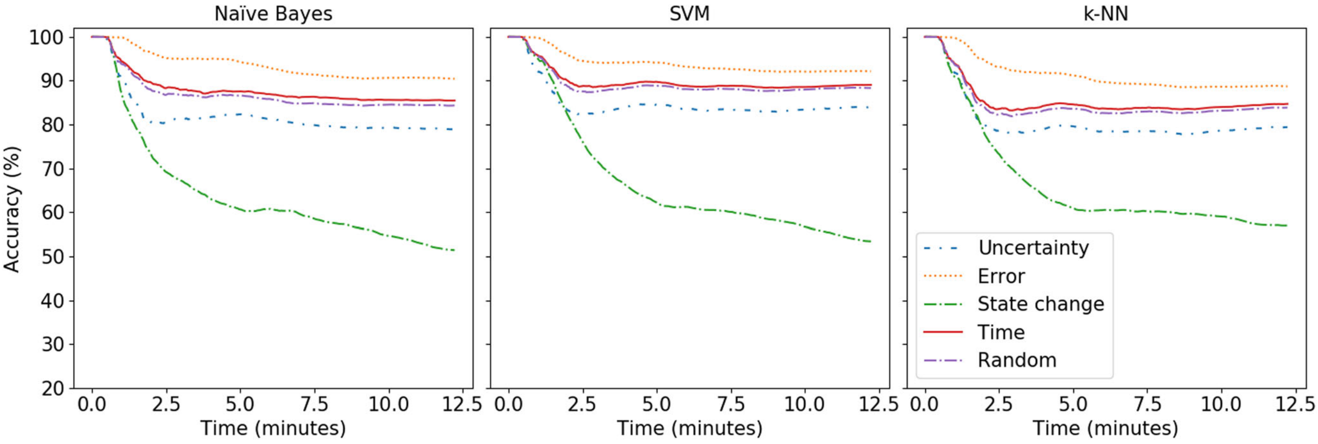

Fig. 5 The accumulated accuracy over time for the separate interactive learning strategies on the Opportunity dataset

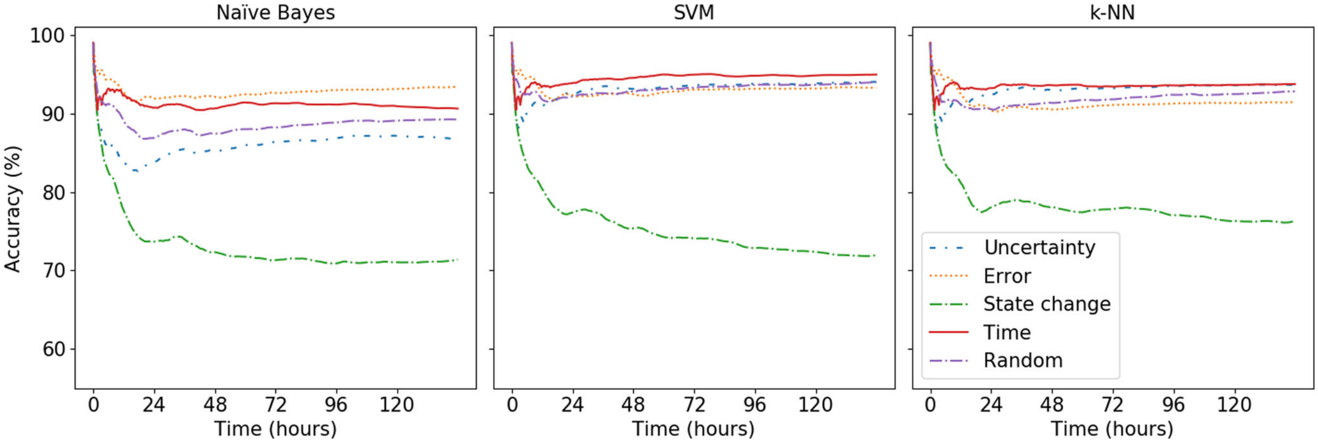

Fig. 6 The accumulated accuracy over time for the separate interactive learning strategies on the Occupancy dataset

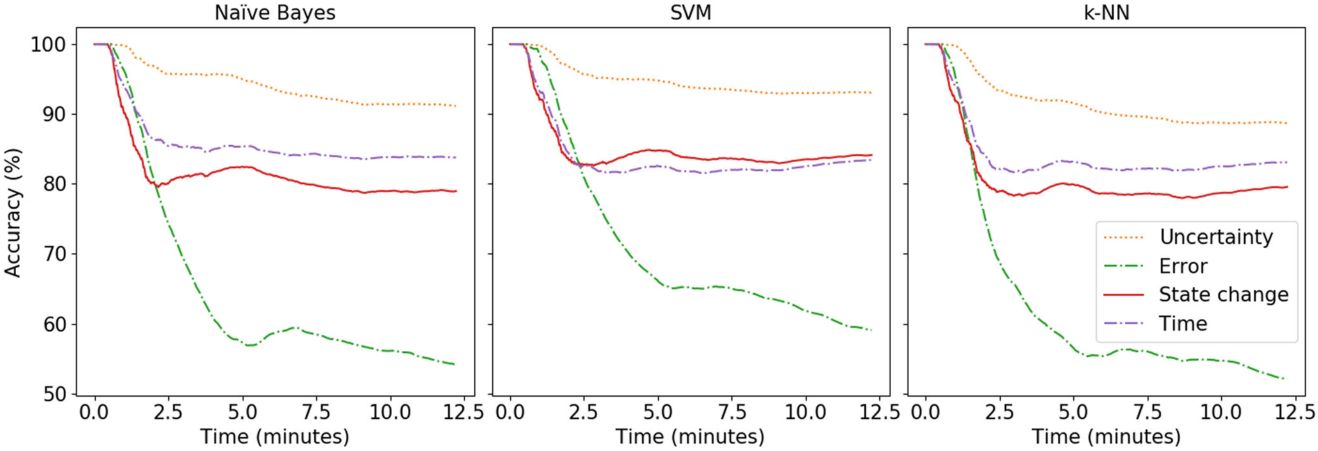

Fig. 7 The accumulated accuracy over time for the hybrid interactive learning strategies on the Opportunity dataset

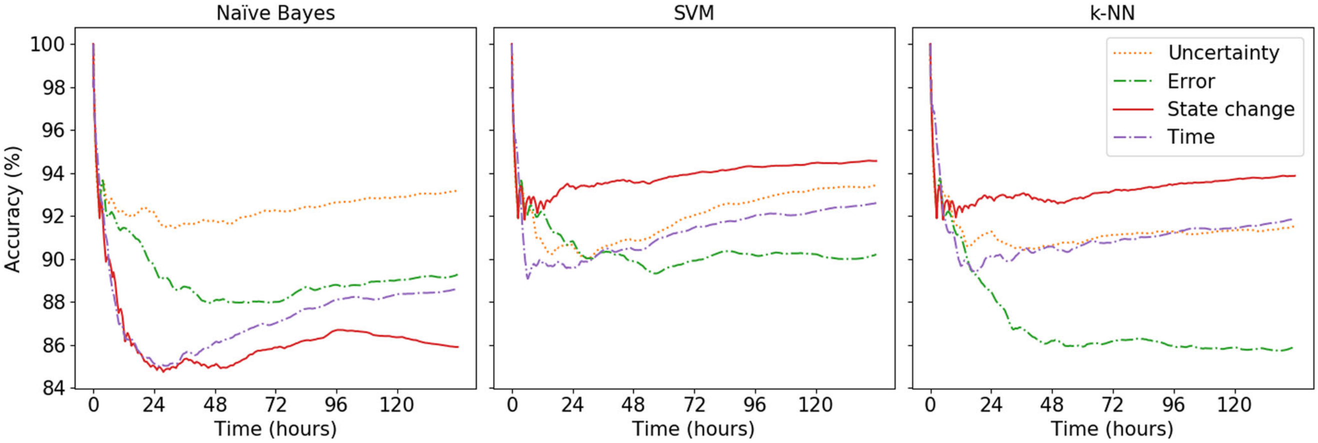

Fig. 8 The accumulated accuracy over time for the hybrid interactive learning strategies on the Occupancy dataset

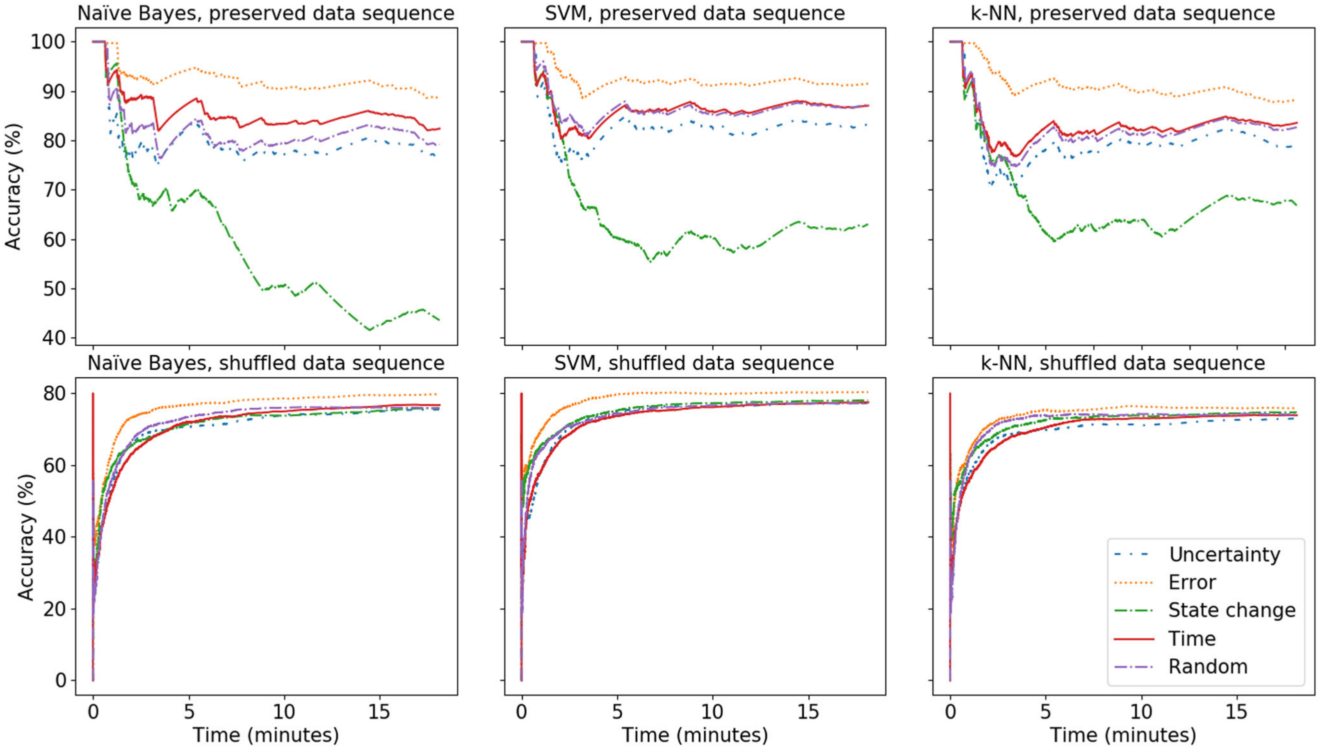

Fig. 9 The accumulated accuracy over time for the separate interactive learning strategies. Top row displays the results when the order of the sequential data is kept, bottom row shows the results when the order is shuffled randomly

where the sequence is kept, while the lower row contains the same data points, but in a randomly shuffled order.

Tables 1 and 2 contain the percentage of all data points that were labelled, i.e. the actual labelling expenses used as opposed to the allowed labelling budget for the experiments with the separate interactive learning strategies. In Table 1 the actual amount of labelling used for the Opportunity dataset is displayed (where the labelling budget was 5%) and in Table 2 the same can be found for the Occupancy dataset (where the labelling budget was 1%). While some strategies used up the allowed labelling budget, all did not. For the experiments on the separate interactive learning strategies, the *Error* and *State change* strategies did not use up the allowed budget in any of the cases. *Error* annotated around 2.26% for the Opportunity dataset and around 0.35% of all incoming samples for the Occupancy dataset. For *State change* the numbers are 1.37% for the Opportunity dataset and 0.30% for the Occupancy dataset. *Uncertainty* even ended up slightly over the allowed budget for the Occupancy dataset, with an actual budget around 1.33%. These numbers where consistent with all of the machine learning approaches used.

Tables 3 and 4 similarly display the percentage of all data points that were labelled for the hybrid interactive learning strategies for the Opportunity dataset and the Occupancy dataset respectively. In the experiments with the hybrid strategies, the number of labelled samples acquired was in general lower, but in some cases on par with the number from the experiments on the separate strategies. The only strategy that used up the budget regardless of dataset and machine learning approach, was *Uncertainty* + *Time*. This approach actually used slightly more than the labelling budget when evaluated on the Occupancy dataset, as the total ended up around 1.33% for all machine learning

Table 1 The labelling expenses used over time on the Opportunity dataset for the separate interactive machine learning strategies and the three machine learning (ML) methods Naïve Bayes (NB), Support Vector Machine (SVM) and k-Nearest Neighbor (k-NN) when the labelling budget was set to 5%

ML	Uncertainty (%)	Error (%)	State change (%)	Time (%)	Random (%)
NB	5.53	2.39	1.37	5.01	5.28
SVM	5.55	2.09	1.37	5.01	5.29
k-NN	5.59	2.30	1.37	5.01	5.27

Table 2 The labelling expenses used over time on the Occupancy dataset for the separate interactive machine learning strategies and the three machine learning (ML) methods Naïve Bayes (NB), Support Vector Machine (SVM) and k-Nearest Neighbor (k-NN) when the labelling budget was set to 1%

ML	Uncertainty (%)	Error (%)	State change (%)	Time (%)	Random (%)
NB	1.34	0.33	0.30	1.00	1.02
SVM	1.30	0.32	0.30	1.00	1.00
k-NN	1.34	0.39	0.30	1.00	1.02

approaches. The hybrid strategy *Uncertainty* + *Error*, had a slightly lower but similar number of labelled data points compared with the separate strategy for both datasets. Both the hybrid strategies *Uncertainty* + *State change* and *Uncertainty* + *Random* acquired a similar amount or fewer samples.

5 Discussion

The results make it clear that the choice of interactive learning strategy has a significant effect on the performance when tested on recordings of streaming data. The *State change* strategy performs worst in almost all instances, regardless of machine learning approach or the dataset tested on. One reason might be that the strategy does not use up the allowed labelling budget, but it is not the only explanation, as the *Error* strategy does not use it up either (see Tables 1 and 2). In fact, *Error* often collects a similar amount of labelled samples as the *State change* strategy for the Occupancy dataset. The *Error* strategy is the best, or among the best, performing strategies for all experiments displayed in Figs. 5 and 6. The reason for this could be that the strategy corrects the model as soon as it has made an incorrect classification, leading to a decreased risk of repeating the mistake. The results indicates that choosing an appropriate interactive learning strategy can have a significant impact on performance, as a higher accuracy can be achieved with fewer labels.

For active learning in a non streaming setting, a selection from a set of unlabelled data is chosen according to a specified strategy. By having all data available at once, the full labelling budget can be utilized, without going over the allowed budget either. Handling a labelling budget with streaming data however, means that estimations has to be made, as it only is possible to obtain a label for the current data point and the size of the total amount of data is unknown (or possibly infinite). Instead of using the entire dataset (which would include future unknown data points), the labelling budget is calculated over a window of time. The total labelling budget can therefore only be estimated at run-time. This leads to the approximate labelling budgets that can be found in Tables 1, 2, 3 and 4 and also explains why the actual amount of labels in some cases do not always exactly match the labelling budget.

Another interesting observation from the experiments on the separate interactive learning strategies is that the active learning strategy, *Uncertainty* , has among the worst results in many cases, implying that other information than the uncertainty estimation of the model can be useful when deciding which samples to label. However, this work only contains one implementation per machine learning approach. While the choice of *Uncertainty* was made based on what would suit the problem at hand, it is not possible without further experiments to conclude if the lower performance is due to the strategy being used in general or if it is due to the specific implementations.

When comparing the hybrid approaches, displayed in Figs. 7 and 8, to their respective separate approaches, it varies whether the combination improved, worsened or did not affect the performance. Maybe unsurprisingly, the separate strategies that performed worse had most to gain from a combination with another strategy. The most drastic improvement can be seen for the *State change*

Table 3 The labelling expenses used over time on the Opportunity dataset for the hybrid interactive machine learning strategies and the three machine learning (ML) methods Naïve Bayes (NB), Support Vector Machine (SVM) and k-Nearest Neighbor (k-NN) when the labelling budget was set to 5%

ML	Uncertainty	Uncertainty	Uncertainty	Uncertainty
	+ Error (%)	+ State change (%)	+ Time (%)	+ Random (%)
NB	2.29	0.49	5.58	4.10
SVM	1.90	0.45	5.55	2.67
k-NN	2.33	0.49	5.59	4.17

Table 4 The labelling expenses used over time on the Occupancy dataset for the hybrid interactive machine learning strategies and the three machine learning (ML) methods Naïve Bayes (NB), Support Vector Machine (SVM) and k-Nearest Neighbor (k-NN) when the labelling budget was set to 1%

ML	Uncertainty + Error (%)	Uncertainty + State change (%)	Uncertainty + Time (%)	Uncertainty + Random (%)
NB	0.33	0.35	1.34	0.69
SVM	0.31	0.34	1.31	0.48
k-NN	0.39	0.35	1.34	0.71

strategy, when evaluated with the Naïve Bayes classifier on the Occupancy dataset. In this case the performance is much increased when compared with both *Uncertainty* and *State change* separately. These results again show that the *Uncertainty* strategies tested here are not always good at assessing their own performance.

Figures 3 and 4 display the effect of varying the number of streaming sensors has on performance. For the Opportunity dataset, Fig. 3, the accuracy is almost unchanged for all the interactive learning strategies, even though the number of streaming sensors varies. This is reasonable however, as the dataset contains a large amount of features. As can be seen from the results, even if all sensors are dynamic, i.e. have an interval where they are not streaming data, there are still on average 153 sensors streaming at each point in time.

For the Occupnacy dataset, Fig. 4, the performance does change when the average number of streaming sensors is altered. Interestingly, the performance does not always increase with an increased number of sensors. For most of the strategies, the performance increases to around an average of 4.19 streaming sensors, after this, it either stagnates or even decreases, before it increases yet again. A possible reason for this result can be found with the features in the Occupancy dataset. When studying how well the different features correlates to the status of occupancy in the room, it becomes clear that some of the features have a low correlation. For instance one feature measuring the humidity in the room, has a correlation coefficient close to zero. In the experiments presented in here there was no feature selection, but these results indicate that the model might benefit from adding a preprocessing step of feature selection however. From the results it can also be seen that *Error* seems to be the most stable interactive learning strategy when the number of streaming sensors is changing.

It is worth noting regarding the labelling budget that the case is quite different with streaming data compared with a static dataset. In a setting with streaming data, it is not possible to store all incoming samples for an infinitely long time or to know what the future samples in the stream are. Instead of calculating the budget on the entire dataset, which would include future unknown samples, it is calculated on a set of the most recent data points. Furthermore, the labelling budget is the maximum percentage of incoming data points that the model can query for labels, but there is no minimum value. This means that a restrictive approach might not use up the allowed labelling budget when calculated over a period of time.

The chosen performance measurement for these experiments was accumulated accuracy over time, starting from the very first prediction, when the model in most cases only has one labelled sample. The accuracy over time is interesting to analyse, because of the data being provided sequentially and the fact that the model begins training without any stored data. The performance measurement falls short of providing a complete picture for several reasons however, with the changing number of classes over time possibly being the most prominent one. At the very beginning of each experiment, the learning model has only encountered one class, making the estimation trivial. Gradually, as more classes are introduced, the estimation becomes increasingly more complex. Other common performance metrics for machine learning problems suffer from the same issue, however. To the best of our knowledge there is not an established measurement that takes this into consideration.

The gradual introduction of new classes is one factor in the decreasing accuracy seen in many of the experiments. The accuracy decreased even after all classes had been incorporated into the model however, implying that it was not the only reason for the downward trend. To see whether the specific sequence of data from the recordings could influence the performance curves, tests were carried out where a sequence of data was shuffled in a random order. As Fig. 9 shows, this does affect the shape of the learning curve significantly. Here, all classes are introduced early on, which results in a poor performance at the start. As more labelled samples are collected, the performance increases rapidly at first, but stabilizes after a while. The reason for the smooth look of the curve with the shuffled dataset, is probably because the entire spectrum of possible feature values for each class is introduced from the start. While this might be an overwhelming task at first, the model can learn more effectively. When the order of the sequence is kept the model might first give an impression of being able to estimate well, but as time moves forward the full complexity of the problem unfolds. In real-world scenarios however, streaming data does arrive with a higher possibility of resembling other data points close in time and not as a shuffled sample set for the model to begin its training on.

6 Conclusion and future work

In this work we explored the effect of interactive machine learning strategies on performance in a real-time learning scenario. The strategies were implemented and evaluated on two datasets related to activity recognition with a simulated dynamic sensor setting, both as separate strategies and as hybrid versions. The experiments show that by giving the user a more proactive role, the performance can be increased. In the overall best performing strategy, the decision of whether to provide a label was based on the accuracy of the previous prediction, i.e. a user would provide feedback in the form of a correct label when the previous estimation was incorrect. Future work includes further developing the taxonomy and broaden the different types of implementations and strategies used in experiments. This can be done by, for instance, testing multiple different active learning strategies for comparison, and in combination, with other interactive learning strategies. Future work also include testing more machine learning methods.

The experiments highlight the difficulty in comparing performance over time when the complexity of the problem is not constant, e.g. by a varying number of classes. In future work we intend to define and explore possible measurements suitable for comparing problems with different complexities or in a streaming data setting where the complexity of the problem is changing over time.

Even though the datasets used here were adequate for evaluating the proposed strategies, there is a need of good datasets in the area of activity recognition or smart environment which includes longer recordings of streaming data, heterogeneous types of sensors and multiple classes. The aim for future work is to create an open source dataset with multiple heterogeneous sensors recording the state of office environments.

Funding information Open access funding provided by Malmö University. This research was partially funded by the Knowledge Foundation through the Internet of Things and People grant number 20140035.

Compliance with Ethical Standards

Conflict of interest The authors declare that there is no conflict of interest.

References

1. Alam F, Mehmood R, Katib I, Albogami NN, Albeshri A (2017) Data fusion and IoT for smart ubiquitous environments: a survey. IEEE Access 5:9533–9554
2. Amershi S, Cakmak M, Knox WB, Kulesza T (2014) Power to the people: the role of humans in interactive machine learning. AI Mag 35(4):105–120
3. Azkune G, Almeida A (2018) A scalable hybrid activity recognition approach for intelligent environments. IEEE Access 6:41,745–41,759
4. Cakmak M, Thomaz AL (2011) Mixed-initiative active learning
5. Candanedo LM, Feldheim V (2016) Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models. Energ Build 112:28–39
6. Chavarriaga R, Sagha H, Calatroni A, Digumarti ST, Tröster G, Millán JdR, Roggen D (2013) The opportunity challenge: a benchmark database for on-body sensor-based activity recognition. Pattern Recogn Lett 34(15):2033–2042
7. Cheng Y, Chen Z, Liu L, Wang J, Agrawal A, Choudhary A (2013) Feedback-driven multiclass active learning for data streams. In: Proceedings of the 22nd ACM international conference on conference on information & knowledge management. ACM, pp 1311–1320
8. Dietrich M, Berlin E, Van Laerhoven K (2015) Assessing activity recognition feedback in long-term psychology trials. In: Proceedings of the 14th international conference on mobile and ubiquitous multimedia. ACM, pp 121–130
9. Dimitrov T, Pauli J, Naroska E (2010) Unsupervised recognition of adls. In: Hellenic conference on artificial intelligence. Springer, pp 71–80
10. Donmez P, Carbonell JG (2008) Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In: Proceedings of the 17th ACM conference on information and knowledge management. ACM, pp 619–628
11. Gama J, Žliobaitė I, Bifet A, Pechenizkiy M, Bouchachia A (2014) A survey on concept drift adaptation. ACM Computing Surveys (CSUR) 46(4):1–37
12. Hasan M, Roy-Chowdhury AK (2015) A continuous learning framework for activity recognition using deep hybrid feature models. IEEE Trans Multimed 17(11):1909–1922
13. Johns E, Mac Aodha O, Brostow GJ (2015) Becoming the expert-interactive multi-class machine teaching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 2616–2624
14. Khan ZA, Samad A (2017) A study of machine learning in wireless sensor network. Int J Comput Netw Appl 4:105–112

15. Krawczyk B (2017) Active and adaptive ensemble learning for online activity recognition from data streams. Knowl-Based Syst 138:69–78
16. Krishnakumar A (2007) Active learning literature survey. In: Technical Report. University of California
17. Kwon Y, Kang K, Bae C (2014) Unsupervised learning for human activity recognition using smartphone sensors. Expert Syst Appl 41(14):6067–6074
18. Lughofer E (2017) On-line active learning: a new paradigm to improve practical useability of data stream modeling methods. Inform Sci 415:356–376
19. Mahdavinejad MS, Rezvan M, Barekatain M, Adibi P, Barnaghi P, Sheth AP (2018) Machine learning for internet of things data analysis: a survey. Digit Commun Netw 4(3):161–175
20. Ordóñez FJ, Roggen D (2016) Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. Sensors 16(1):115
21. Perera C, Zaslavsky A, Christen P, Georgakopoulos D (2013) Context aware computing for the internet of things: a survey. IEEE Commun Surv Tutor 16(1):414–454
22. Pohl D, Bouchachia A, Hellwagner H (2018) Batch-based active learning: application to social media data for crisis management. Expert Syst Appl 93:232–244
23. Riboni D, Sztyler T, Civitarese G, Stuckenschmidt H (2016) Unsupervised recognition of interleaved activities of daily living through ontological and probabilistic reasoning. In: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp 1–12
24. Ronao CA, Cho SB (2016) Human activity recognition with smartphone sensors using deep learning neural networks. Expert Syst Appl 59:235–244
25. Settles B (2009) Active learning literature survey. Tech. rep., University of Wisconsin-Madison Department of Computer Sciences
26. Sezer OB, Dogdu E, Ozbayoglu AM (2017) Context-aware computing, learning, and big data in internet of things: a survey. IEEE Internet Things J 5(1):1–27
27. Stikic M, Schiele B (2009) Activity recognition from sparsely labeled data using multi-instance learning. In: International Symposium on Location- and Context-Awareness. Springer, pp 156–173
28. Stikic M, Van Laerhoven K, Schiele B (2008) Exploring semi-supervised and active learning for activity recognition. In: 2008 12th IEEE International Symposium on Wearable Computers. IEEE, pp 81–88
29. Tegen A, Davidsson P, Mihailescu RC, Persson JA (2019) Collaborative sensing with interactive learning using dynamic intelligent virtual sensors. Sensors 19(3):477
30. Tegen A, Davidsson P, Persson JA (2019) Interactive machine learning for the internet of things: a case study on activity detection. In: Proceedings of IoT'19: International Conference on the Internet of Things (IoT'19). ACM
31. Tegen A, Davidsson P, Persson JA (2019) Towards a taxonomy of interactive continual and multimodal learning for the internet of things. In: Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 International Symposium on Wearable Computers. ACM, pp 524–528
32. Wang J, Chen Y, Hao S, Peng X, Hu L (2019) Deep learning for sensor-based activity recognition: a survey. Pattern Recogn Lett 119:3–11
33. Ye J, Stevenson G, Dobson S (2014) Usmart: an unsupervised semantic mining activity recognition technique. ACM Trans Interac Intell Syst (TiiS) 4(4):1–27
34. Zhu X (2015) Machine teaching: an inverse problem to machine learning and an approach toward optimal education. In: Twenty-Ninth AAAI Conference on Artificial Intelligence
35. Zhu X, Singla A, Zilles S, Rafferty AN (2018) An overview of machine teaching. arXiv:180105927
36. Žliobaitė I, Bifet A, Pfahringer B, Holmes G (2013) Active learning with drifting streaming data. IEEE Trans Neural Netw Learn Syst 25(1):27–39