Environmental Modelling and Software 134 (2020) 104869

Contents lists available at ScienceDirect

Environmental Modelling and Software

journal homepage: http://www.elsevier.com/locate/envsoft

Active learning for anomaly detection in environmental data

Stefania Russo [a,*], Moritz Lürig [a], Wenjin Hao [b], Blake Matthews [a], Kris Villez [a,c]

[a] Eawag, Swiss Federal Institute of Aquatic Science and Technology, 8600, Dübendorf, Switzerland
[b] Institute of Environmental Engineering, ETH Zürich, Switzerland
[c] Oak Ridge National Laboratory, Oak Ridge, TN, 37831, USA

ARTICLE INFO

Keywords:
Active learning
Anomaly detection
Machine learning
Environmental monitoring

ABSTRACT

Due to the growing amount of data from in-situ sensors in environmental monitoring, it becomes necessary to automatically detect anomalous data points. Nowadays, this is mainly performed using supervised machine learning models, which need a fully labelled data set for their training process. However, the process of labelling data is typically cumbersome and, as a result, a hindrance to the adoption of machine learning methods for automated anomaly detection. In this work, we propose to address this challenge by means of active learning. This method consists of querying the domain expert for the labels of only a selected subset of the full data set. We show that this reduces the time and costs associated to labelling while delivering the same or similar anomaly detection performances. Finally, we also show that machine learning models providing a nonlinear classification boundary are to be recommended for anomaly detection in complex environmental data sets.

1. Introduction

Due to the intensified deployment of in-situ sensors for environmental monitoring, experts in environmental science and monitoring are inundated with ever increasing quantities of high-resolution data, whose quality is not guaranteed (Horsburgh et al., 2010; Rieger and Vanrolleghem, 2008). Extensive research has therefore been focused on the design of automated Anomaly Detection (AD) systems, with the aim of automatically identifying unusual patterns in data (Aggarwal, 2015; Hill and Minsker, 2010; Alferes and Vanrolleghem, 2016; Aguado and Rosen, 2008). Compared to manual AD, it is surmised, automatic AD systems can accelerate the otherwise laborious task of visually identifying any outlying data samples in complex data sets.

Machine learning (ML) is a branch of artificial intelligence that focuses on learning from data (Murphy, 2012; Domingos, 2012). ML models provide a valuable set of tools in science and engineering thanks to their ability of extracting meaningful information from data and can be used for automatic AD (Aggarwal, 2015). Active learning (AL) (Aggarwal, 2015; Mussmann and Liang, 2018) is a special case of ML and has been proposed as a way to enable use of supervised learning models while also minimising the burdens associated with human expert labelling. In AL, the domain expert is queried for the labels of a subset of the available data samples. This is an iterative approach, which selects a number of samples that, when paired with the expert label, are expected to improve the supervised learning model the most. At each iteration, the supervised ML model is trained with the data set consisting of labelled data samples (from here on, data records, to distinguish them from unlabelled data samples). This procedure is usually repeated until the model reaches satisfactory performance.

In the literature, AL has been successfully employed in several applications. In the AL challenge (Guyon et al., 2011a, b), several data sets including handwriting and speech recognition, document classification and protein engineering, have been used and tested by different research groups to study the performance of AL. In the results achieved among the almost 300 participants, the teams showed that AL could achieve good performance in terms of classification accuracy with less labelled data. However, AL was also showing lower performance at the beginning of the learning curve, while it was faster and smoother in reaching high levels of accuracy in the second stage, when a higher number of labelled examples was available. In Pimentel et al. (2018), an AL approach for unsupervised anomaly detection was presented and tested on synthetic and real data sets comprising images, medical data for thyroid, and arrhythmia issues identification. In Romero et al. (2018), AL was successfully used for handwritten text recognition. Pelleg and Moore (2005) identify anomalies of special interest by using an AL model in the presence of noisy data. The method was tested over a number of distinct fields at different scales as engineer space shuttle data, abalone biological data and astronomical data from UCI

* Corresponding author.
E-mail address: stefania.russo@eawag.ch (S. Russo).

https://doi.org/10.1016/j.envsoft.2020.104869
Accepted 3 September 2020
Available online 15 September 2020

(University of California, Irvine) data repository.[1] In an intrusion detection application, the labelling time was reduced from two weeks to 1 h, which was as high as 99% (Almgren and Jonsson, 2004). However, quantifying the typical labelling times is generally hard as this depends on the nature of the input data (e.g. images or multivariate timeseries), the expertise of the domain expert, and the design of the user interface.

While the studies cited above show promise, only few of them have explored AD applications, where AL may be more challenging as the whole data set is highly unbalanced, with infrequent anomalies present in only 0.1–10% of the total data. Moreover, no work has been published so far to study the performance of AL in environmental AD applications. While there are no studies that provide empirical evidence for the differences between environmental and other types of data, most of them argue in broad terms regarding its challenges not encountered elsewhere, such as: (a) seasonality at yearly, monthly, weekly, and daily scales, (b) non-stationarity and non-ergodicity, (c) nonlinear system dynamics, (d) a variety of systematic and incipient sensor faults, such as sensor drift (Cherkassky et al., 2006; Hill and Minsker, 2010; Eggimann et al., 2017; Leigh et al., 2019). This makes the application of AD in an environmental setting more challenging as anomalies might not only be few, but also different from each other.

Therefore, the main purpose of this paper is to evaluate whether AL can successfully be used for AD in environmental monitoring. To evaluate the utility of AL, we compare it with conventional supervised learning and random sampling strategy using 5 distinct ML model types. The following models were evaluated based on their off-the-shelf availability in many data science platforms and ease of use: Random Forest, k-Nearest Neighbours classifier, Logistic Regression, Naive Bayes and Artificial Neural Networks. AL is implemented with the uncertainty sampling strategy (Settles, 2010). We discuss how AL can reduce the time spent by the domain expert to label a data set, and how the ML model chosen as a base for the AL strategy can influence the effectiveness of the algorithm.

2. Methods

The first part of this section provides the reader with the necessary information on supervised ML for AD applications. Next, we discuss the most important aspects of the tested ML models. After that, our implementation of AL and the chosen sampling strategies are discussed. We then explain the performance metrics used to evaluate our experimental results. The second part of the section describes the case study and the data used for this work. Finally, we provide a detailed explanation of our experimental procedure and we explain how the methods previously described are implemented in the generation and evaluation of the results.

2.1. Supervised learning

Supervised AD requires a training data set, $\mathscr{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ consisting of N data records where $\mathbf{x}_i \in \mathscr{X}^d$ is a data sample represented as a d-dimensional feature vector and $y_i \in \mathscr{Y}$ is the provided label for it. We refer to the elements of $\mathbf{x}_i$ as features. In AD applications, y_i can only acquire two label values, that is: $\mathscr{Y} = \{0, 1\}$. In AD applications, the normal data is generally named as negative class, with label value 0, and the anomalies are named positive class, with label value 1. A class denotes a set of data having common characteristics.

During training of the model φ, the task is to learn a function, $f : \mathscr{X} \rightarrow \mathscr{Y}$, to predict the most likely label given a new unseen data sample. After training, the model φ is able to compute probabilities $P_\varphi(y = 0|\mathbf{x}^*)$ and $P_\varphi(y = 1|\mathbf{x}^*)$ (between 0 and 1) for a new test data sample $\mathbf{x}^*$, where $P_\varphi(y = 0|\mathbf{x}^*)$ ($P_\varphi(y = 1|\mathbf{x}^*)$) is the predicted chance that a human expert would label the sample $\mathbf{x}^*$ as normal (anomalous). The classifier model then predicts the most likely label: $\widehat{y} = f(\mathbf{x}^*) = \underset{y^*}{\text{argmax}} P_\varphi(y = \mathbf{y}^*|\mathbf{x}^*)$.

Different classification ML models can be used for supervised AD applications. In this work, we have deliberately focused on on-the-shelf softwares and chosen to test 5 ML models from a wide range as they are popular in the ML community and available in every commonly used data science platform (e.g. scikit-learn, Matlab ML Toolbox, Spark MLlib, Weka). Each model can be configured with different hyperparameters. Hyperparameters are defined as those parameters that determine the detailed structure and flexibility of the models. These do not change during model training, as they cannot be learned by the model and are fixed a priori based on prior knowledge, experience, and/or exploratory analysis of the training data. In what follows, we discuss the most important aspects of the chosen models. For a deeper dive into these models, we refer to Murphy (2012); Bishop (2006) and James et al. (2013).

- A Naive Bayes (NB) classifier (Rish et al., 2001) models the distribution of individual classes using Bayes' theorem and predicts the class probabilities for each class. In this work, we use Gaussian NB. Unlike the other models used in this study, the NB model models the joint distribution of both data samples and labels, specifically by assuming (a) that the features are completely independent of each other and (b) that the data samples in each class follow a multivariate normal distribution. A particular advantage of the NB model is that there are no hyperparameters to be determined before training.
- Logistic Regression (LR) (Ng and Jordan, 2002) is the classification equivalent to linear regression. To this end, the predicted probability consists of a linear combination of the input values subsequently transformed non-linearly with the log-sigmoid function. To solve LR's optimization function, different types of solvers are available, here we report the ones that are implemented as libraries in many data science platforms: the "newton-cg" solver (Böhning, 1992); the solver "liblinear" (Fan et al., 2008) which uses a coordinate descent algorithm; and the "sag" solver (Schmidt et al., 2017). The regularisation type, the amount of penalty and solver are all hyperparameters of the LR model (Bishop, 2006).
- k-Nearest Neighbours (kNN) (Liao and Vemuri, 2002), is a non-parametric model which classifies a new data sample by a plurality vote amongst its k closest neighbours. In kNN, the predicted class probabilities are computed as the frequency of data records in the set of selected neighbours that belong to the considered class. To determine what the k closest neighbours to a new data sample are, a distance measure is used. Popular distance measures are: Euclidean, Hamming or Minkowski. In the simplest case, all neighbours are given equal weight. However, one can also discount points that are further away by giving closer neighbours more weight. The main hyperparameters are (a) the number of neighbours k, (b) the chosen distance measure, and (c) the weighting method.
- A Random Forest (RF) (Quinlan, 1987), is an ensemble approach based on decision trees. Decision trees are used repeatedly to split the input data in a top-down approach with the intent to separate samples with different labels from each other. The most important hyperparameters for this classifier are (a) the number of trees in the forest, (b) the optimization function used to measure the quality of a split (e.g. Gini impurity or entropy information gain) and (c) the maximum depth of the trees. The predicted class probability is the fraction of decision trees in the ensemble that predicts the considered class.
- Artificial Neural Network (ANN) classifiers (Dreiseitl and Ohno-Machado, 2002) transform the input data into the predicted class probabilities by means of a complex network of simple yet non-linear unit operations (neurons). Each neuron has its own set of parameters. While ANNs are essentially nonlinear regression models, they are extremely flexible due to the ability to string an almost

[1] https://archive.ics.uci.edu/ml/index.php.

arbitrary number of neurons together, specifically by using large numbers of layers between data samples and predicted labels (hidden layers), each including many neurons. The main hyperparameters for ANN classifiers are (a) the number of hidden layers, (b) the number of neurons in each hidden layer, (c) the type of nonlinearity in each neuron, (d) the solver algorithm used for calibrate the parameters of each neuron (Bishop, 1995), and e) the learning rate for weight updates. Note that the main impact of the learning rate is on convergence speed. While a large learning rate allows the model to learn faster, this is at the cost of arriving on a sub-optimal final set of weights (local minimum). Small values for the learning rate instead can cause small weight changes and slow learning (Attoh-Okine, 1999; Zeiler, 2012). In this work, we have selected a small learning rate. As for the solver, recent research (Nur et al., 2014; Kingma and Ba, 2015) suggests that the chosen optimization algorithm leads to inductive bias, which, in turn, can be interpreted as an implicit type of regularisation or prior.

2.1.1. Additional notes on the chosen models

As mentioned above, the NB model describes the joint density of both data samples and labels. As a result, this allows generation of artificial data records according to the calibrated model. For this reason, it is known as a generative model. In contrast, the LR, kNN, RF, and ANN models do not describe the joint density, only the distribution of the labels conditional to the samples. These models are known as discriminative models. Generative models like NB often outperform discriminative models such as RFs and ANNs on smaller data sets because their generative assumptions prevent overfitting (Ng and Jordan, 2002). Discriminative models, on the other hand, are generally expected to perform better when (a) the assumptions in the generative models are untrue and (b) large and representative data sets are available. ANN models in particular are well-known for their flexibility and outperform other models especially in the large-and-representative data regime (Sarle, 1994). Nevertheless, they need extensive hyperparameter tuning to correctly choose the right architecture (Bardenet et al., 2013). Additionally, ANN models are prone to overfitting. To avoid this issue, in this work we only consider small ANN architectures (low number of hidden layers and neurons in each hidden layer). In contrast, RFs are faster and easier to train as they require fewer hyperparameters to tune. In addition, compared to ANN classifiers, RFs are less prone to overfitting and can learn from smaller data sizes (Liu et al., 2013). LR is a simple discriminative classifier but, as NB, fits the data with low flexibility compared to other methods. This is also because they both present a linear decision boundary between the classes. Finally, kNN is also easy to implement in terms of hyperparameter tuning and training time and, as it is an instance-based learning classifier, it can immediately adapt as new training data comes in. However, it is also known to be sensitive to noisy data and might not perform well on unbalanced data unless the classes are well separated (Cho et al., 1991). Note that we did not consider other well-known models such as fuzzy models or Support Vector Machines (SVMs). As for fuzzy classification models, unlike statistical classification models, they require the assumptions that a single data point can simultaneously belong to multiple classes through use of fuzzy memberships. In this work, we assumed that a data point can belong to one class only, thus leading to the use of models that predict a probability, not a fuzzy membership. As for SVMs, we avoided this model structure as the large size of the kernel matrix prevented an efficient execution of our experiments.

2.1.2. Data pre-processing

Data preprocessing steps such as centring and scaling, are a common practice of data preparation for ML for two main reasons (Kotsiantis et al., 2006). First, these operations can improve the chances of convergence to optimal parameters during model training and the rate of convergence to the final parameters, in turn improving the efficiency of the applied training algorithms. Second, for models based on distance measures, like kNN, centring and scaling can affect the modelled relationship. In this work we have standardised all samples by centring to zero mean and scaling to unit variance, as is common for classification purposes. To this end, the mean m and standard deviation s were always computed for each feature separately and only on the basis of the data records available for training.

2.2. Labelling strategies

In this study, we have applied three methods for labelling: (a) complete labelling, (b) random sampling and (c) AL with uncertainty sampling.

Complete labelling. Conceptually speaking, complete labelling is the most simple method. It assumes that a human expert has manually labelled all available data samples and corresponds to conventional use of supervised learning models.

Labelling based on random sampling (RND). This incremental learning method is initiated by querying the human expert for labels for a small set of data samples. This leads to the production of an initial set of data records, $\mathscr{D}^0$, which is used to train the initial supervised model φ^0. The pool of unlabelled samples is given as $\mathscr{T} = \{\mathbf{t}_i\}_{i=1}^{M}$, where M is the data pool size. A small number of samples are randomly selected from this pool. The domain expert is then asked once more to provide a label y for the selected samples. Following this, the new augmented set of data records, $\mathscr{D}^1 = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N+n}$, is used to re-train the model, obtaining φ^1. The process is repeated for n_{it} number of iterations until all samples have been labelled or the model reaches satisfactory performance. This workflow is shown in Fig. 1. Note that other stop criteria can be used e.g. the data samples for the query can only be obtained at a cost (generally the labelling costs) and the procedure is repeated until the labelling budget is reached (Settles, 2011). To select the samples, in this work we have employed a pseudo-random number generator as a Python module in our code and used the generated numbers to select the indices of the data points to sample in such a way that each sample in the pool is equally likely to be drawn.

Labelling based on active learning using an uncertainty criterion (UNC). AL is also an incremental learning method that differs from RND sampling in how the samples are selected from the unlabelled data pool. After obtaining an initial model (φ^0, as with random sampling), the model is tested on the pool of unlabelled samples ($\mathscr{T}$). The predicted class probabilities ($P_\varphi(y = \mathbf{y}^*|\mathbf{x}^*)$) are then used to select those samples that are expected to improve the model prediction performance the most (following labelling, inclusion in the augmented labelled data set, and model re-training). As with random sampling, the domain expert is then asked to provide a label, y, for the selected samples. The operating assumption underlying AL is that predicted class probabilities carry information about the utility of a label for model training before this label is actually available (i.e., expected utility). AL comes in many variants reflecting the circumstances under which data are produced and how soon after data collection one can expect a domain expert to provide labels. For instance, there are AL methods that query samples by selecting them from on-line signals while being collected simultaneously (Angluin, 1988; Lewis and Gale, 1994; Atlas et al., 1990). In this work we adopt pool-based AL (Lewis and Gale, 1994) where the complete set of (unlabelled) data samples ($\mathbf{x}_i$) is already available before querying starts. This is the most common case in many AD research works (Meng et al., 2013; Almgren and Jonsson, 2004). AL methods can also differ in how the utility of a yet unseen label is estimated (Settles, 2010). In this work, we employ uncertainty sampling (UNC), a selection strategy where the AL model selects the input, ($\mathbf{x}_i$), for which the model's predicted label is most uncertain:

$$i = \underset{i}{\operatorname{argmin}} \left(\max_{\mathbf{y}} P_\varphi(y = \mathbf{y}|\mathbf{x}_i) \right) \quad (1)$$

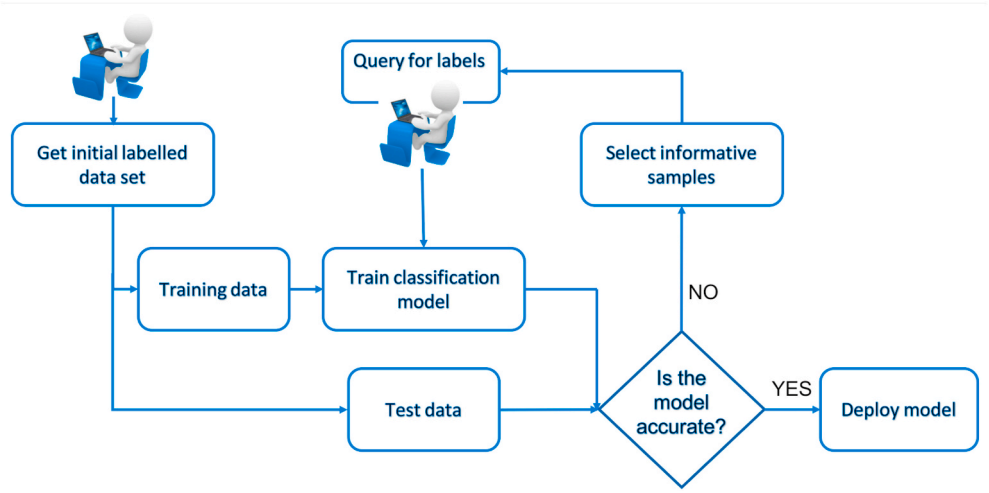


Fig. 1. Active learning workflow. A small labelled training set is used to build an initial classification model, then tested on an unseen pool of data from which informative data samples are selected for querying. Once queried and a label is obtained, the updated set of data records is used to re-train the model. This is repeated until the model reaches satisfactory performance on the test data. This depends on the application and properties of data sets.

where $\max_{\mathbf{y}} P_{\varphi}(y = \mathbf{y}|\mathbf{x}_i))$ is the maximal class probability given $\mathbf{x}_i$. Our implementation of AL is described in Algorithm 1. Note that another sampling method is based on selecting the samples with maximum entropy (Holub et al., 2008), instead of label uncertainty. This however produces the same selection in binary classification problems and is not tested in this work. Another possibility includes using membership query sampling, in which the most informative samples are selected based on classification disagreement between different ML models trained on the same data set. As this involves training multiple models at the same time, the computational costs are too high and unfeasible for the goal of this study.

Algorithm 1. Supervised anomaly detection with active learning

2.3. Performance evaluation

In this work, we are dealing with an unbalanced data set. Therefore, intuitive detection accuracy metrics (e.g. ratio of correctly identified data samples vs total number of data samples) are not recommended as performance metrics (see e.g. Géron, 2019). We will refer as true positives (*tp*) the number of correctly identified anomalies ($y = 1, \hat{y} = 1$). true negatives (*tn*) is the number of correctly classified normal data ($y = 0, \hat{y} = 0$). Finally, false negatives (*fn*) and false positives (*fp*) are respectively the number of incorrectly classified normal data ($y = 0, \hat{y} = 1$) and anomalies ($y = 1, \hat{y} = 0$). Accordingly to the above definitions, the following measures can be computed:

$$precision = \frac{tp}{tp + fp} \qquad recall = \frac{tp}{tp + fn} \tag{2}$$

While *recall* expresses the ability to find all relevant instances in a dataset (what proportion of actual anomalies was identified correctly), *precision* expresses the proportion of the data points the model classifies as anomalies were real (e.g. a model that produces no false positives has a *precision* = 1). Generally, there is an inverse relationship between these measures: as *precision* increases, *recall* decreases. This is called the precision-recall tradeoff. For this reason, a popular score used to measure model performance for unbalanced classification problems is the *F*1 score. This is computed as the geometric mean of precision and recall:

$$F1 = 2\,\frac{precision \cdot recall}{precision + recall} \tag{3}$$

Note that we have intentionally avoided testing for multiple criteria to keep the evaluation simple. However, other performance measures exist and can be also used. One example is the Area Under the Receiver Operating Characteristic (AUC-ROC), which provides an aggregate measure of performance across all possible model classification thresholds.

Finally, to critically evaluate the results and understand the behaviour of the models, we will also investigate the type of data that is queried during the iterations from UNC and RND sampling strategies.

2.4. Case study

Data set The data set used for this study is a multivariate time series data of high spatial and temporal resolution that was collected as part of a long term ecological experiment described in Narwani et al. (2019).

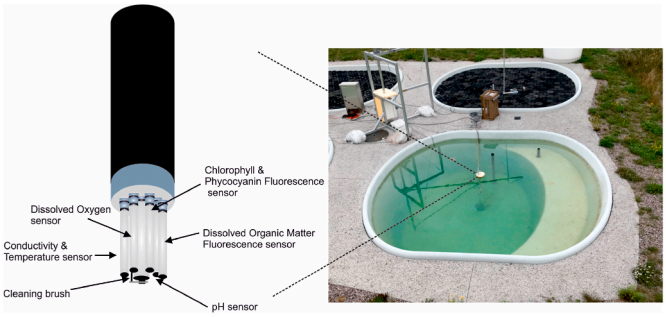
The primary goal of the experiment was to quantify the resilience of aquatic ecosystems in the face of eutrophication using replicated experimental pond ecosystems (hereafter ponds). 16 of such 15,000 L fiberglass ponds (Fig. 2a)) were set up at Eawag Dübendorf (Switzerland) with a layer of gravel and a mix of tap and lake water. Successively, macrophytes (Myriophyllum spicatum) and dreissena mussels (*Dreissena polymorpha*) were added to the ponds and inorganic nutrients were increased progressively as part of a fully factorial design: four randomly chosen ponds received either no species, one of the species, or both species together. In this work, we focused on data from four ponds added with macrophytes as these data were the ones with the most anomalies. Each of these ponds was equipped with a multi-variable instruments (EXO2 Sonde from Xylem Inc.[2]). Each of the 16 instruments contained sensors for eight variables: conductivity, chlorophyll and phycocyanin fluorescence, dissolved organic matter fluorescence, dissolved oxygen (saturation and concentration), pH, and temperature (Fig. 2b). Measurements of these eight water parameters were recorded simultaneously in each multi-sensor, with a fixed time interval of 15 min.

As for the calibration protocol for the instruments, before placing them in the mesocosms, a 48 h cross-comparison trial was performed where the water parameters were measured using all 16 instruments inside a single tank. With this data, initial off-factory differences between the instruments were corrected. Then, the same cross-comparison and a calibration were repeated for two maintenance periods during the experiment.

Data labelling The domain expert manually labelled anomalies for

[2] https://www.ysi.com/exo2/.

(a)



(b)

Fig. 2. Case study. (a) Overview of the experimental ponds at Eawag, Dübendorf (Switzerland); (b) Sensor platform for data collection with 5 installed sensors.

specific conductivity. The labelling was conducted through visual inspection of interactive high-resolution charts that were created in R using ggplot2 (Wickham, 2016), plotly (Sievert, 2018), and htmlwidgets (Vaidyanathan et al., 2018). These interactive graphs were saved in html and opened in a standard internet browser and screened by the domain expert for anomalies. All measured water parameters were plotted in full temporal resolution within a single chart window, 2 weeks at a time. This way, the domain expert could compare all data streams for a given period, and decide whether a segment would qualify as anomalous. After screening the data from the measured variables, the domain expert supplied us with the labels for specific conductivity. The above process resulted in a labelled data set containing $N = 22464$ data records with $d = 8$ dimensions (the measured water quality variables). The data set covers a period of 234 consecutive days and includes 2% anomalies. This was then used to train the chosen ML models, which involves learning (determining) the values for all the model's parameters from the provided labelled data records. Details on the training procedure are described next.

2.5. Experimental procedure

The complete experiment performed in this work consisted of the following steps:

1. Data set splitting. The full data set was split into 80% training data $\mathscr{D}$ and 20% test data in a stratified manner (Domingos, 2012).
2. Hyperparameter tuning. Hyperparameter tuning was based on supervised learning with complete labelling, as described above. During this step, optimal hyperparameters were selected by means of grid search. To this end, model performance were measured by stratified-k-fold-cross-validation of the F1 score, where k is the number of folds. In our case, k = 10 (Refaeilzadeh et al., 2009). This was done by using the training data set only. The implemented hyperparameters for the chosen ML models are reported in Table 1, and the grid search results for each model are shown and discussed in detail in the Supporting Information. Note that we have included the regularisation hyperparameter C for LR it in the grid search, even though the under-parametrised regime we are working in does not require this (Bishop, 2006). The results confirm that LR without regularisation is to be preferred in our case. This is further discussed in the Supporting Information.
3. Best performance. After selection of the optimal hyperparameters, each model was trained with the complete training set and using complete labelling. The F1 score was computed with the test data set to obtain the best performance for each model.
4. Incremental learning. Each model was then tested in combination with the incremental learning methods using the UNC and RND sampling strategies. An initial subset of the training data ($\mathscr{D}^0$) containing 182 data records (0.25% of the training data) was used for both learning methods and all 5 models. Each model was then updated in an iterative manner through one of the incremental learning methods, as described earlier. To this end, $n = 10$ samples were selected for querying at every iteration. This choice is in line with current practice (Smailović et al., 2014; Ramirez-Loaiza et al., 2017; Zhu and Hovy, 2007). For example, in (Zhu et al., 2008) the selection of $n = 10$ samples with uncertainty sampling showed efficient results, as well as in (Brinker, 2003), where the authors proved that smaller batch sizes e.g. 8, 16 have higher learning efficiency. At each iteration, the $F1$ score over the test data was used to measure model performance. The incremental learning strategies were continued until all data samples had been labelled. Considering that the initial set of data records ($\mathscr{D}^0$) can influence the benefit of incremental learning relative to complete labelling as well as the benefit of AL to learning with RND sampling, we repeated the execution of each incremental learning method $R = 10$ times. The initial sets of data records were sampled in a stratified manner and without repetition so that the fraction of anomalies in the initial sets of data records would match the fraction in the complete training data set (2%) and that no data record was used more than once for initialisation. In total, incremental learning was applied 100 times (5 models, 2 incremental learning methods, 10 repetitions).
5. To evaluate the benefits of incremental learning, and AL in particular, the F1 scores on the test data set are reported as a function of the model type, the learning method, and repetitions. In addition, the following summary statistics were computed:
   - Best $F1$: the $F1$ score on the test data set obtained with the complete labelling strategy, for all 5 models.

Table 1
Selected hyperparameters for the chosen ML classification algorithms. In bold, the chosen hyperparameters by means of grid search.

ML Algorithm	hyperparameter
Naive Bayes	None
Logistic Regression	penalty: $\mathbf{l_2}$, l_1
	C: 0.001,0.01,0.1,1,10,**100**: weak/no regularisation[33]
	solver: **liblinear**, newton-cg, sag
kNN Classifier	neighbours: 1, 2, 3, **4**, 5, 6, 7, 8, 9, 10
	weights: uniform, **distance**
Random Forest	criterion: **Gini**, entropy
	max depth:2, 6, 8, **10**, 20
	estimators: 8, 10, **16**, 20, 24, 200
ANN	activation: tanh, **relu**
	hidden layer: **(10,10,10)**, (50,50,50), (50,100,50), (100)
	solver: sgd, **adam**
	alpha: **0.0001**, 0.1, 0.05
	learning rate: **constant**, adaptive

[33]Note that this is the inverse of the regularisation strength as in sklearn implementation. More details can be found in the Supporting Information.

- $F1_{init}$: the mean $F1$ score on the test data set after initialisation of incremental learning, across the $R = 10$ repetitions. This is reported for all 5 models and both incremental learning methods.
- Std_{init}: is the standard deviation of the $F1$ score after initialisation of incremental learning, across the $R = 10$ repetitions. This is reported for all 5 models and both incremental learning methods.
- $F1_{95}$: 95% of the best $F1$ score and is used to compute the amount of labelled data needed to reach this value. This is reported as *Labels* @ $F1_{95}$; note that we report the lowest $F1$ score between the $R = 10$ repetitions.
- Std_{95}: the standard deviation of F1 score between the different data folds when 95% of $F1$ score is reached.
- QAnom: the number of selected anomalies at $F1_{95}$.
- Tot Q: the total number of queries at $F1_{95}$.

3. Results

We discuss the results obtained with complete labelling first. This is followed by a detailed discussion of the results obtained with incremental learning. The code developed for this study and the labelled ecological data needed to reproduce the results reported in this work have been made publicly available at https://doi.org/10.25678/00023Y.

3.1. Best performance with complete labelling

The selected ML models were first trained with the full training data set and then tested on the test data. The resulting best $F1$ scores were computed following Equation (3) on the test set and are reported in Table 2. It is evident that RF, kNN and ANN models produce a superior AD performance compared to LR and NB on the full data set. These results might be explained by the reduced flexibility of LR and NB models. We discuss this further in Section 4.1.0.1.

3.2. Random and Uncertainty Sampling

Evolution of model performance during incremental learning. Fig. 3 shows the average $F1$ score over 10 repetitions as a function of the queried number of samples with the RF model and with both UNC and RND sampling strategies. Qualitatively speaking, it can be seen that the RF model using UNC sampling has reached the best F1 score already after few AL iterations, while the same model needed a greater number of iterations to reach the same value using the RND sampling strategy. Expanding these results to the other models, we plot in Fig. 4 the amount of training data needed to reach $F1_{95}$ for all models and both incremental learning strategies.

As RF, kNN and ANN models produced a high performance with complete labelling, their $F1_{95}$ was higher than LR and NB models. Additionally, it is worth noticing that these models have reached their corresponding $F1_{95}$ with considerable fewer iterations using UNC sampling than using RND sampling strategies. On the other hand, LR and NB models, while presenting lower $F1_{95}$, also did not show significantly difference between the two sampling strategies. These results are reported in Table 2 and further discussed in Section 4.1.0.2.

Going back to Fig. 3, we have plotted the standard deviation of the F1 score across the 10 repetitions, which is shown as a grey area. Obviously, at 100% of training data, the learning curve for both strategies converged to the same value, because we have used the same training set in all cases. Additionally, the initial value of standard deviation was high, meaning that the variability in model performance induced by random initialisation disappeared as more training data were selected. This is true for both incremental learning strategies but appears to occur faster with active learning (UNC). The above results are quantitatively reported in the form of summary statistic in Table 2 for all models and both incremental learning strategies.

Queried samples. As discussed in Section 2.3, UNC sampling selects data samples for which the model is the least confident about its prediction. For RND sampling, these data are randomly selected. For this reason, it is interesting to understand what kind of samples are selected for querying by the human expert. Fig. 5 shows the type of samples that have been queried with UNC and RND sampling for the RF model in the first 50 iterations (corresponding to 0.94% of the training data) for a specific repetition (repetition 1 out of 10). Here, red squares represent anomalies and green squares normal data. The horizontal axis represents the iteration number, and the vertical axis are the $n = 10$ queries for each iteration. It is easy to see that the UNC sampling strategy has selected a considerable amount of anomalies. Based on this result, we speculate that, due to the severe class unbalance, the model was more uncertain about examples of anomalies which were the least seen during training (the initial model φ^0 was trained on only 182 data records, of which 2% are anomalies). For the RND strategy however, where the data samples were selected randomly, the data samples that were queried the most were mainly from the normal class due to their dominant presence in the data set. Fig. 6 shows the same kind of information in a different format: the accumulated number of queried samples that are anomalous are shown as a function of the total number of queried samples, both as a fraction of the number of samples in the training data. It is visible that the RND sampling, starting from the first iteration, selected fewer anomalies than the UNC sampling strategy. The curves' shapes follow a similar trend to the curves in Fig. 3, which might indicate that the selection of anomalies is decisive for improving the models' performance. The results from all the models using UNC and RND sampling strategies are shown in Table 2, where we report the number of selected anomalies QAnom @ $F1_{95}$ and the total number of queries Tot Q @ $F1_{95}$.

Table 2
Obtained results for the selected ML models using UNC and RND sampling strategies.

		Naive Bayes	Logistic Regression	kNN Classifier	Random Forest	Neural Network
Best $F1$		0.74	0.84	0.98	0.97	0.98
$F1_{init}$		0.70	0.77	0.72	0.61	0.75
Std_{init}		0.06	0.05	0.05	0.05	0.09
Std_{95}	RND	0.02	0.03	0.01	0	0
	UNC	0.06	0.03	0.05	0.06	0
Labels	RND	0.33%	1.08%	9.10%	10.95%	9.79%
@ $F1_{95}$	UNC	0.59%	0.48%	1.22%	0.58%	0.48%
QAnom	RND	2	19	156	171	152
@ $F1_{95}$	UNC	18	79	284	106	78
Tot Q	RND	250	170	700	240	170
@ $F1_{95}$	UNC	250	170	240	700	170

4. Discussion

Our results demonstrate that, for AD applications in environmental monitoring, the labelling efforts could be greatly reduced by using an AL strategy, specifically UNC sampling. We discuss below the main findings of this work and its consequences for sensor management in the environmental sector. We then conclude with our practical recommendations for the implementation of AL strategies.

4.1. Summary of results

Best F1 score The experimental results for our case study showed that in terms of best F1 score, RF, kNN and ANN models result in a considerably higher performance compared to LR and NB models. This result might be explained by the reduced flexibility of LR and NB models which present a linear decision boundary between the 2 classes. This jeopardises their learning process, preventing them to clearly separate the anomalous from the normal class. Although further tests are needed to

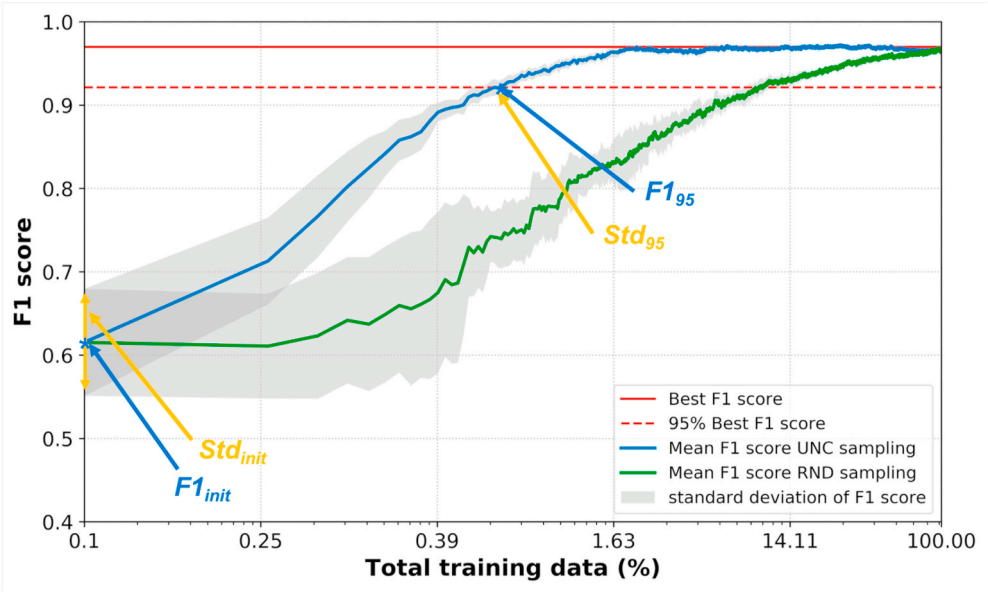


Fig. 3. Results from Random Forest for UNC and RND sampling strategies. The x-axis is a log scale of the percentage of total labelled data. The y-axis represents the model's F1 score on the test data. In figure, the key performance indicators for our experiments are also shown.

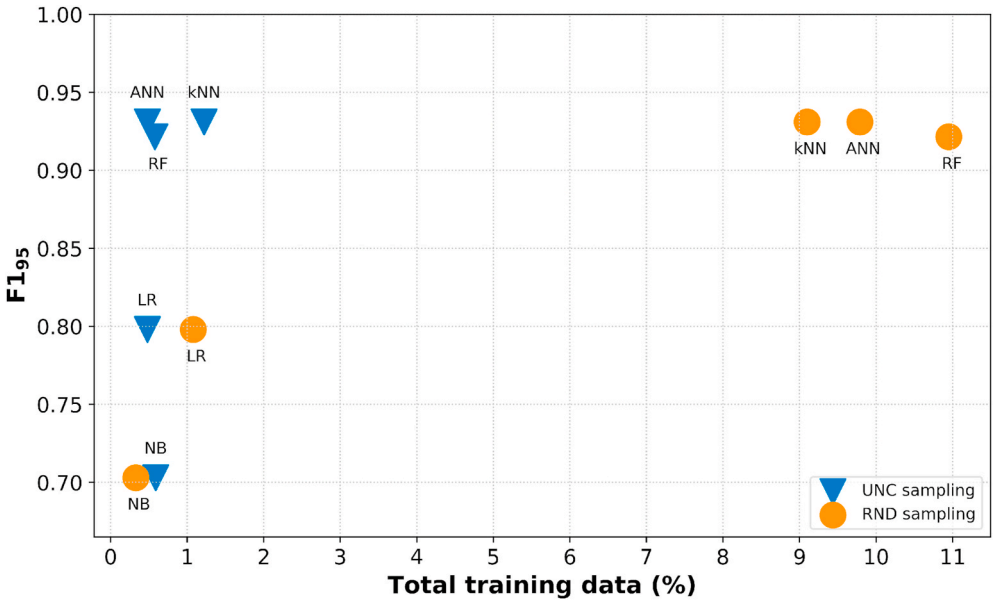


Fig. 4. Total training data necessary to reach each model's corresponding $F1_{95}$ for UNC and RND sampling strategies.

prove this hypothesis, we believe that this is the most important cause for the reduced performance of NB and LR. Since we do not apply regularisation in LR, this can be excluded as a factor for its reduced performance. In addition, specifically for NB, the above results may be explained by the nature of the input data. In environmental applications, normal data might not be generated from the same distribution as it presents baseline changes (due to different seasons), and anomalies may be generated by different events which do not show the same pattern. We suspect that NB model failed at correctly classifying such data because, due to its generative properties, it makes assumptions on the distribution of the data. Lastly, the LR model presented better performance than NB for best $F1$ and for starting $F1$, $F1_{init}$. A reason behind this behaviour is that LR is still a discriminative model. This outcome follows the results presented in (Ng and Jordan, 2002), where it is shown that as the number of data records available for training is increased, LR overtakes the performance of NB because of its discriminative behaviour. This was also articulated by Vapnik (1999): "one should solve the [classification] problem directly [using discriminative models, thus modelling $P(y|x)$] and never solve a more general problem as an intermediate step [thus modelling $P(x|y)$ and $P(y)$]." Whether this holds also for more flexible and nonlinear model structures, of which only discriminative variants (ANN, kNN, RF) were used in this study, remains to be studied.

Random and Uncertainty Sampling NB and LR models however, even if their best F1 score was 0.1–0.2 points lower than the other models, were able to converge to 95% of their best $F1$ only with 0.33–1.08% of RND sampling iterations. This means that there is an important trade-off to be made between two competing and important objectives: model performance and labelling cost. Additionally, as for the RND sampling type the models were initially mainly trained with normal data (because it has

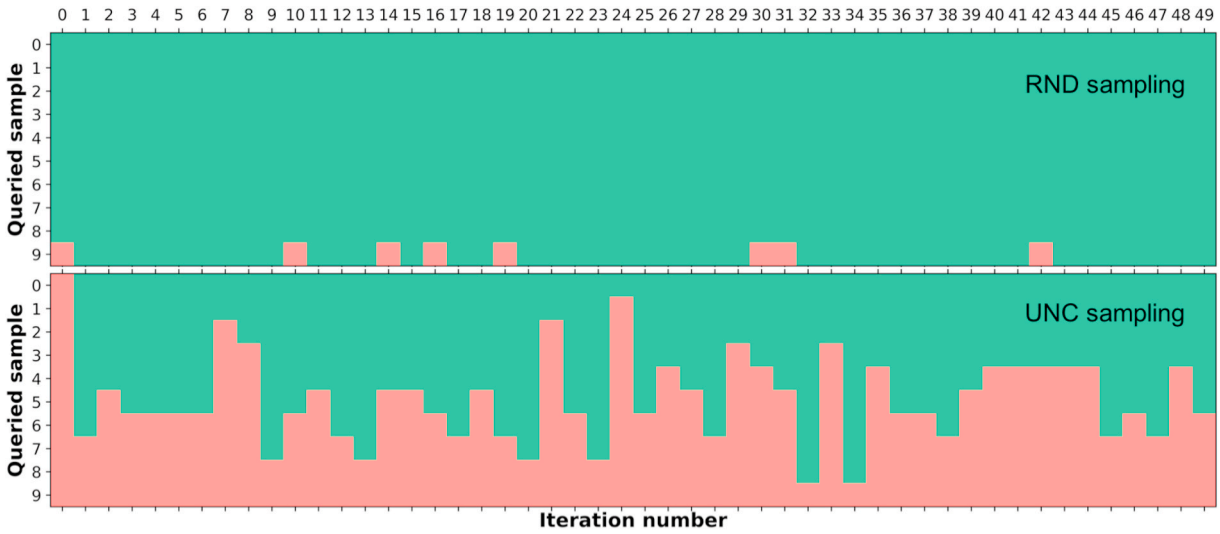


Fig. 5. Selected queries from RND and UNC sampling with RF in 50 iterations, corresponding to 0.94% of labelled data. Red squares represent anomalies and green squares normal data. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

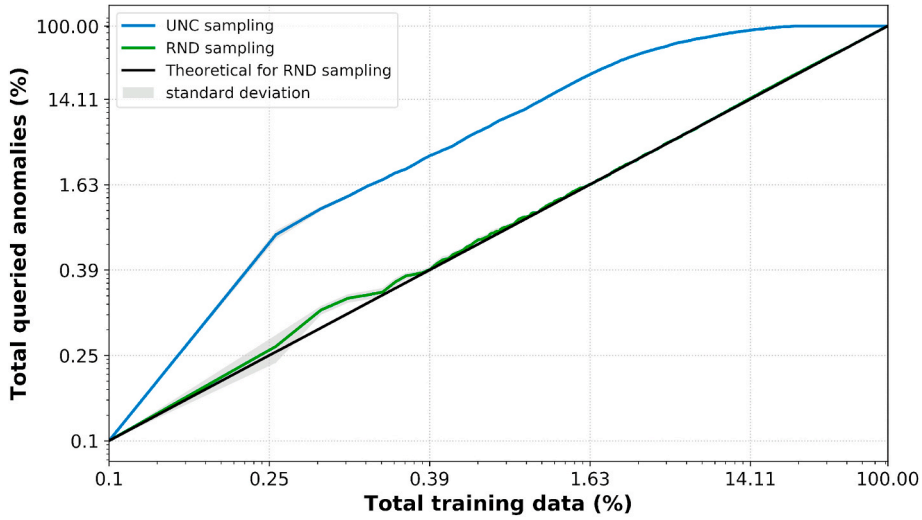


Fig. 6. Selected anomalies from UNC and RND sampling with RF until 100% total available training data is selected. The x-axis is a log scale of the percentage of total labelled data. The y-axis represents the total selected anomalies. In black, we also show the expected accumulated fraction of queried anomalies for RND sampling.

the highest probability of being selected), this suggests that NB and LR models might not need many examples of anomalies during training to learn the best available decision function between the two classes.

As expected, the UNC sampling strategy is much more effective than the RND sampling strategy. It offers better or equal classification performance regardless of the number of samples that have been queried. For these cases in fact, our results show that by applying AL with UNC strategy it is possible to reach a high model performance in just a few iterations. In the best case, the ANN model only needed 0.48% of labelled data to reach 95% of the best F1 performance score. Note also that incremental learning, with either RND or UNC sampling, was always more effective than complete labelling and could potentially save time and costs associated with labelling large data sets.

In environmental applications, anomalies, in addition to being low in number, can be caused by a large variety of disturbances, so ML models cannot easily generalise from them. We have shown in Table 2 that all models with UNC strategy favoured the selection of anomalous data samples for querying, which suggests that ML models tend to be particularly uncertain about the predicted label for anomalous samples relative to normal samples, in turn leading to the selection of anomalous samples with higher frequency. This result corroborates that providing enough and representative data records for model training is one of the main bottlenecks of supervised AD.

4.2. *Consequences for data and sensor management in the environmental sector and future research*

Our results indicate that the amount of labelled data could be greatly reduced by using an incremental learning strategy; this lifts one of the main barriers to the applications of ML techniques in the environmental sector, which is the burden of labelling. In our experiments, during consecutive AL iterations, anomalous data samples are identified as the samples with maximal uncertainty. This suggests that AL is not only useful to reduce the burden associated with labelling for model training, but may also help in identifying anomalous samples as they are added to a data set. Indeed, one could conceive of alerting human operators of the sensor network not only when an input is classified as anomalous but also when the predicted class is uncertain. A similar idea was developed by Giudici et al. (2020) where AL has been used to identify the most informative scenarios in the optimization process for robust planning in

decision making, all while decreasing computational requirements.

The same result could also inspire use of models that only learn from normal data. Potentially, one-class models trained with normal data only could offer a more certain classification for anomalies in the test class. One implementation of AL for AD with one-class models can be found in (Barnabé-Lortie et al., 2015). Note however that one-class models still require an expert-based separation between anomalous and normal data records in the data used for training, thus not eliminating the need for expert-based labelling. This is contrary to frequent claims in the literature on one-class models (Amer et al., 2013; Sabokrou et al., 2018). Another option could be to use a background or garbage class, which can be used to account for the presence of anomalies that are so diverse and rare that their class cannot be learned effectively. For more details on this and related concepts, please see Dhamija et al. (2018). In doing so, the model would only need to represent the normal class, which is likely easier to describe mathematically compared to the anomaly class.

As for the variables used for model training, our domain expert has supplied us with the labels for specific conductivity, and we have trained our models based on this sole information. However, the other variables (chlorophyll and phycocyanin fluorescence, dissolved organic matter fluorescence) may also present anomalies, which could not be easily identified by our domain expert. Note that pH sensor signals are known to be subject to incipient and always-present drift phenomena (Ohmura et al., 2019), which we consider faults but not anomalies. For this reason, we considered removing the pH from the data set variables, but this has not resulted in an improvement of the model performance. While we acknowledge that our choice of using all variables for model was subjective, compared to using the conductivity signal exclusively, the tested models have resulted in an improved performance when provided with all the available variables. Nonetheless, the effects of including these variables, which were possibly contaminated with unlabelled anomalies, on anomaly detection performance should be quantified in future studies e.g. by labelling all anomalies in all sensor signals, or by implementing feature selection if dealing with high-dimensional data sets (Mukherjee and Sharma, 2012; Zargari and Voorhis, 2012).

In this work, we have treated the domain expert as an oracle, i.e. providing perfect and time-invariant labels. However, uncertainty in the labels provided by the domain expert exists (Russo et al., 2019, Villez and Habermacher, 2016) and may be due to: fatigue, learning curve of the human expert, user interface, etc. Methods to account for imperfect oracles exist (Donmez and Carbonell, 2008; Du and Ling, 2010) but they are not commonly studied or tested broadly. For example, Magder and Hughes (1997) discuss that when the degree of uncertainty of a diagnostic test (in our case the labels) is known, this information can be incorporated into the training of LR models, improving their performance. We find these aspects very important for future research. Additionally, we believe it would be beneficial to incorporate some sort of mechanism in AL to obtain additional or revised labels from human experts as a way to gauge temporal, inter- and intra-personal variability in expert opinion.

In this work, and as is typical for AL research, we have quantified the uncertainty associated with individual samples as the uncertainty in the predicted label conditional the most up-to-date model. While this has led to very convincing results, it is very likely that AL could be improved further by accounting for model uncertainty as well. For example, Van Daele et al. (2015) take confidence intervals of the predicted probabilities (James et al., 2013) into account. Another strategy could consist of replacing the mean probability with a distribution, which can be performed for example by implementing the delta and Laplace's methods (Xu and Long, 2005; Tierney and Kadane, 1986). Note that these methods quantify the uncertainty given the currently available information in the data records available for training and the set of unlabelled data samples. A further improvement in the number and/or utility of the queried samples could be expected from quantification of the expected model output change (Cai et al., 2016). In this case, one simulates the effects of obtaining yet unknown labels for an input, thus trying to evaluate the uncertainty in potential future models -not the current one-as a way to select the most informative samples for labelling. This is similar in philosophy to the anticipatory experimental design methods developed by Donckels et al. (2012) and Schwaab et al. (2008) for the purpose of mechanistic model identification. In our opinion, AL strategies, being a special kind of experimental design, could be improved further by borrowing from these and other experimental design methods in the context of mechanistic modelling. Finally, as mentioned earlier, the choice of number of queried samples per iteration needs further study in greater detail, as most of the current practice is based on empirical evidence (Smailović et al., 2014; Ramirez-Loaiza et al., 2017; Zhu and Hovy, 2007).

Another important consideration for future works is the integration on AL with models that incorporate temporal dynamics explicitly. This would include recurrent neural networks or long short-term memory networks, which exist in the family of the deep learning models (Malhotra et al., 2015), but also more conventional, linear models like multivariate ARIMA (Tsitsika et al., 2007; Peter and Silvia, 2012). AL approaches for these models have yet to be well studied.

4.3. Criticism and practical concerns for implementation of active learning strategies

In this work we have used k-fold cross validation on the training data set for hyperparameter tuning on the employed ML classifiers. Note that, for experimental purposes, our hyperparameter selection was performed based on the models' performance on the entire training data set. We acknowledge that in real time applications this could not be implemented beforehand, as hyperparameter selection is too time-intensive to be executed at each iteration of the AL algorithm. The grid search times for each model are shown in the Supporting Information. Therefore, even if our results suggest that ANN models could be more suitable for applying AL for AD in environmental applications, we believe there some aspects of ANN models need to be taken into considerations. For example, we envision that ANN models would be harder to deploy in real time applications of AL because they are more sensitive to hyperparameter tuning, which cannot be performed at the start of an AL procedure due to the lack of data. Furthermore, performing grid search on the wide ANNs' hyperparameter space at each AL iteration is likely too expensive, computationally speaking, for practical applications. However, if running a hyperparameter optimization becomes necessary in real time, one could also think about choosing a different method such as Bayesian optimization (Bergstra et al., 2013), simplex search method (Nelder and Mead, 1965), random search (Bergstra and Bengio, 2012) or direct search (Hooke and Jeeves, 1961). Additionally, the training speed of ANN can be decreased by increasing the learning rate, with the risk of degrading the learning as the model could only arrive to a sub-optimal final set of weights. Training speed remains a significant drawback that should be taken into account when implementing ANNs for AD using AL and should be weighted against the considerable advantages that ANN models bring because of their flexibility, especially when dealing with complex data.

Something else to take into account is that it is advisable to always have the model's predictions compared to a test labelled data set. The reason behind this is twofold: first, it allows to monitor the performance's improvement during the AL iterations, exactly as it was done in this study; second, specifically for environmental applications where data can change over time, it might be beneficial to monitor the model's performance over a long run. However, that also implies that the test data set should be frequently updated with more recent data. Alternatively, it could be advisable to schedule models to retrain at specific times of the year.

Finally, although we have studied the performance of AL for conductivity data collected in an ecological experiment, which might

present some of the broad characteristics of environmental data, further works are needed to generalise the current results to a wider range of environmental monitoring applications.

5. Concluding remarks

The results of this work have indicated that, for anomaly detection applications in environmental monitoring, *i)* active learning could make anomaly detection feasible as it reduces the burden of data labelling by human experts, regardless of model choice; *ii)* flexible model structures like kNN, ANN, and RF are recommended for anomaly detection in complex environmental sets, as opposed to rigid model structures, such as NB and LR, because they lead to higher accuracy, as measured by the F1 score; *iii)* the class prediction of flexible model structures tends to be more uncertain for anomalous data samples. As a result, active learning strategies tend to select anomalous data samples more than normal samples, which in our experiments has resulted in considerable benefits for model identification.

Copyright notice

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to thank Anita Narwani and Piet Spaak for their contributions to the work presented in this paper. The study has been made possible by the Eawag Discretionary Funds (grant number: 5221.00492.012.02, project: DF2018/ADASen).

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.envsoft.2020.104869.

References

Aggarwal, C.C., 2015. Outlier analysis. In: Data Mining. Springer, pp. 237–263.
Aguado, D., Rosen, C., 2008. Multivariate statistical monitoring of continuous wastewater treatment plants. Eng. Appl. Artif. Intell. 21, 1080–1091.
Alferes, J., Vanrolleghem, P.A., 2016. Efficient automated quality assessment: dealing with faulty on-line water quality sensors. AI Commun. 29, 701–709.
Almgren, M., Jonsson, E., 2004. Using active learning in intrusion detection. In: Proceedings of the 17th IEEE Computer Security Foundations Workshop. IEEE, pp. 88–98.
Amer, M., Goldstein, M., Abdennadher, S., 2013. Enhancing one-class support vector machines for unsupervised anomaly detection. In: Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description, pp. 8–15.
Angluin, D., 1988. Queries and concept learning. Mach. Learn. 2, 319–342.
Atlas, L.E., Cohn, D.A., Ladner, R.E., 1990. Training connectionist networks with queries and selective sampling. In: Advances in Neural Information Processing Systems, pp. 566–573.
Attoh-Okine, N.O., 1999. Analysis of learning rate and momentum term in backpropagation neural network algorithm trained to predict pavement performance. Adv. Eng. Software 30, 291–302.
Bardenet, R., Brendel, M., Kégl, B., Sebag, M., 2013. Collaborative hyperparameter tuning. In: International Conference on Machine Learning, pp. 199–207.
Barnabé-Lortie, V., Bellinger, C., Japkowicz, N., 2015. Active learning for one-class classification. In: 14th International Conference on Machine Learning and Applications (ICMLA), pp. 390–395. IEEE.
Bergstra, J., Bengio, Y., 2012. Random search for hyper-parameter optimization. J. Mach. Learn. Res. 13, 281–305.
Bergstra, J., Yamins, D., Cox, D.D., 2013. Hyperopt: a python library for optimizing the hyperparameters of machine learning algorithms. In: Proceedings of the 12th Python in Science Conference, p. 20. Citeseer.
Bishop, C.M., 1995. Neural Networks for Pattern Recognition. Oxford university press.
Bishop, C.M., 2006. Pattern Recognition and Machine Learning. Springer.
Böhning, D., 1992. Multinomial logistic regression algorithm. Ann. Inst. Stat. Math. 44, 197–200.
Brinker, K., 2003. Incorporating diversity in active learning with support vector machines. In: Proceedings of the 20th International Conference on Machine Learning (ICML-03), pp. 59–66.
Cai, W., Zhang, M., Zhang, Y., 2016. Batch mode active learning for regression with expected model change. IEEE Trans. Neural Network. Learn. Syst. 28, 1668–1681.
Cherkassky, V., Krasnopolsky, V., Solomatine, D.P., Valdes, J., 2006. Computational intelligence in earth sciences and environmental applications: issues and challenges. Neural Network. 19, 113–121.
Cho, T.H., Conners, R.W., Araman, P.A., 1991. A comparison of rule-based, k-nearest neighbor, and neural net classifiers for automated industrial inspection. In: Proceedings of the IEEE/ACM International Conference on Developing and Managing Expert System Programs, pp. 202–209.
Dhamija, A.R., Günther, M., Boult, T., 2018. Reducing network agnostophobia. In: Advances in Neural Information Processing Systems, pp. 9157–9168.
Domingos, P.M., 2012. A few useful things to know about machine learning. Commun. ACM 55, 78–87.
Donckels, B.M., De Pauw, D.J., Vanrolleghem, P.A., De Baets, B., 2012. Performance assessment of the anticipatory approach to optimal experimental design for model discrimination. Chemometr. Intell. Lab. Syst. 110, 20–31.
Donmez, P., Carbonell, J.G., 2008. 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.
Dreiseitl, S., Ohno-Machado, L., 2002. Logistic regression and artificial neural network classification models: a methodology review. J. Biomed. Inf. 35, 352–359.
Du, J., Ling, C.X., 2010. Active learning with human-like noisy oracle. In: International Conference on Data Mining. IEEE, pp. 797–802.
Eggimann, S., Mutzner, L., Wani, O., Schneider, M.Y., Spuhler, D., Moy de Vitry, M., Beutler, P., Maurer, M., 2017. The potential of knowing more: a review of data-driven urban water management. Environ. Sci. Technol. 51, 2538–2553.
Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J., 2008. Liblinear: a library for large linear classification. J. Mach. Learn. Res. 9, 1871–1874.
Géron, A., 2019. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media.
Giudici, F., Castelletti, A., Giuliani, M., Maier, H.R., 2020. An active learning approach for identifying the smallest subset of informative scenarios for robust planning under deep uncertainty. Environ. Model. Software 127, 104681.
Guyon, I., Cawley, G., Dror, G., Lemaire, V., 2011a. Datasets of the active learning challenge. J. Mach. Learn. Res.
Guyon, I., Cawley, G.C., Dror, G., Lemaire, V., 2011b. Results of the active learning challenge. In: Active Learning and Experimental Design Workshop in Conjunction with AISTATS 2010, pp. 19–45.
Hill, D.J., Minsker, B.S., 2010. Anomaly detection in streaming environmental sensor data: a data-driven modeling approach. Environ. Model. Software 25, 1014–1022.
Holub, A., Perona, P., Burl, M.C., 2008. Entropy-based active learning for object recognition. In: Computer Society Conference on Computer Vision and Pattern Recognition Workshops. IEEE, pp. 1–8.
Hooke, R., Jeeves, T.A., 1961. Direct search solution of numerical and statistical problems. J. ACM 8, 212–229.
Horsburgh, J.S., Jones, A.S., Stevens, D.K., Tarboton, D.G., Mesner, N.O., 2010. A sensor network for high frequency estimation of water quality constituent fluxes using surrogates. Environ. Model. Software 25, 1031–1044.
James, G., Witten, D., Hastie, T., Tibshirani, R., 2013, ume 112. Springer.
Kingma, Diederik P., Ba, Jimmy, 2015. Adam: A Method for Stochastic Optimization. In: Bengio, Yoshua, LeCun, Yann (Eds.), 3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA. May 7-9. http://arxiv.org/abs/1412.6980.
Kotsiantis, S., Kanellopoulos, D., Pintelas, P., 2006. Data preprocessing for supervised leaning. Int. J. Comput. Sci. 1, 111–117.
Leigh, C., Alsibai, O., Hyndman, R.J., Kandanaarachchi, S., King, O.C., McGree, J.M., Neelamraju, C., Strauss, J., Talagala, P.D., Turner, R.D., et al., 2019. A framework for automated anomaly detection in high frequency water-quality data from in situ sensors. Sci. Total Environ. 664, 885–898.
Lewis, D.D., Gale, W.A., 1994. A sequential algorithm for training text classifiers. In: SIGIR Conference on Research and Development in Information Retrieval. Springer, pp. 3–12.
Liao, Y., Vemuri, V.R., 2002. Use of k-nearest neighbor classifier for intrusion detection. Comput. Secur. 21, 439–448.
Liu, M., Wang, M., Wang, J., Li, D., 2013. Comparison of random forest, support vector machine and back propagation neural network for electronic tongue data classification: application to the recognition of orange beverage and Chinese vinegar. Sensor. Actuator. B Chem. 177, 970–980.

Magder, L.S., Hughes, J.P., 1997. Logistic regression when the outcome is measured with uncertainty. Am. J. Epidemiol. 146, 195–203.
Malhotra, P., Vig, L., Shroff, G., Agarwal, P., 2015. Long short term memory networks for anomaly detection in time series. In: Proceedings. Presses universitaires de Louvain, pp. 89–92.
Meng, Y., et al., 2013. Enhancing false alarm reduction using pool-based active learning in network intrusion detection. In: International Conference on Information Security Practice and Experience. Springer, pp. 1–15.
Mukherjee, S., Sharma, N., 2012. Intrusion detection using naive bayes classifier with feature reduction. Procedia Technol. 4, 119–128.
Murphy, K.P., 2012. Machine Learning: a Probabilistic Perspective. MIT press.
Mussmann, S., Liang, P., 2018. On the Relationship between Data Efficiency and Error for Uncertainty Sampling arXiv preprint arXiv:1806.06123.
Narwani, A., Reyes, M., Pereira, A.L., Penson, H., Dennis, S.R., Derrer, S., Spaak, P., Matthews, B., 2019. Interactive effects of foundation species on ecosystem functioning and stability in response to disturbance. Proc. R. Soc. B 286, 20191857.
Nelder, J.A., Mead, R., 1965. A simplex method for function minimization. Comput. J. 7, 308–313.
Ng, A.Y., Jordan, M.I., 2002. On discriminative vs. generative classifiers: a comparison of logistic regression and naive bayes. In: Advances in Neural Information Processing Systems, pp. 841–848.
Nur, A.S., Radzi, N.H.M., Ibrahim, A.O., 2014. Artificial neural network weight optimization: a review. TELKOMNIKA Indones. J. Electr. Eng. 12, 6897–6902.
Ohmura, K, Thürlimann, C M, Kipf, M, Carbajal, J P, Villez, K, 2019. Characterizing long-term wear and tear of ion-selective pH sensors. Wat Sci Technol 80 (3), 541–550.
Pelleg, D., Moore, A.W., 2005. Active learning for anomaly and rare-category detection. In: Advances in Neural Information Processing Systems, pp. 1073–1080.
Peter, Ď., Silvia, P., 2012. Arima vs. arimax–which approach is better to analyze and forecast macroeconomic time series. In: Proceedings of the 30th International Conference Mathematical Methods in Economics. Karviná, Czech Republic, pp. 11–13.
Pimentel, T., Monteiro, M., Viana, J., Veloso, A., Ziviani, N., 2018. A Generalized Active Learning Approach for Unsupervised Anomaly Detection arXiv preprint arXiv: 1805.09411.
Quinlan, J.R., 1987. Simplifying decision trees. Int. J. Man Mach. Stud. 27, 221–234.
Ramirez-Loaiza, M.E., Sharma, M., Kumar, G., Bilgic, M., 2017. Active learning: an empirical study of common baselines. Data Min. Knowl. Discov. 31, 287–313.
Refaeilzadeh, P., Tang, L., Liu, H., 2009. Cross-validation. Encycl. Database Syst. 5.
Rieger, L., Vanrolleghem, P., 2008. Mon eau: a platform for water quality monitoring networks. Water Sci. Technol. 57, 1079–1086.
Rish, I., et al., 2001. An empirical study of the naive bayes classifier. In: IJCAI Workshop on Empirical Methods in Artificial Intelligence, pp. 41–46.
Romero, V., Sánchez, J.A., Toselli, A.H., 2018. Active learning in handwritten text recognition using the derivational entropy. In: 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 291–296. IEEE.
Russo, S., Disch, A., Blumensaat, F., Villez, K., 2019. Anomaly detection using deep autoencoders for in-situ wastewater systems monitoring data. In: Proceedings of the 10th IWA Symposium on Systems Analysis and Integrated Assessment (Watermatex2019).
Sabokrou, M., Khalooei, M., Fathy, M., Adeli, E., 2018. Adversarially learned one-class classifier for novelty detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3379–3388.
Sarle, W.S., 1994. Neural Networks and Statistical Models.
Schmidt, M., Le Roux, N., Bach, F., 2017. Minimizing finite sums with the stochastic average gradient. Math. Program. 162, 83–112.
Schwaab, M., Monteiro, J.L., Pinto, J.C., 2008. Sequential experimental design for model discrimination: taking into account the posterior covariance matrix of differences between model predictions. Chem. Eng. Sci. 63, 2408–2419.
Settles, B., 2010. Active Learning Literature Survey. Computer Sciences Technical Report. Department of Computer Sciences, University of Wisconsin-Madison.
Settles, B., 2011. From theories to queries: active learning in practice. In: Active Learning and Experimental Design Workshop in Conjunction with AISTATS 2010, pp. 1–18.
Sievert, C., 2018. Plotly for R. R Package Version 4.
Smailović, J., Grčar, M., Lavrač, N., Žnidaršič, M., 2014. Stream-based active learning for sentiment analysis in the financial domain. Inf. Sci. 285, 181–203.
Tierney, L., Kadane, J.B., 1986. Accurate approximations for posterior moments and marginal densities. J. Am. Stat. Assoc. 81, 82–86.
Tsitsika, E.V., Maravelias, C.D., Haralabous, J., 2007. Modeling and forecasting pelagic fish production using univariate and multivariate arima models. Fish. Sci. 73, 979–988.
Vaidyanathan, R., Xie, Y., Allaire, J., Cheng, J., Russell, K., 2018. Htmlwidgets: Html Widgets for R. R Package, version 1.2.
Van Daele, T., Van Hoey, S., Nopens, I., 2015. pyideas: an open source python package for model analysis. In: Computer Aided Chemical Engineering, vol. 37. Elsevier, pp. 569–574.
Vapnik, V.N., 1999. An overview of statistical learning theory. IEEE Trans. Neural Network. 10, 988–999.
Wickham, H., 2016. ggplot2: Elegant Graphics for Data Analysis. Springer.
Xu, J., Long, J.S., 2005. Using the delta method to construct confidence intervals for predicted probabilities, rates, and discrete changes. STATA J.
Zargari, S., Voorhis, D., 2012. Feature selection in the corrected kdd-dataset. In: 2012 Third International Conference on Emerging Intelligent Data and Web Technologies, pp. 174–180. IEEE.
Zeiler, M.D., 2012. Adadelta: an Adaptive Learning Rate Method arXiv preprint arXiv: 1212.5701.
Zhu, J., Hovy, E., 2007. Active learning for word sense disambiguation with methods for addressing the class imbalance problem. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 783–790.
Zhu, J., Wang, H., Yao, T., Tsou, B.K., 2008. Active learning with sampling by uncertainty and density for word sense disambiguation and text classification. In: Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pp. 1137–1144.
Villez, K, Habermacher, J, 2016. Shape anomaly detection for process monitoring of a sequencing batch reactor. Comp Chem Eng 91, 365–379.