Doctoral theses at NTNU, 2024:186

Davide Cacciarelli

Active Learning for Data Streams

Doctoral thesis

NTNU
Norwegian University of Science and Technology
Thesis for the Degree of
Philosophiae Doctor
Faculty of Information Technology and Electrical
Engineering
Department of Mathematical Sciences

NTNU
Norwegian University of
Science and Technology

DTU

Davide Cacciarelli

Active Learning for Data Streams

Thesis for the Degree of Philosophiae Doctor

Trondheim, May 2024

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

NTNU
Norwegian University of
Science and Technology

DTU

NTNU
Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

ISBN 978-82-326-7964-5 (printed ver.)
ISBN 978-82-326-7963-8 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

Doctoral theses at NTNU, 2024:186

Printed by NTNU Grafisk senter

Ph.D. Thesis
Doctor of Philosophy

Active Learning for Data Streams

Davide Cacciarelli

DTU

NTNU
Norwegian University of
Science and Technology

Double Ph.D. Degree program

between

Technical University of Denmark (DTU)
Department of Applied Mathematics and Computer Science

and

Norwegian University of Science and Technology (NTNU)
Department of Mathematical Sciences

Supervisor: Prof. Murat Kulahci (DTU)
Co-supervisor: Prof. John Sølve Tyssedal (NTNU)

DTU Compute
Department of Applied Mathematics and Computer Science
Technical University of Denmark
Building 324
2800 Kongens Lyngby, Denmark

Summary

As businesses increasingly rely on machine learning models to make informed decisions, the ability to develop accurate and reliable models is critical. However, in many industrial contexts, data annotation represents a major bottleneck to the training and deployment of predictive models. This thesis focuses on data-efficient strategies for developing machine learning models in label-scarce settings. The increasing availability of unlabeled data in various applications has led to the need for efficient methods that minimize the cost associated with collecting labeled observations. Traditional active learning approaches, such as pool-based methods, have been extensively studied, but the emergence of data streams has necessitated the development of stream-based active learning strategies able to select the most informative observations from data streams in real time.

The thesis begins with a survey of active learning, providing an overview of recently proposed approaches for selecting informative observations from data streams. It presents the strengths and limitations of the state of the art and discusses the challenges and opportunities that arise in this area of research. Next, the thesis presents a novel stream-based active learning strategy for linear models inspired by the optimal experimental design theory. By setting a threshold on the informativeness of unlabeled data points, the proposed strategy enables the learner to decide in real time whether to label an instance or discard it. Then, the thesis investigates the robustness of online active learning in the presence of outliers and irrelevant features. The thesis also provides initial results related to an adaptive sampling scheme for drifting regression data streams.

Finally, the thesis presents a stream-based active distillation framework for developing lightweight yet powerful object detection models. This approach combines active learning and knowledge distillation, allowing a compact student model to be fine-tuned using pseudo-labels generated by a large pre-trained teacher model.

Overall, this thesis contributes to the field of stream-based active learning by providing insights into various techniques and addressing concerns related to robustness and scalability. The findings expand the potential applications of active learning in real-time data streams and pave the way for more efficient and effective model development.

“Discovery commences with the awareness of **anomaly**, i.e. with the recognition that nature has somehow violated the paradigm-induced expectations that govern normal science. It then continues with a more or less extended **exploration** of the area of anomaly. And it closes only when the paradigm theory has been adjusted so that the anomalous has become the expected.”

Thomas Kuhn, The Structure of Scientific Revolutions

List of publications

Contributions included in the thesis:

1. D. Cacciarelli and M. Kulahci. Sampling strategies for industrial applications through active learning (2023). *Preprint.*
2. D. Cacciarelli and M. Kulahci. Active learning for data streams: a survey (2023). *Machine Learning.*
3. D. Cacciarelli, M. Kulahci and J.S. Tyssedal. Stream-based active learning with linear models (2022). *Knowledge-Based Systems.*
4. D. Cacciarelli, M. Kulahci and J.S. Tyssedal. Robust online active learning (2023). *Quality and Reliability Engineering International.*
5. D. Cacciarelli, J.S. Tyssedal and M. Kulahci. Stream-based active learning for regression with dynamic feature selection (2023). *IEEE International Conference on Transdisciplinary AI.*
6. D. Manjah, D. Cacciarelli, B. Standaert , M. Benkedadra, G. Rotsart, B. Macq and C. De Vleeschouwer. Stream-Based Active Distillation for Scalable Model Deployment (2023). *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops.*

Contributions not included in the thesis:

7. D. Cacciarelli and M. Kulahci. A novel fault detection and diagnosis approach based on orthogonal autoencoders (2022). *Computers & Chemical Engineering.*
8. D. Cacciarelli and M. Kulahci. Hidden dimensions of the data: PCA vs autoencoders (2023). *Quality Engineering.*
9. D. Cacciarelli, M. Kulahci and J.S. Tyssedal. Online Active Learning for Soft Sensor Development using Semi-Supervised Autoencoders (2022). *ICML Workshop on Adaptive Experimental Design and Active Learning in the Real World.*

Work related to these papers was presented at the following conferences:

† indicates the presenting author

1. D. Cacciarelli†, M. Kulahci and J.S. Tyssedal. Real-time sampling strategies for data streams. *INFORMS Annual Meeting – Quality, Statistics and Reliability Best Student Poster Competition*, Phoenix, US, 10/2023.
2. D. Cacciarelli†, M. Kulahci and J.S. Tyssedal. Stream-based active learning for regression with dynamic feature selection. *IEEE Conference on AI for Industries*, Los Angeles, US, 09/2023.
3. S.O.N. Topalian†, H.H. Hansen, D. Cacciarelli and M. Kulahci. A Python library for multivariate statistical process control and dynamic principal component analysis. *ENBIS Annual Meeting*, Valencia, Spain, 09/2023.
4. D. Manjah†, D. Cacciarelli, B. Standaert , M. Benkedadra, G. Rotsart, S. Galland, B. Macq and C. De Vleeschouwer. Stream-based active distillation for scalable model deployment. *CVPR Workshop on Learning with Limited Labelled Data for Image and Video Understanding*, Vancouver, Canada, 06/2023.
5. D. Cacciarelli†, M. Kulahci and J.S. Tyssedal. Invited seminar on active learning. *Baker Hughes*, virtual, 11/2022.
6. D. Cacciarelli†, M. Kulahci and J.S. Tyssedal. Real-time process monitoring with label scarcity. *Danish Data Science Academy*, Billund, Denmark, 11/2022.
7. D. Cacciarelli†, M. Kulahci and J.S. Tyssedal. Stream-based active learning in the presence of outliers. *INFORMS Annual Meeting – Quality, Statistics and Reliability Workshop*, Indianapolis, US, 10/2022.
8. D. Cacciarelli†, M. Kulahci and J.S. Tyssedal. Online active learning with semi-supervised autoencoders. *ICML Workshop on Adaptive Experimental Design and Active Learning in the Real World*, Baltimore, US, 07/2022.
9. D. Cacciarelli†, M. Kulahci and J.S. Tyssedal. A semi-supervised approach to stream-based active learning for industrial processes. *ENBIS Annual Meeting*, Trondheim, Norway, 06/2022.
10. D. Cacciarelli†, M. Kulahci and J.S. Tyssedal. Active Learning: training predictive models with less data. *Math meets Industry*, Trondheim, Norway, 06/2022.
11. D. Cacciarelli† and M. Kulahci. A novel fault detection and diagnosis approach based on orthogonal autoencoders. *ENBIS Annual Meeting*, virtual, 09/2021.

Abbreviations

AI	Artificial intelligence
AP	Average precision
CCTV	Closed-circuit television
CDO	Conditional D-optimality
CNN	Convolutional neural network
DoE	Design of experiments
DPM	Deformable parts models
EWMA	Exponentially weighted moving average
GP	Gaussian process
GPR	Gaussian process regression
HOG	Histogram of orientated gradients
IoU	Intersection over union
KDE	Kernel density estimation
MAB	Multi-armed bandit
mAP	Mean average precision
MCMC	Markov chain Monte Carlo
MH	Metropolis-Hastings
NMS	Non-maximum suppression
OLS	Ordinary least squares
PDF	Probability density function
SE	Squared exponential

SPC	Statistical process control
SSD	Single shot multibox detector
UCL	Upper control limit
UPV	Unscaled prediction variance
YOLO	You only look once

Contents

List of Figures

CHAPTER 1

Introduction

1.1 Label scarcity

In machine learning, obtaining curated and annotated data is essential for the development of accurate and reliable predictive models. However, the annotation process typically represents a significant bottleneck in model training and deployment [1]. For many industrial applications, acquiring labeled observations can be laborious, expensive, and occasionally unattainable, making the limited availability of such data a relevant barrier in training machine learning models suitable for real-world applications. Indeed, despite significant recent advancements, the integration of artificial intelligence (AI) algorithms into real-world applications, such as autonomous vehicles, industrial robotics, and healthcare, continues to present substantial challenges. These challenges largely emanate from the multifaceted nature of the data, which highlights the necessity of machine learning models to be trained on extensive datasets that cover as many scenarios as possible. However, in complex engineering problems, providing such raw data, not to mention annotated data, becomes incredibly difficult. Consider a scenario where our objective is to learn a supervised model $f : \mathcal{X} \to \mathcal{Y}$ using a finite dataset of labeled examples $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$. Here, $\mathbf{x}_i \in \mathcal{X}$ represents a vector of input measurements, and $y_i \in \mathcal{Y}$ is a scalar corresponding to the response value. In this context, label scarcity describes a common dichotomy where obtaining the feature values $(\mathbf{x})$ is relatively easy, but acquiring corresponding labels (y) is challenging, leading to a shortage of labeled data for training supervised learning models. This disparity poses substantial challenges in supervised learning, as the effectiveness of learning algorithms heavily depends on the availability of a set of paired instances $(\mathbf{x}_i, y_i)$ to learn the mapping $f : \mathcal{X} \to \mathcal{Y}$.

The labeling process can be difficult for various reasons. This need for high-quality labeling extends across various domains, each presenting unique challenges:

- *Computer vision.* In computer vision, labeling involves manually annotating images or videos, which can be extremely time-consuming and requires a level of expertise, particularly when dealing with complex scenarios such as identifying defects in industrial components or interpreting medical images [2].

- *Quality control.* In industrial quality control, labeling might involve expert assessment to identify defects or anomalies in products. This process can be highly subjective, leading to inconsistencies in labels, and is often limited by the availability and expertise of human inspectors [3].

- *Design optimization.* In engineering design optimization, labels may represent performance metrics of different design alternatives. Generating these labels often requires extensive simulations or physical testing, which are resource-intensive and time-consuming processes [4].

- *Drug development.* In the context of drug development, labels could be the efficacy or side effects of compounds, which are determined through complex biological experiments. The high cost and ethical considerations involved add additional layers of complexity to the labeling process [5].

- *Clinical trials.* Similarly, in clinical trials, labeling patient data often involves detailed medical diagnosis and monitoring patient responses over time, which not only requires specialized medical knowledge but also faces stringent regulatory and privacy constraints [6].

In all these cases, the scarcity of labels poses a significant challenge, limiting the ability of machine learning models to learn effectively and generalize to new, unseen data. This challenge underlines the need for innovative approaches in data processing and model training that can efficiently leverage limited labeled data, a key area where active learning strategies offer promising solutions.

1.2 Active learning

In recent years, active learning has emerged as a key research area for addressing the challenges posed by label scarcity in machine learning. Active learning is fundamentally driven by the goal of training machine learning models using less labeled data, thereby reducing the need for extensive human supervision [7]. More generally, this paradigm involves a learning algorithm that iteratively queries an oracle to label new data points, aiming to efficiently improve model performance while minimizing the need for extensive labeled data [8]. Within the context of active learning, an oracle is any entity capable of providing an accurate label y_i for an unlabeled data point $\mathbf{x}_i$. In many cases, an oracle can be a human annotator. For example, in computer vision tasks, this role is often filled by a person who manually inspects and labels images. In other cases, an oracle can be a large computational model that provides the outcome of a simulation. As illustrated in Figure 1.1, active learning can be categorized into three primary variants, namely membership query synthesis, pool-based, and stream-based active learning, each distinguished by its approach to data selection and labeling.

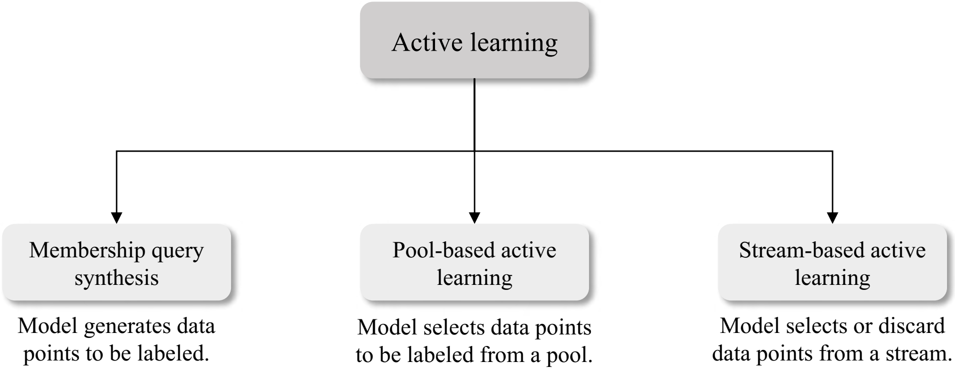

Active learning
Membership query synthesis
Pool-based active learning
Stream-based active learning
Model generates data points to be labeled.
Model selects data points to be labeled from a pool.
Model selects or discard data points from a stream.

Figure 1.1. Active learning scenarios.

1.2.1 Membership query synthesis

Membership query synthesis in active learning allows the algorithm to generate its own queries by creating new data instances, rather than selecting from pre-existing ones. This can be beneficial for investigating specific areas of the input space where available data is sparse, enabling targeted exploration and learning. However, a key limitation of this approach is its potential to produce data points that may be unrealistic or purely hypothetical [8]. Such artificial instances might represent scenarios that are unlikely or impossible to occur in real-world settings. Consequently, human annotators might find it challenging to assign meaningful labels to these contrived data points, as they fall outside the realm of practical or recognizable examples (e.g., a mixture between a letter and a number). For these reasons, this scenario is less explored within the active learning domain, particularly in applied industrial contexts.

1.2.2 Pool-based active learning

Pool-based active learning is the simplest and most extensively studied variant of active learning. In this scenario, we have access to a large pool of unlabeled data, $\mathcal{U} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \subset \mathcal{X}$, and our interest lies in selecting the $b < n$ most informative labels for training a model. Here, b is commonly referred to as the budget. The selection of b instances from $\mathcal{U}$ for training is critical, as it substantially influences the predictive performance of the resulting model [9,10]. To provide a simple example, let us assume we have a folder with thousands of unlabeled images and we are interested in training an object detection model. If we only have a few hours to prepare the data and do not have time to manually label all the images, can we use a smarter strategy than random sampling to select the images that we will need to label for training our model? In this context, pool-based active learning can be defined as a strategy to identify the optimal subset of $\mathcal{U}$ with cardinality b for training our model. The ultimate goal is to enhance model performance while minimizing the number of queries to the oracle, thereby addressing label scarcity and reducing the cost of data annotation. As depicted in Figure 1.2, active learning is not a static process but an iterative one. It involves using a selection rule to guide the labeling process, with this rule often being refined and updated as new labeled data becomes available to the learning system. This dynamic nature of active learning ensures that the model is continually updated with the most informative data, thereby enhancing its learning efficiency and effectiveness.

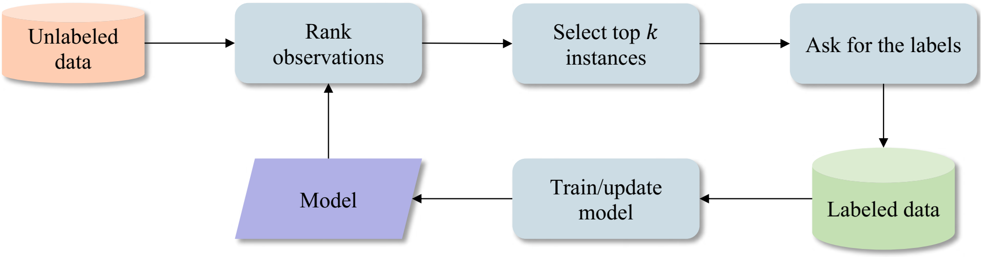


Figure 1.2. The pool-based active learning framework (from PAPER 2), where we prioritize the labeling of the most informative observations. This iterative process usually continues until a budget constraint for label acquisition is met, thereby optimizing learning efficiency.

In PAPER 1 (Appendix A), we introduce the key active learning methods in the pool-based setting and showcase the potential benefits of active learning through an industrial case study.

1.2.3 Stream-based active learning

Stream-based active learning is a variant of active learning in which a learning model sequentially receives unlabeled data points $\mathbf{x}_i$ from a continuous stream $\mathcal{S} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots\} \subset \mathcal{X}$. The model must evaluate the informativeness of each incoming instance on the fly. If an observation $\mathbf{x}_i$ is deemed informative, its corresponding label y_i is queried, and the model is subsequently updated. Stream-based active learning extends the active learning paradigm to scenarios where data arrives sequentially, such as in data streams [11]. The key difficulty in this scenario is that the learner faces the challenge of making real-time decisions about which instances to label as they arrive in a stream. This requires efficient and effective sampling strategies that can cope with the dynamic nature of data streams, enabling the learner to make informed decisions on the fly. The stream-based active learning framework is illustrated in Figure 1.3.

To facilitate the understanding of this framework, we hereby provide two practical examples. The first example is a statistical riddle commonly referred to as the

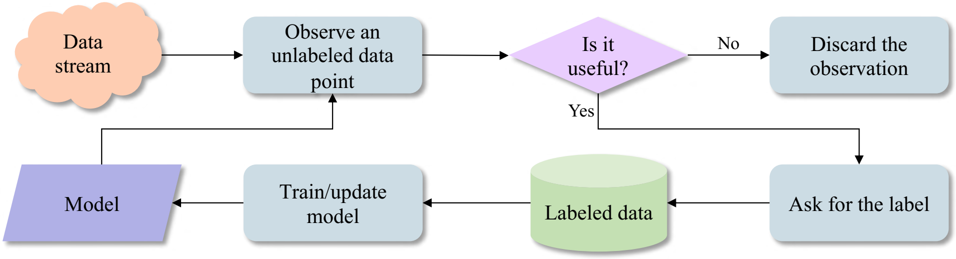


Figure 1.3. Stream-based active learning framework (from Paper 2), where observations are sequentially evaluated as they arrive in a stream.

secretary problem, also known as the optimal stopping theory [12]. This problem is depicted in Figure 1.4, which illustrates the key steps of the decision-making process. The secretary problem is a classic exercise in probability and decision theory that involves selecting the best candidate for a position from a sequence of applicants. Each candidate must be either hired or rejected immediately after their interview, with no option to revisit previous candidates. The secretary problem metaphorically represents the challenges faced in scenarios where decisions must be made sequentially and irrevocably, often with incomplete information. In the context of stream-based active learning, the decision-making process in the secretary problem parallels the determination of whether to label a data point and include it in the model, based on its perceived informativeness. Here, the interview process can be analogized to the evaluation of an unlabeled information criterion for each observation in the stream, guiding the decision on whether the costly label should be requested or not.

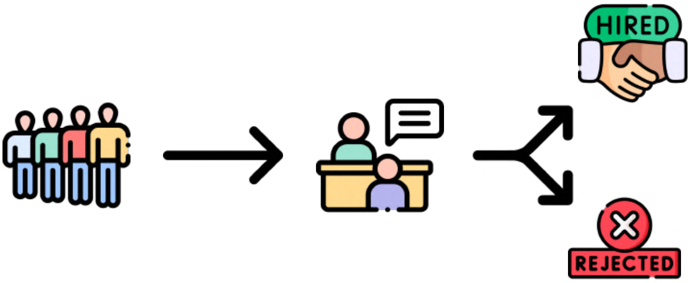


Figure 1.4. Illustration of the secretary problem, highlighting the immediate decision-making process after each interview (individual icons composing the flowchart are downloaded from [13]).

An additional example to motivate the relevance of stream-based active learning in real-world scenarios comes from industrial manufacturing, as shown in Figure 1.5.

Manufacturers often rely on predictive models and soft sensors to estimate measurements that are challenging to acquire directly [14–16]. However, the crux of employing these soft sensors lies in the training of the underlying models, which necessitates real-world labeled data [17, 18]. In a typical production scenario, where components are produced continuously, there might be a narrow window of opportunity to decide whether a particular item should undergo further inspection to acquire a label. This decision is crucial for updating the predictive model and must be made promptly before the component progresses further in the manufacturing process, such as being added to a work-in-progress inventory or undergoing physical alteration.

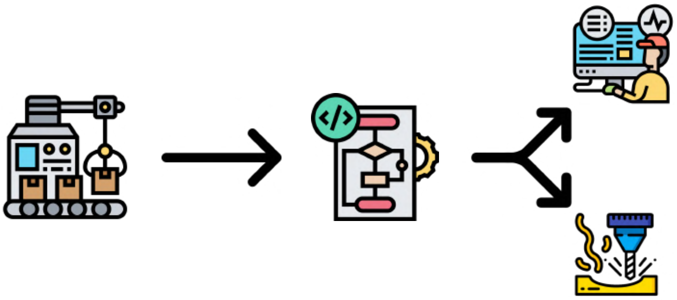


Figure 1.5. Illustration of a real-time sampling problem in industrial production, where the process involves deciding on the fly whether to perform a quality inspection on a product or forward it to the subsequent workstation or machinery (individual icons composing the flowchart are downloaded from [13]).

In Paper 2 (Appendix B), we provide a comprehensive survey on active learning approaches for data streams, highlighting key concepts like:

- *Active learning scenarios.* We explain the key characteristics of the three active learning scenarios: membership query synthesis, pool-based active learning, and stream-based active learning.

- *Instance selection criteria.* We list the main approaches used to evaluate the informativeness of the unlabeled data points: uncertainty-based query strategies, expected error or variance minimization, expected model change maximization, disagreement-based query strategies, diversity- and density-based approaches, and hybrid strategies.

- *Application areas.* We provide a description of potential application areas: chemical or manufacturing processes, video streaming, clinical trials, fraud detection, and online customer service.

- *Data drifts.* We describe the different types of drifts that can affect the data stream: abrupt drift, gradual drift, incremental drift, and recurring concepts.

- *Taxonomy of stream-based active learning methods.* We classify the state-of-the-art approaches into four categories: stationary data stream classification approaches, drifting data stream classification approaches, evolving fuzzy system approaches, and experimental design and bandit approaches.

- *Evaluation strategies.* We explain the differences between the two main evaluation approaches: holdout test set and prequential evaluation (test-then-train).

- *Challenges.* We highlight the key challenges and opportunities for future improvements in this research area: algorithm scalability, labeling quality, distribution shift, model interpretability, real-life assessment, and human-computer interaction.

It is important to note that within Paper 2, the term online active learning is frequently used interchangeably with stream-based active learning. Both terms refer to the challenge of selecting the most informative observations from a stream in real-time. However, this should not be confused with online learning, a related but distinct research area. Online learning is an approach where models are incrementally updated with the continuous influx of new data. It aims to make accurate predictions in a sequence, based on previous outcomes and any additional available information. Online learning spans various research fields including game theory, information theory, and machine learning [19, 20]. This methodology is essential in scenarios where processing the entire dataset at once is impractical or impossible, often due to data volume or the dynamic nature of data generation. Online learning algorithms are designed to adapt their parameters incrementally, making them ideal for real-time applications and environments characterized by evolving data distributions.

1.3 Data-centric AI

Active learning is an approach to data collection and experimentation that highlights the importance of carefully selecting relevant observations for training machine learning models. In a broader landscape, even when data collection is feasible, various inherent biases and anomalies can permeate the process, potentially leading to biased or inaccurate predictive models. These challenges form the foundation of the recent shift from model-centric to data-centric AI [21]. For many years, a model-centric approach dominated the AI field. Model-centric AI places a stronger emphasis on the design, architecture, and optimization of AI models themselves. It focuses on developing sophisticated algorithms and architectures that can learn from the available data. Model-centric AI often involves engineering complex neural network architectures, fine-tuning hyperparameters, and optimizing the model structure and parameters to achieve high accuracy and performance. While model-centric AI acknowledges the importance of data, it tends to assume that the data is readily available and of sufficient quality. Conversely, data-centric AI is a paradigm that places a significant emphasis on data quality over mere quantity, advocating for the careful collection, annotation, and management of datasets. This perspective recognizes that while a larger volume of data may provide a broader context, the accuracy and relevance of the data for the task at hand are crucial. The juxtaposition between model-centric and data-centric AI highlights the different perspectives and priorities within the AI workflow. Both approaches are valuable and can be complementary, with data-centric AI providing the foundation for effective model training and model-centric AI enhancing the capabilities of the models.

Essentially, data-centric AI is a machine learning approach where the emphasis is placed on enhancing the quality, structure, and consistency of the dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$. This paradigm asserts that the performance of learning algorithms, which approximate the function $f : \mathcal{X} \to \mathcal{Y}$, is contingent not only on the algorithmic sophistication but critically on the integrity and representativeness of $\mathcal{D}$. The key elements of the data-centric approach related to developing and maintaining reliable training data include:

1. *Data collection and integration.* Gathering and combining relevant data from various sources, such as sensors, databases, social media, or other platforms [22].

2. *Data labeling and annotation.* Assigning annotations to the data, a task which is often performed by human experts, to provide ground truth labels for supervised learning tasks [23].

3. *Data preprocessing and feature extraction.* Cleaning, transforming, and organizing data in a format suitable for the analysis. This step often involves tasks like data normalization, feature engineering, and handling missing values [24]. At this stage, statistical techniques or machine learning models can also be used to extract salient features from the data.

4. *Data reduction and augmentation.* Undersampling techniques can be used to improve the performance on underrepresented populations [25], or to cope with class-imbalance [26]. Data augmentation can be used either to increase the training set size by creating variants of the same data point [27], or to generate observations that are relevant but rarely encountered in real life (e.g., low-probability high-risk events) [28]. Finally, data reduction can also be seen from a feature perspective. That is, column-wise data reduction rather than row-wise, in the case of a data matrix.

5. *Learning and feedback loop.* Adopting mechanisms to continuously update and improve AI models as new data becomes available. This involves monitoring the performance of the model and collecting new observations to adapt to evolving and drifting data distributions [29].

1.4 Contributions

The key objective of this thesis is to investigate the development of predictive models in label-scarce environments through the use of active learning-based sampling strategies. The main contributions of this thesis can be summarized as follows:

- We present an introductory paper on active learning, highlighting practical aspects of sampling strategies and their potential benefits, illustrated through a case study using real-world data.

  ⟶ Paper 1 – *Sampling strategies for industrial applications through active learning* (Appendix A).

- We provide a comprehensive survey on active learning for data streams, which will be beneficial to researchers and practitioners interested in the development and application of online active learning. The survey adopts a pedagogical approach to make the field accessible to those seeking to learn the basics of this research area in a comprehensible manner. In addition to analyzing sampling strategies in detail, we also elucidate related methodologies, offering a broader perspective of the field.

  ⟶ Paper 2 – *Active learning for data streams: a survey* (Appendix B).

- We propose a novel stream-based active learning approach for linear regression models, based on optimal experimental design theory. This approach is based on the connection between conditional D-optimality (CDO) and unscaled prediction variance (UPV). We demonstrate how this methodology can be used to develop accurate regression models under a limited labeling budget.

  ⟶ Paper 3 – *Stream-based active learning with linear models* (Appendix C).

- We investigate how the presence of outliers affects the performance of the proposed stream-based active learning strategy (and related benchmark strategies) and propose a two-fold solution to mitigate their impact on the predictive models. The solution includes the use of robust estimators and a double-threshold approach to bound the search area of the active learning algorithm.

  ⟶ Paper 4 – *Robust online active learning* (Appendix D).

- We conduct an initial analysis of the impact of irrelevant features on the learning efficiency of stream-based active learning strategies for linear regression models. We explain how incorporating a feature selection step can enhance the estimation process, especially when the learner has access to only a small number of observations.

  ⟶ Paper 5 – *Stream-based active learning for regression with dynamic feature selection* (Appendix E).

- We propose a residual-based sampling strategy to identify localized concept drifts. This strategy is likely to expedite the monitoring of concept drift, facilitating faster discovery and more prompt model updates.
- We introduce the stream-based active distillation framework, wherein specialized lightweight student models are fine-tuned using pseudo-labels from a large pre-trained teacher model. Our analysis reveals that careful selection of the frames used for fine-tuning leads to more efficient training.

  $\longrightarrow$ Paper 6 – *Stream-based active distillation for scalable model deployment* (Appendix F).

1.5 Organization

Chapter 1 serves as an introduction to active learning, providing basic definitions and explaining the critical issue of label scarcity in machine learning. Following this, Chapter 2 delves into the broader research field. It offers a comprehensive description of alternative methodologies in industrial statistics and machine learning that parallel or complement active learning. In Chapter 3, we present the conditional D-optimality stream-based active learning method, specifically tailored for linear models. This chapter also encompasses related works that address the challenges posed by outliers and irrelevant features. Chapter 4 is dedicated to exploring the dynamics of concept drift in data streams. We examine how shifts in data distribution can impact the effectiveness of learning strategies and propose methods to adaptively respond to such changes. Finally, in Chapter 5, we provide a brief overview of object detection models and present the stream-based active distillation framework.

1.6 Notation

Whenever possible, lower-case italic letters are used for scalars (e.g., x), lower-case bold letters indicate vectors (e.g., $\mathbf{x}$), and upper-case bold letters indicate matrices (e.g., $\mathbf{X}$).

CHAPTER 2

Related research areas

This thesis navigates the intersection of industrial statistics and machine learning, bringing together insights and methodologies from both fields. In this chapter, we offer a general overview of alternative approaches pertinent to sampling and label scarcity, prevalent issues in these broad disciplines. We categorize these approaches into two groups. The first group presents methodologies (mostly from industrial statistics) related to sampling and monitoring, which can be used to drive data collection schemes in industrial settings. The second group describes training approaches for machine learning models that can facilitate model development in environments characterized by label scarcity. Our aim is to contextualize our research within the wider landscape of these fields, highlighting the connections and distinctions that characterize our study. For a comprehensive literature review on the core topic of the thesis, stream-based active learning, readers may consult Paper 2 in Appendix B.

2.1 Sampling and monitoring methods

In industrial statistics, a wide range of statistical methods have been historically developed to assist practitioners in complex tasks such as sampling and data collection. This section will provide an overview of several sampling-related research areas that have been pivotal in industrial applications. These methodologies not only form the foundation for modern statistical practices but also offer insights into the evolution of data collection techniques in response to industrial challenges. By providing a framework for efficient and effective data collection and experimentation, they ensure quality and reliability in diverse industrial processes. Furthermore, the principles and methodologies of these traditional strategies continue to influence contemporary statistical techniques, including the growing field of active learning in machine learning. Thus, this exploration serves not only as a retrospective analysis but also as a bridge to understanding the transition to and relevance of newer methodologies like active learning.

2.1.1 Design of experiments

Experimentation is at the core of scientific discovery. In industrial statistics, design of experiments (DoE) represents a systematic approach to the planning of experiments in order to support process understanding and optimization. It involves planning the experiments in such a way that appropriate data is collected, which can then be analyzed to yield valid and objective conclusions. In most of the cases, DoE is used to analyze the relationship between the input and output factors of a process. In general, various tests are performed to see the effect of varying levels of input factors on the response. For example, in a chemical process, we might be interested in designing an experiment to understand the effect of temperature and pressure on chemical yield. By systematically varying these factors, researchers can identify optimal operating conditions and interactions between variables. The critical aspect of DoE is the way the factor levels are varied throughout the experiment.

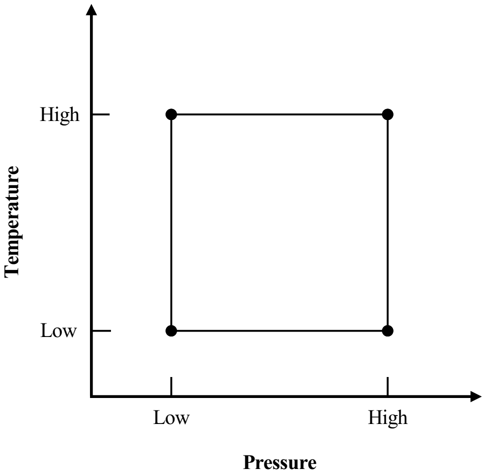


Figure 2.1. A two-factor factorial experiment for a chemical process, where we are interested in observing the effect of pressure and temperature on the yield.

One of the most common types of design is the factorial design, where in each complete trial or replicate of the experiment all possible combinations of the levels of the factors are investigated [30]. Figure 2.1 shows a factorial experiments where both temperature and pressure can be set at two levels. The black dots represent the points where the tests are performed, namely where the response is measured. Then, the effect of a factor is defined as the change in response produced by a change in the level of the factor. Assuming linearity in the factor effects within the design space, an ordinary least square (OLS) regression model is usually fit on the experimental data. The regression model allows to analyze the response surface and identify process optimization directions.

In factorial designs, the property of orthogonality plays a pivotal role in enhancing the effectiveness and clarity of the experimental analysis. Orthogonality in this context means that the effects of any factor, or combination of factors, can be estimated independently of other factors. This independence is crucial as it allows for the isolation of each factor's impact on the response variable, free from the influence of other factors' levels. A key benefit of this independence is the prevention of confounding, ensuring that the main effects and interaction effects are distinct and not mixed with each other. Consequently, the statistical analysis, particularly techniques like analysis of variance, becomes more straightforward and effective due to the ability to calculate the sum of squares for each factor independently. Additionally, orthogonal designs ensure balanced comparisons between factor levels, meaning each level of a factor appears equally with each level of every other factor, facilitating fair and unbiased estimates of effects. This balance and independence in factorial designs not only maximize the information obtained from experimental data but also significantly reduce experimental error, thereby enhancing the precision of effect estimation. Hence, the orthogonality of factorial designs is a fundamental attribute that contributes substantially to their efficacy in experimental research, especially when investigating the simultaneous effects of multiple factors.

In traditional settings, DoE primarily aids practitioners in planning experiments within a controllable, offline environment. Here, factors can be deliberately manipulated at different levels to observe the corresponding responses. In contrast, machine learning often deals with observational data, where control over factor levels is not allowed. Despite this, the principles of DoE can be highly relevant and beneficial in assessing the quality of training data in machine learning applications [31]. Crucially, the concepts of design optimality, foundational to DoE, can be adeptly applied to evaluate observational data. Optimal experimental design is a research methodology that closely aligns with the objectives of active learning, where the focus is on estimating accurate and unbiased models while minimizing the number of experimental runs [32]. By leveraging design optimality principles, one can effectively choose data points that enhance the training process of a regression model. A deeper discussion on the conjunction between optimal experimental design and stream-based active learning is presented in Paper 2 (Appendix B).

2.1.2 Adaptive sampling

Adaptive sampling represents an interesting research area within industrial statistics that is closely linked to data collection methodologies and active learning principles. It is a dynamic strategy where the sampling process is adjusted in real-time, based on insights gained from previously collected data. This method is particularly effective in environments characterized by heterogeneous populations, where certain subgroups warrant more focused investigation. For instance, consider the application of adaptive sampling in environmental monitoring, specifically in the assessment of water quality in a lake. Imagine an initial survey reveals higher pollution levels in a certain section of the lake. Subsequent sampling efforts can then be strategically concentrated in this area. This approach not only optimizes resource utilization but also ensures more detailed and relevant data collection. Figure 2.2 depicts the scenario where the lake is explored with initial sampling points distributed uniformly across its surface. The plot illustrates a region with higher pollution levels, where additional sampling points should be placed. As the sampling progresses, the visual representation would typically use distinct colors or symbols to differentiate between the initial uniform sampling points and the later, concentrated points, clearly demonstrating the adaptive nature of the sampling strategy.

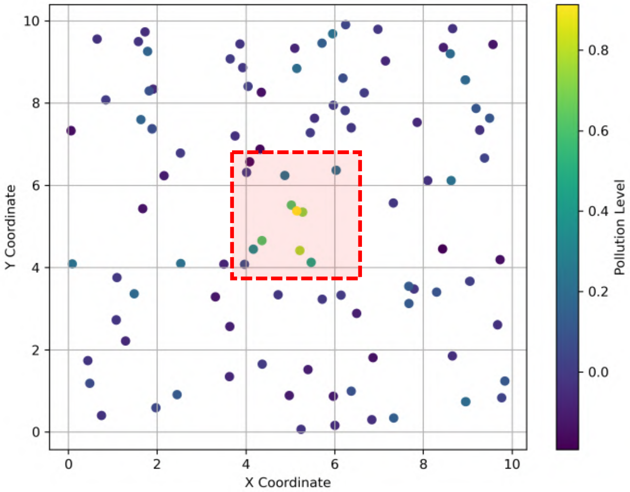


Figure 2.2. A fictional example of adaptive sampling for environmental monitoring. After a passive uniform sampling phase, a region of higher pollution (shaded in red) is identified for further investigation.

Recently, adaptive sampling has extended its utility beyond traditional applications, embracing the complexities of modern data-rich environments [33, 34]. This evolution of adaptive sampling closely aligns with the principles of stream-based active learning, particularly in its ability to discern and prioritize valuable information from vast data streams for efficient process monitoring. For example, adaptive sampling can be employed when a stream of images is sequentially observed, and the goal is to detect alterations within these images while adhering to a budget on the number of pixels that can be investigated [35]. Here, the budgeted sampling highlights a common aspect of stream-based active learning. However, while active learning primarily focuses on model improvement, adaptive sampling, even when applied to data streams, is predominantly concerned with sampling for process monitoring and anomaly detection.

2.1.3 Acceptance sampling

Acceptance sampling, a foundational technique in statistical quality control, is utilized to decide whether to accept or reject a lot of products based on the inspection of a sample from that lot. This method is deeply rooted in the history of quality assurance, tracing back to the early 20th century [36]. It represented a shift in focus towards inspection and decision-making regarding the quality of products, a cornerstone of quality assurance practices. To give a simple example, consider the case of an electronic components manufacturer, who needs to decide whether to accept or not a batch of incoming raw material. Using acceptance sampling, a random sample of the raw material is tested, and if the number of defective items is below a pre-determined threshold, the entire batch is accepted. This decision is usually referred to as lot sentencing. Lot inspection does not only refer to incoming material from suppliers but it is often performed at various stages of the production process to ensure quality and reliability. When compared with 100% inspection, acceptance sampling offers many advantages [36]. First of all, it is a less expensive practice due to reduced inspection efforts. Moreover, the decreased sampling also minimizes the risk of product damage within the lot, allowing the use of destructive testing (since we do not need to test all products). Another key benefit is that the rejection of entire lots based on a few samples might provide a stronger incentive for suppliers to enhance quality. However, acceptance sampling carries inherent risks, such as the potential acceptance of substandard lots and rejection of satisfactory ones. The method generates less information about the product and its manufacturing process compared to complete inspection. Additionally, it requires meticulous planning and documentation of the sampling procedure, which is not a necessity in 100% inspection.

In general, we can distinguish between two types of acceptance sampling plans:

1. *Single sampling plans.* In this case, the decision about the lot is taken after the inspection of n items randomly sampled from the lot. Random sampling is usually employed to avoid introducing biases into the process.

2. *Multiple sampling plans.* After an initial random sample, we can decide whether to accept the lot, reject the lot, or take another sample before achieving lot sentencing.

Sequential sampling plans extend the concept of multiple sampling plans, offering a more advanced and dynamic approach to acceptance sampling. Unlike single or multiple sampling plans, which use a predetermined sample size, sequential sampling allows for decisions to accept, reject, or continue sampling based on each item inspected. This method is especially advantageous when sampling costs are high or when rapid decision-making is essential. In this regard, sequential sampling shares similarities with the stream-based active learning framework and the secretary problem, as it requires immediate decisions after observing each sample. Furthermore, just as stream-based active learning is categorized into single-pass and batch-based methods [37], multiple sampling plans can be divided into item-by-item and group sequential sampling [36]. However, despite these similarities, it is important to note that sequential sampling and stream-based active learning are fundamentally different. Active learning involves sequential evaluation of data points to determine their inclusion in a predictive model. In contrast, item-by-item sequential sampling is focused on iteratively assessing whether a batch of products meets conformity standards.

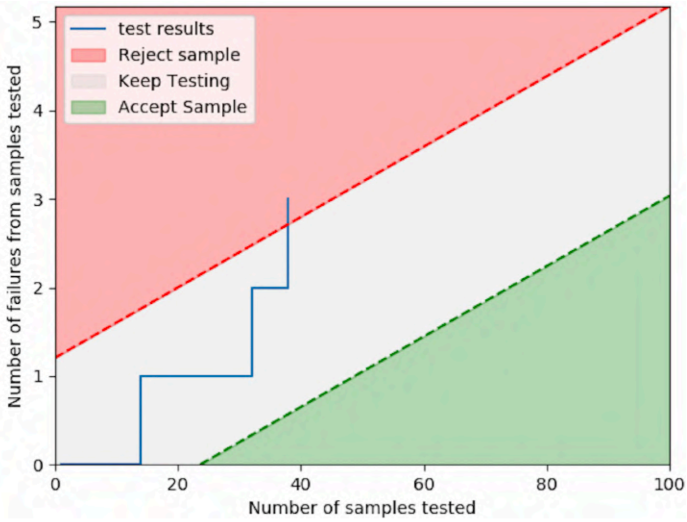


Figure 2.3. Example of a sequential sampling plan from [38], based on the data from the Engineering Statistics Handbook published by the National Institute of Standards and Technology [39].

Figure 2.3 shows the typical sequential sampling test, which is based on the sequential probability ratio test proposed by Wald [40]. In the plot, the test results show the cumulative number of defective items observed against the total number of items inspected. We can see how two parallel decision lines guide the acceptance or rejection of a lot. If the plot of cumulative defectives against items inspected falls within these lines, the inspection process continues with the next item. The lot is rejected if a point falls on or above the upper line (rejection region) and accepted if it falls on or below the lower line (acceptance region). The process of the item-by-item sequential sampling plan may continue potentially until the entire lot is inspected. However, it is common practice to truncate the inspection process, typically after the number of items inspected reaches three times the number inspected in a corresponding single sampling plan. The decision lines are determined by specific equations, which are functions of parameters such as the probabilities of type I and type II errors (α, β), the acceptable quality level (p_1), and the rejectable quality level (p_2) [39]. The efficiency of this sequential sampling scheme is measured by the average sample number, which reflects the average number of samples needed over many trials at a fixed incoming defect level. This metric can be somewhat similar to the measurement of learning efficiency in active learning, which is commonly assessed using learning curves [37].

2.1.4 Statistical process control

Statistical process control (SPC) involves using statistical methods to monitor and control a process [41–44]. This technique is designed to detect significant changes in process behavior through continuous sampling and analysis. It is particularly relevant for stream-based active learning, especially because the methodology proposed in the upcoming chapters is largely influenced by the thresholding approach used in SPC. SPC comprises two key steps:

1. *Phase I.* This phase represents the offline segment of the analysis, where data considered to be under control is examined to estimate thresholds. These thresholds statistically determine whether a process output is behaving as expected or not.
2. *Phase II.* This phase encompasses the online segment of the analysis. Here, the thresholds established in Phase I are applied to detect deviations from standard conditions as they occur.

The application of SPC is profoundly relevant to stream-based active learning and scenarios with limited labeled data for several reasons. Primarily, SPC is inherently an unsupervised technique, relying predominantly on process measurements gathered via sensors to detect anomalies. Secondly, the online nature of Phase II, where measurements are analyzed sequentially, aligns with the real-time sampling objectives of stream-based active learning. However, a critical distinction between SPC and active learning lies in their primary objectives; SPC is primarily focused on anomaly detection, and it does not offer a mechanism for selecting specific observations to be labeled for enhancing a predictive model. For a deeper discussion on SPC, readers may wish to consult Montgomery [36].

Figure 2.4 shows an example of a Hotelling T^2 control chart, which is used in contexts where the data stream comprises multiple variables [45–50]. When each incoming observation $\mathbf{x}_i$ follows a multivariate normal distribution $\mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$, the Hotelling T^2 chart employs Phase I data to estimate $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$, and to set the upper control limit (UCL), which is used to decide whether the observations from Phase II are in-control or not. We can see how the chart in Figure 2.4 effectively highlights out-of-control situations, as most of the Phase II data points fall outside the UCL.

2.1.5 Multi-armed bandits

Multi-armed bandit (MAB) problems are another noteworthy example of sequential decision-making and sampling, where the primary objective is to systematically choose actions that maximize an overall outcome [51–54]. In the MAB framework,

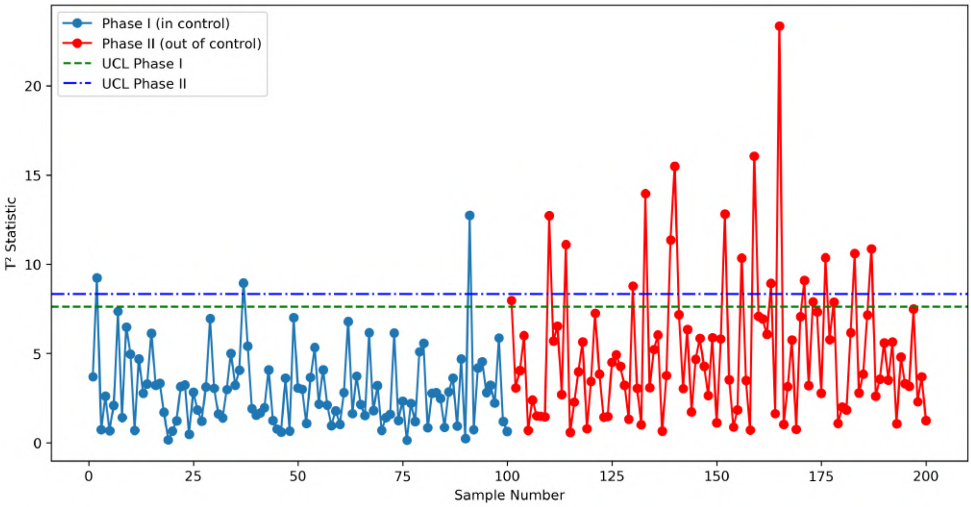


Figure 2.4. Example of a Hotelling T^2 control chart where three variables are being monitored and the mean of the process variables is increased from 0 to 1 while transitioning from Phase I to Phase II.

the learner needs to discern the most rewarding option from a set of alternatives, each characterized by distinct reward probabilities. This scenario is akin to a gambler choosing which arm to pull on a bank of slot machines, where each machine offers varying rewards. Similar to active learning techniques, MAB problems revolve around making sequential choices and sampling to obtain more information. In active learning, the choice involves selecting the most informative data points for labeling to enhance a learning model. In contrast, MAB focuses on picking the arm that promises the highest reward. Both paradigms are united by their reliance on feedback — learning from each action taken, whether it is acquiring a data label in active learning or receiving a reward in MAB.

Two primary formulations of MAB problems have been proposed, each distinct in its approach and objective. The first, regret minimization, seeks to optimize cumulative rewards across numerous trials. This strategy hinges on a nuanced interplay between exploration, which entails testing different arms to glean insights into their reward patterns, and exploitation, where the focus shifts to utilizing accumulated knowledge to select the arm with the highest expected payoff. Algorithms grounded in this approach strive to strike a balance between effective learning and the attainment of high rewards. This method finds its utility in diverse applications such as online advertising, recommendation systems, and treatment design. Conversely, the pure exploration strategy is centered around the identification of the most promising arm, given specific constraints like a finite number of trials. Unlike regret minimization, the emphasis here is not on immediate rewards but rather on acquiring a deeper comprehension of the system at play with the fewest possible trials. This approach is particularly relevant in scenarios where safety concerns or resource constraints are dominant, such as drug discovery and design optimization. In these cases, the goal is to discern the most effective course of action with minimal experimentation, ensuring safety and efficiency.

Both MAB problems and stream-based active learning exemplify frameworks of sequential experimental design, where each step in the process builds progressively upon previous knowledge. However, while active learning primarily targets model improvement through selective data labeling, MAB problems focus on maximizing rewards through strategic choices among available options. The key contrast lies in their objectives and application domains, with active learning concentrated on data-driven model enhancement and MAB on reward optimization. Within the realm of MAB problems, the study of linear bandits, where the reward is a linear combination of some input parameters, bears the closest resemblance to active learning [55–58]. A more detailed discussion about the commonalities between linear bandits and stream-based active learning can be found in Paper 2 (Appendix B).

2.2 Training Methods

Beyond sampling and monitoring strategies from industrial statistics, another relevant research area is represented by techniques that can be used to enhance training efficiency and model deployment in scenarios characterized by label scarcity. These methodologies are particularly adept at harnessing the value of the available labeled data while also leveraging the knowledge hidden in the unlabeled portion of the data or different models. This section delves into some of these innovative training methods, shedding light on their principles and applications in the context of label scarcity and streaming data. We explore how these methods, though distinct in their operational mechanics, share the goal of enhancing learning efficiency and model performance with limited labeled data. The techniques discussed here complement the sampling strategies addressed in the previous section, providing a broader spectrum of tools for tackling the challenges associated with label scarcity in machine learning.

2.2.1 Semi-supervised learning

Semi-supervised learning is one of the most commonly employed approaches for training models when we have a large dataset, but only a portion of it is labeled. It tackles the challenge of limited labeled data from a perspective opposite to that of active learning. While active learning strategically minimizes the labeling requirement for model training, semi-supervised learning integrates both labeled and unlabeled data into its training process [59]. Semi-supervised learning approaches can be classified into three categories:

1. *Unsupervised preprocessing.* This approach involves employing unsupervised learning techniques, such as dimensionality reduction, clustering, or feature extraction, across the entire dataset (both labeled and unlabeled) before its utilization in a supervised model. The aim is to transform the data into a format that facilitates the supervised task.

2. *Wrapper methods.* These methods utilize one or more supervised learners, trained on a combination of labeled data and (pseudo-labeled) unlabeled data. Pseudo-labels can be defined as labels that are inferred or estimated for the unlabeled data based on the predictions of the supervised model. These labels, though not verified by human annotators, are used to extend the training dataset, allowing the model to leverage more data for learning and improving its overall predictive accuracy. There are two key variants:

   - *Self-training.* A single supervised model is trained on labeled data, and confident predictions are used to pseudo-label other data points.
   - *Co-training.* Multiple supervised models exchange confident predictions for generating pseudo-labels for the unlabeled portion of the data.

3. *Graph-based methods.* In this strategy, a graph is constructed using all available data, and a supervised model is trained with a loss function that includes both a supervised component and a regularization term. This term penalizes discrepancies in predicted labels for connected data points in the graph.

Combining semi-supervised learning with active learning can refine data selection strategies, potentially leading to enhanced performance and greater efficiency. Figures 2.5 and 2.6 illustrate how semi-supervised learning can be integrated into the stream-based active learning routine.

2.2.2 Transfer learning

While semi-supervised learning involves leveraging knowledge from unlabeled data, transfer learning tries to leverage models trained on different, yet related, data. It involves transferring knowledge from a related task, which has abundant labeled data, to a target task with limited labeled data [60]. This approach leverages the commonalities between the source and target domains, enabling the learning process to benefit from pre-existing knowledge, thus reducing the need for extensive labeled data in the target domain. There are several forms of transfer learning, but the most common one involves fine-tuning a model pre-trained on a large dataset (like ImageNet or COCO) using a smaller dataset from the target domain [61]. This method has proven particularly effective in deep learning, where models trained on millions of images can learn general features that are applicable across a wide range of tasks.

The connection between transfer learning and the methodologies discussed in this thesis lies in their shared goal of overcoming the limitations imposed by label scarcity.

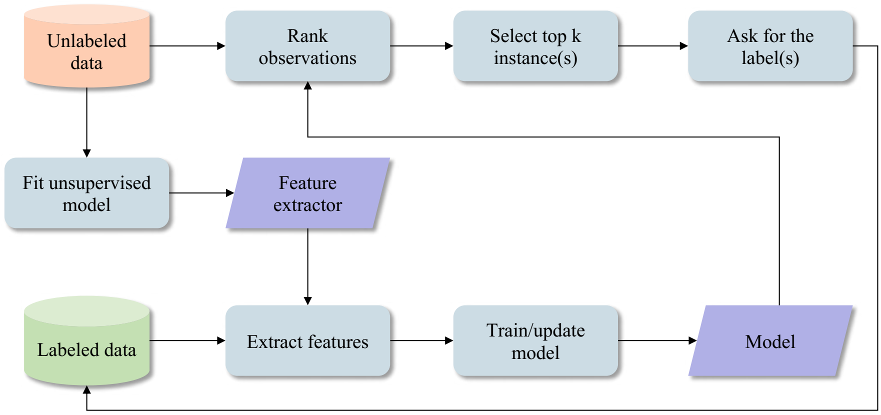


Figure 2.5. Combining semi-supervised learning based on preprocessing with stream-based active learning.

However, transfer learning differs significantly in its approach. While active learning strategies in this thesis focus on selecting the most informative samples from data streams for labeling, transfer learning circumvents the need for large amounts of labeled data by leveraging pre-existing models and knowledge.

2.2.3 Continual learning

Continual learning, also known as lifelong learning, is a dynamic approach in machine learning where the model aims to learn new tasks from new data while retaining previously acquired knowledge [62, 63]. This is particularly important in environments where data distribution evolves over time or when new tasks are introduced sequentially. The primary challenge in continual learning is mitigating catastrophic forgetting, a phenomenon where a model loses previously learned information upon learning new data. To address this, several strategies have been developed:

- *Regularization approaches.* These methods, such as elastic weight consolidation [64], add constraints to the learning algorithm in order to preserve important parameters for previous tasks while learning new ones. The general idea is not to let the new data significantly affect the parameters that are important for the previous task.
- *Reharsal methods.* These techniques, like experience replay, involve retaining a subset of previous data and mixing it with new data during training [65].

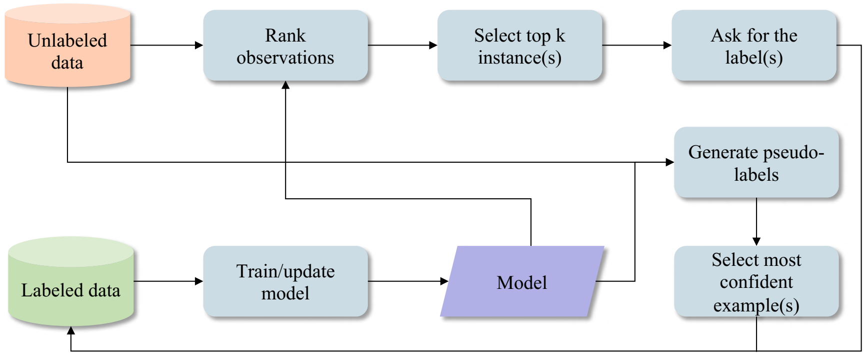


Figure 2.6. Combining semi-supervised learning based on self-learning with stream-based active learning.

Similar to an actor rehearsing lines to retain them in memory, this approach repeatedly presents old data to the model to prevent it from forgetting what it has previously learned.

- *Architectural methods.* This approach, exemplified by progressive neural network models [66], involves dynamically expanding the network architecture to accommodate new knowledge without altering the existing structure.

Continual learning can prove beneficial in scenarios with label scarcity as it enables the use of knowledge from previous tasks to enhance the efficiency of learning new tasks [67]. In this context, continual learning can be viewed as an extension of transfer learning, where the objective is not only to leverage knowledge from previous models to build new ones but also to maintain accuracy on previous tasks. Unlike traditional transfer learning, where the focus is on efficiently learning a new task often at the expense of older knowledge, continual learning seeks a balance to avoid catastrophic forgetting.

2.2.4 Few-shot learning

Meta-learning is a research area that is highly related to continual learning. It is often described as learning to learn, and it encompasses techniques that improve the learning process of algorithms, enabling them to quickly adapt to new tasks or retain knowledge across different tasks. Within meta-learning, few-shot learning is one of the most promising approaches, which emphasizes the model's capability to learn and generalize from very limited data. There are two primary approaches to few-shot learning:

- *Model-agnostic meta-learning.* This methodology focuses on training a model in such a way that it can be quickly adapted to new tasks with only a few training examples [68]. This is achieved by finding a model initialization that is particularly effective for fine-tuning on various tasks.

- *Matching networks.* This approach employs a unique training scheme that involves creating support and query sets to simulate few-shot learning scenarios during the training process [69]. Matching networks aim to learn a model that can generalize well to new tasks based on the learning experience from these simulated scenarios.

Both methods address the fundamental challenge in few-shot learning, developing models capable of making accurate predictions from a small number of samples.

2.2.5 Curriculum learning

Curriculum learning is another intriguing approach in the field of machine learning, which is related both to sampling and training. This technique aims at improving the training efficiency while providing examples to the learning model in a specific order [70]. It draws inspiration from the way humans learn: starting with simpler concepts and gradually progressing to more complex ones. In machine learning, this translates to initially training the model on simpler or easier examples and progressively introducing more complex or difficult ones. This approach is believed to improve the learning efficiency and final performance of the model. The rationale is that by starting with easier instances, the model can quickly learn the basic patterns, which then serve as a foundation for understanding more complex patterns. Curriculum learning shares similarities with active learning in terms of enhancing training efficiency by prioritizing the training on specific labeled examples. However, while active learning focuses on selecting the most informative samples (which may not necessarily be the easiest or simplest), curriculum learning is specifically about structuring the learning process based on the difficulty level of the examples.

CHAPTER 3

Stream-based active learning with linear regression models

This chapter introduces and discusses the methodology developed to address the stream-based active learning problem for linear regression models. Our investigation of this scenario is detailed in three different papers:

- PAPER 3 (Appendix C), which presents the active learning methodology based on the concept of CDO.
- PAPER 4 (Appendix D), which extends CDO by accounting for the presence of outliers.
- PAPER 5 (Appendix E), which explores the impact of irrelevant features in the data stream.

This chapter serves as a cohesive narrative that illustrates how these works are interconnected, placing the developed methodology in context and discussing its main strengths and limitations. These papers are grouped together in this chapter because they represent variations of the same methodology, and they share a common experimental design and underlying theoretical framework, which collectively contribute to a more comprehensive understanding of the stream-based active learning process in the context of linear regression models.

3.1 Problem statement

Regression models are pervasive in data science and rank among the most commonly employed types of models for various applications, ranging from forecasting in energy and finance to quality prediction in manufacturing. Among regression models, linear models are widely used due to their ease of interpretation, with the possibility of performing tests on coefficients to verify relationships between specific input variables and process outcomes. Indeed, for industrial practitioners, process understanding and optimization are often more interesting than prediction per se. That is why linear models continue to be widely employed in many fields. A multiple linear regression model is generally defined as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{3.1}$$

where $\mathbf{y}$ is the $n \times 1$ vector containing the response variable, $\mathbf{X}$ is an $n \times p$ design matrix, $\boldsymbol{\beta}$ is a $p \times 1$ vector of regression coefficients, and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ is an $n \times 1$ vector representing the zero-mean Gaussian noise. Here n is the number of observations, and p is the number of predictors (or covariates).

The general stream-based active learning framework for linear regression is illustrated in Figure 3.1. We typically start with an initial random design comprising k observations, which is used to obtain an estimate of the regression coefficients, $\widehat{\boldsymbol{\beta}}$. This initial model might be trained on historical data or on preliminary experiments conducted to gain a basic understanding of the problem. Subsequently, the model interacts with a data stream $\mathcal{S}$, receiving unlabeled observations, $\mathbf{x}$, in real time. For these observations, the learning model can decide whether to request the corresponding labels, y, or not. However, there is an operational budget limiting the number of labels that can be collected. The main objective is to refine the model estimate $\widehat{\boldsymbol{\beta}}$ by iteratively selecting the most informative labeled examples, $(\mathbf{x}, y)$. These are the observations that, in hindsight, would lead to the most significant improvement in the model if all the labels were known. Here, model improvement is often measured in terms of predictive performance, such as the root mean square error on an external test set or using the prequential evaluation scheme [37].

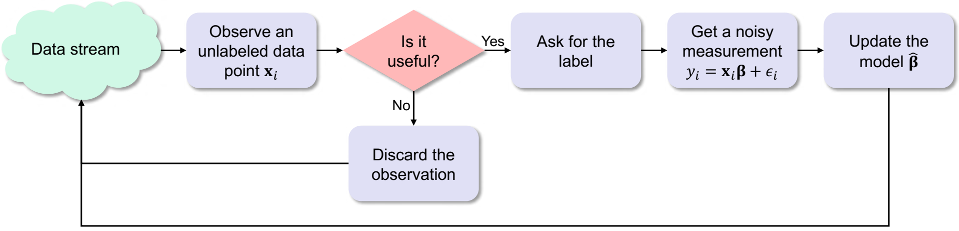

Data stream
Observe an unlabeled data point $\mathbf{x}_i$
Is it useful?
Yes
No
Ask for the label
Get a noisy measurement $y_i = \mathbf{x}_i\boldsymbol{\beta} + \epsilon_i$
Update the model $\widehat{\boldsymbol{\beta}}$
Discard the observation

Figure 3.1. Stream-based active learning with linear regression models. This routine is repeated until we meet the budget constraint on the number of labels that can be queried.

3.2 Proposed methodology

The general approach we adopt for the stream-based active learning problem draws inspiration both from the secretary problem (see Chapter 1, Figure 1.4) and from the traditional SPC approach (see Section 2.1.4). Indeed, our strategy is based on two phases:

1. *Warm-up Phase.* This phase is analogous to Phase I data in SPC, where, in this case, we observe the process without querying any labels. Instead, we collect an unlabeled calibration set that is used to estimate the covariance matrix of the data and a threshold Γ, which is later utilized to identify the most informative data points.

2. *Sampling Phase.* This phase is similar to Phase II data in SPC and represents the phase where we select the most informative observations. Instead of declaring faulty or abnormal situations, data points falling above Γ are the observations used to update the model $\widehat{\boldsymbol{\beta}}$.

This approach is also similar to the general solution to the secretary problem. Indeed, the general solution involves deciding a sample size r, blindly rejecting the first $r-1$ candidates, and then selecting the first candidate whose skill set is superior to those observed in the previous $r-1$ candidates [71]. In Figure 3.2, we can see how this is somewhat similar to our proposed methodology, where we first gather a reference set to gain a general idea of the level of the candidates and then use that set to inform our hiring decision. The key difference in our framework is that we are not focused on selecting a single best data point; instead, we continuously select data points until our budget is exhausted. This approach simplifies the problem, as it allows for some margin of error in individual selections while still progressing towards overall model improvement.

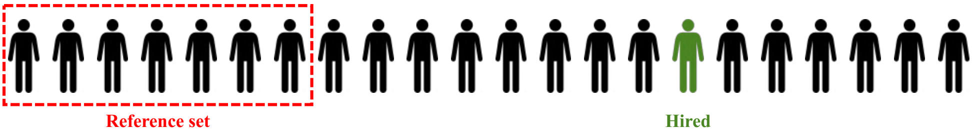


Figure 3.2. General solution of the secretary problem, where we initially reject a sample of candidates and then select the first one who is better than those in the reference set.

3.2.1 Conditional D-optimality

The previous section introduced the key thresholding approach adopted. However, the fundamental question of how to determine if a data point is informative was not answered. In Paper 3 (Appendix C), we propose the CDO approach for selecting the most informative observations in the stream-based active learning framework for linear regression. We propose the use of an instance evaluation criterion borrowed from the optimal experimental design theory. The key contributions of the paper are:

- We propose labeling observations that have a high UPV, highlighting the connection between UPV and D-optimality. Indeed, points with high UPV are those that contribute to maximizing the determinant of the moment matrix. These points are useful to the model as they belong to less explored locations of the input space; thus, their inclusion in the model encourages the exploration of new regions.
- We propose a thresholding approach, demonstrating how a warm-up set can be used to estimate the covariance matrix of the data (which can be used for whitening purposes) and to estimate the threshold Γ.
- We propose the use of kernel density estimation (KDE) for determining Γ. In particular, we used a KDE with a Gaussian kernel to estimate the UPV scores of the observations in the warm-up set.
- We provide extensive results on numerical simulations and the Tennessee Eastman Process, focusing on two aspects:
  - *Learning efficiency.* We show how the proposed approach allows for a faster reduction in the error rate compared to random sampling and the norm-thresholding approach [72].
  - *Computational time.* We demonstrate that the decision time of the proposed sampling strategy is extremely low (around 0.004 milliseconds), proving that it is a suitable method for streams with very high arrival rates.

3.2.2 Dealing with the presence of outliers

One of the key aspects of seeking points with high UPV is that these points will likely be far from the current design space. While this approach allows us to explore new, unseen regions and improve our model, it might be counterproductive if the data stream is affected by the presence of outliers. Indeed, if isolated outliers exist in the data stream (e.g., measurement errors), the model is likely to be attracted to them due to their high UPV, but their inclusion in the training set could eventually degrade prediction performance. In Paper 4 (Appendix D), we provide a solution to this problem. The contributions of this paper are:

- The inclusion of robust estimators in the CDO scheme, based on the Huber and Tukey loss functions. These approaches allow for the development of models that are less sensitive to the inclusion of outliers. This means that even if outliers are inadvertently included in the training set, the model estimation will not be significantly influenced.

- The suggestion of using two thresholds to bound the search area of the CDO algorithm. If extremely large UPV values suggest that a data point is an outlier, we will not query its label. The double-threshold approach works by estimating two thresholds (Γ_1, Γ_2) to identify a safe sampling interval.

- An investigation into the difference in performance when the initial training set, composed of k observations, includes outliers versus when it is clean.

- The exploration of a weighted UPV, using the weight matrix $\boldsymbol{W}$ derived from the robust estimators.

- An investigation into the performance of two performance-based stopping criteria. Instead of stopping the active learning routine when the budget is exhausted, we consider approximating the true (unobservable) learning curve. This approach allows us to stop the procedure earlier if we are not improving the model or to suggest the experimenters continue sampling even after exhausting the budget.

3.2.3 Dealing with the presence of irrelevant features

Another potential challenge that may affect the stream-based active learning framework is the presence of an excessive number of features. This is increasingly common, as collecting process data is usually straightforward, leading to a large number of predictors, p. However, in supervised modeling processes, it is unlikely that all p features significantly affect the response y. In Paper 5 (Appendix E), we provide an initial analysis of the potential benefits of including a feature selection step in the CDO algorithm. The contributions of the paper are:

- The evaluation of the performance of different feature selection methods, including an information criteria-based one, a sparse method, and a linear regression-specific method.

- The assessment of the effectiveness of the feature selection methods in two aspects:
  - *Learning curve.* We compare how the learning methods perform relative to the optimal approach, where we know beforehand which features truly affect the response.
  - *Detection rapidity.* We evaluate how quickly the feature selection strategies can identify the relevant features, i.e., the number of learning steps required before they can effectively screen.

- We highlight two potential causes for the decreased efficiency when irrelevant features are present in the stream:

- *Parameter estimation.* When we have a few number of observations to estimate the regression coefficients, the burden posed by the presence of additional features might significantly impact the efficiency of the active learning routine.
- *Suboptimal exploration.* If we calculate the UPV based on irrelevant features, we might be exploring areas of the input space which are uncertain for a misspecified model, and thus might not be optimal queries for the true model which is only based on a subset of the observed features.

3.3 Key results

The first significant result pertains to the improvement achieved using the CDO strategy compared to random sampling. In Figure 3.3, we can observe how the proposed CDO approach yields substantial improvements to the learning curve, especially when the experimental budget is limited. Generally, as more observations are collected, all strategies converge to the performance level achieved by random sampling. However, in the initial steps, a performance improvement of up to 25% compared to the passive random approach can be observed. This is also evident in Figure 3.4, where the residuals of the model obtained after five learning steps are displayed. Here, we see how carefully selecting a small number of observations can significantly impact the performance of the resulting model.

In Figure 3.5, we can see how the presence of outliers can dramatically decrease the learning efficiency of the proposed methodology. Adopting a flexible strategy based on robust estimators and two thresholds proves to be an effective solution to this problem. Lastly, Figure 3.6 reveals the varied effectiveness of different feature selection methods in promptly identifying the relevant features. The Lasso estimator quickly and accurately identifies relevant features, while F-tests, though slower, are effective over time. Mutual information heatmaps initially show potential but tend to mistakenly include irrelevant features, especially in the early stages with limited data. This highlights the diverse strengths and limitations of each feature selection approach in such a dynamic learning environment.

3.4 Discussion

The proposed methodology offers new insights and approaches to stream-based active learning for linear regression models. Here are some discussion points that underscore the limitations of the works and potential directions for future research:

- *Model choice.* While the approach is specifically tailored for linear models, it is not restricted to simple first-order models. It can be easily adapted for high-order polynomials that are linear in parameters, allowing the learning of more

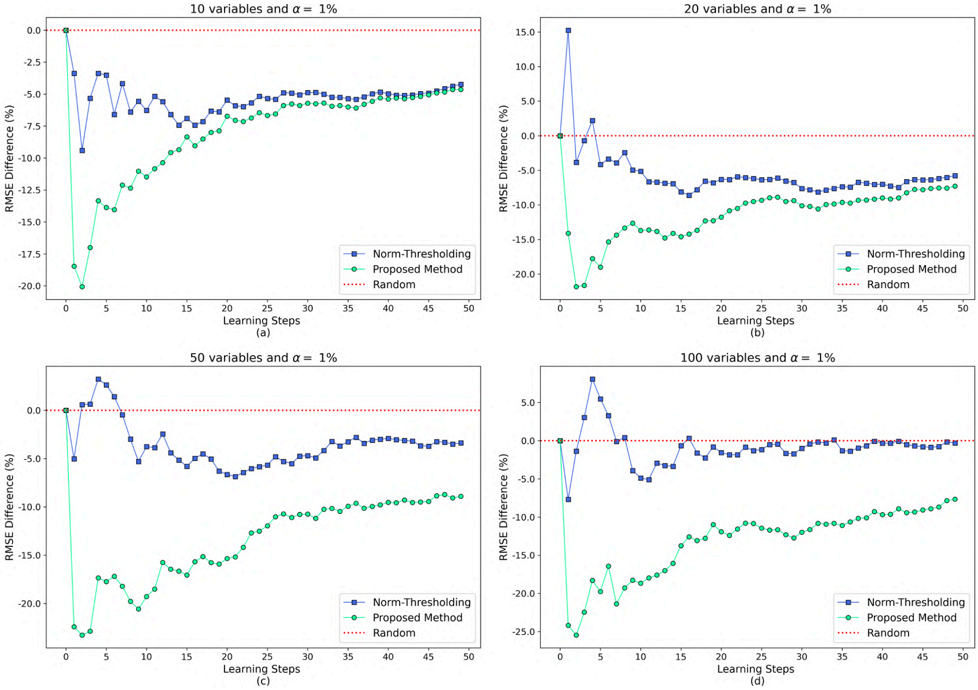


Figure 3.3. Learning curves obtained on numerical simulations (from Paper 3). The proposed method is the CDO strategy.

complex nonlinear functions. Additionally, a linear model could be employed after extracting nonlinear features from the data using unsupervised learning methods, as highlighted in Section 2.2.1.

- *Threshold estimation.* KDE is a generally efficient method, surpassing simple empirical quantile approaches. However, it introduces complexities such as increased computational demands and the challenge of bandwidth selection. We utilized a scalar factor multiplied by the standard deviation of the scores, resulting in a bandwidth that adapts to data spread. Nevertheless, the bandwidth choice significantly influences the KDE shape, and different methods may be more suitable depending on the specific data distribution.

- *Outlier definition.* The statistical definition of an outlier can greatly affect the performance of the developed procedure. In Paper 4, an outlier is defined as an isolated point with a shift in covariates, and a response not deriving from the same underlying model as the other points in the stream. It is an anomalous point that does not contribute to estimating the true regression coefficients $\boldsymbol{\beta}$. While this yields interesting results, many ways to define an outlier exist, and

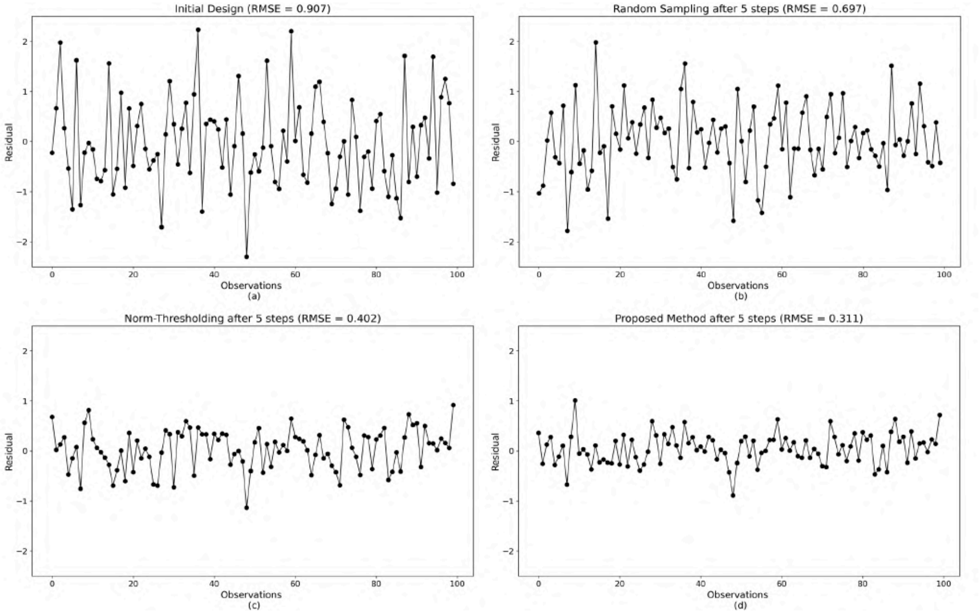


Figure 3.4. Residuals after five learning steps on the Tennessee Eastman Process (from Paper 3). The proposed method is the CDO strategy.

the effectiveness of the proposed solution may vary significantly based on these definitions.

- *Screening scenario.* The experimental scenario in Paper 5 is quite simplistic, considering a data stream with uncorrelated predictors. The investigation was also limited to relatively basic feature selection methods. Future work should explore more complex data distributions (e.g., with varying levels of feature correlation) and more sophisticated feature selection techniques.

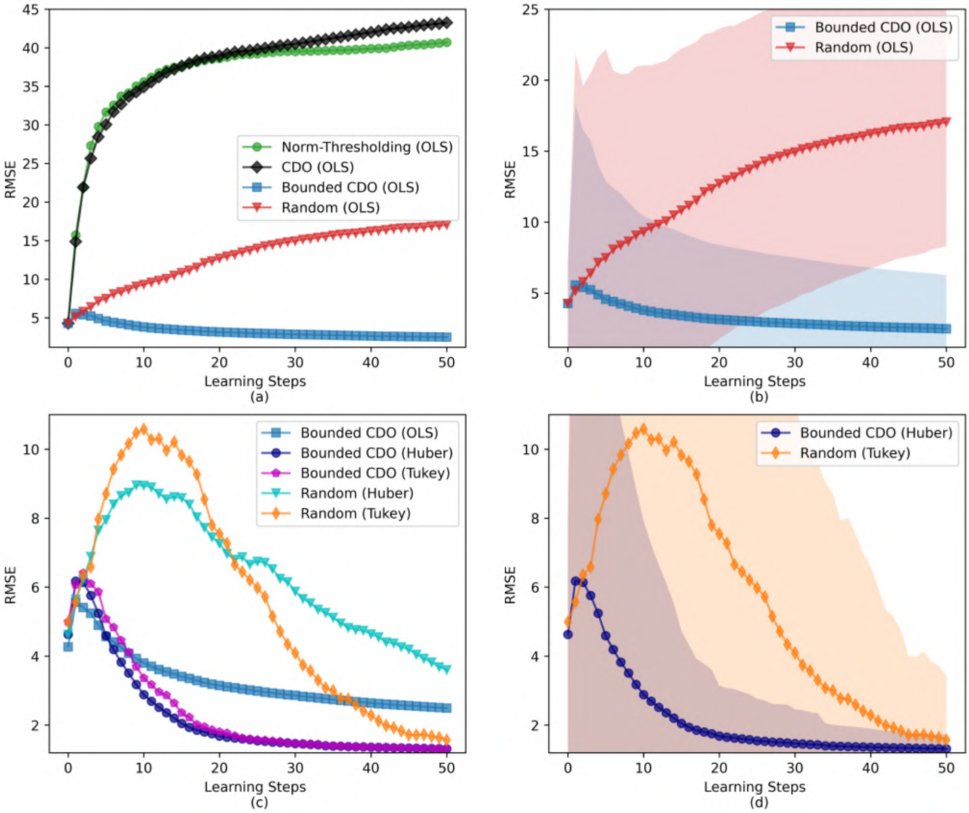


Figure 3.5. Learning curves obtained on numerical simulations, with 5% of the observations in the data stream corresponding to outliers (from Paper 4).

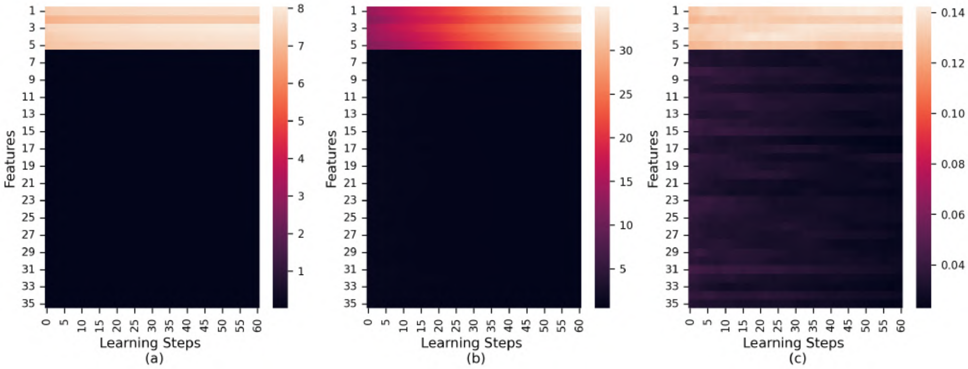


Figure 3.6. Scores obtained with the different feature selection methods with 30 irrelevant features. Subplot (a) shows the squared regression coefficients obtained with Lasso. Subplot (b) shows the feature importance scores obtained by performing univariate F-tests. Subplot (c) shows the mutual information scores (from Paper 5).

CHAPTER 4

Adaptive sampling for concept drift monitoring in data streams

This chapter investigates the use of adaptive sampling methods for identifying concept drifts in data streams. Before delving into the adaptive sampling methodology, we provide a preliminary section where we underline the significance and potential impacts of concept drift for the active learning routine. We then present initial results on the use of a residual-based sampling strategy for data streams, based on weighted KDE and the Metropolis-Hastings (MH) algorithm. For an explanation of concept drift and other types of distribution shifts in data streams, readers may wish to consult Section 3.2 of Paper 2 (Appendix B).

4.1 Preliminaries on active learning with concept drift

Concept drift in data streams can generally be defined as a circumstance where the presence of hidden effects alters the relationship between the input features and the response of a model [73]. This leads to a change in the conditional distribution of the response from time t to time $t+\Delta$, as in

$$P_t(y|\mathbf{x}) \neq P_{t+\Delta}(y|\mathbf{x}), \quad \text{while} \quad P_t(\mathbf{x}) = P_{t+\Delta}(\mathbf{x}) \tag{4.1}$$

This means that concept drift cannot be detected simply by monitoring covariates, as it is usually done in SPC (see Section 2.1.4). In this brief experiment, we demonstrate what happens to the residuals of a regression model, measured using a prequential evaluation scheme [37], in three different scenarios. In each scenario, the ith covariates vector is generated as $\mathbf{x}_i \sim \mathcal{N}_p\left(\mathbf{0}, \mathbf{\Sigma}_0\right)$, where p is the number of input features and $\mathbf{\Sigma}_0 = \sigma_{\mathbf{x}}^2\mathbf{I}$, with $\sigma_{\mathbf{x}} = 1$. At each time step, the response is obtained using $y_i = \mathbf{x}_i\boldsymbol{\beta} + \varepsilon_i$, where $\boldsymbol{\beta}$ are the true regression coefficients, and $\varepsilon_i \sim \mathcal{N}\left(0, \sigma_\varepsilon^2\right)$ is the zero-mean Gaussian noise, with $\sigma_\varepsilon = 1$. After fitting an initial model on an initially labeled training set, we collect additional observations by drawing, at each time step, a number $r \sim \mathcal{U}(0,1)$ and selecting the data point only if $r \geq 0.9$. This corresponds to a random sampling scheme with a sampling rate of 10%. Random sampling is used for simplicity, as the main scope of this chapter is to show the behavior of the residuals in drifting environments.

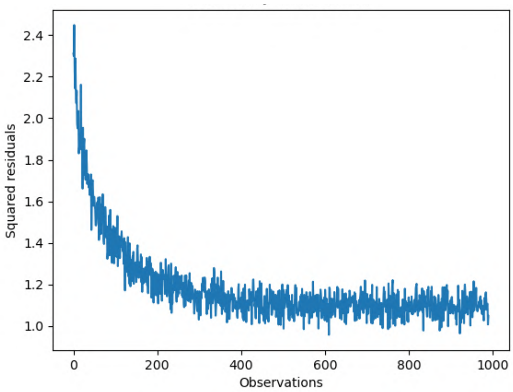


Figure 4.1. Learning curve on a stationary data stream (random sampling with $\alpha = 10\%$, 100 simulation runs).

Figure 4.1 reports the performance obtained on a stationary data stream, where the coefficients $\boldsymbol{\beta}$ used to generate the response remain constant throughout the observation period. It is evident how the performance decreases, indicating that the additional labeled observations allow for a better estimation of the underlying model relating the process variables to the response. Conversely, Figure 4.2 illustrates what happens to the residuals when abrupt concept drifts affect the data stream. Here, an abrupt concept drift is represented by a sudden change in the true regression coefficients $\boldsymbol{\beta}$, suggesting that the underlying relation between predictors and response variable is altered by hidden effects [73]. Specifically, we introduced a concept drift every 100 observations, which is evident from the spikes in the residuals. When a drift occurs, the model trained on obsolete data cannot accurately predict the response. However, with the collection of new labels, the learning curves show an attempt to learn the new concept, which is continually negated by newly induced drifts. Then, once the sampling budget of 50 observations is exhausted (approximately around observation number 500), the residuals cease decreasing as the learner has used all of its exploration budget. A similar behavior is evident in Figure 4.3, where the only difference from the previous case is that the concept drift is gradually introduced, with a transition phase showing the switch between the two underlying models.

These simple experiments aim to highlight the main challenges that arise when sampling from data streams experiencing concept drift, and the actions that may need to be taken to address them:

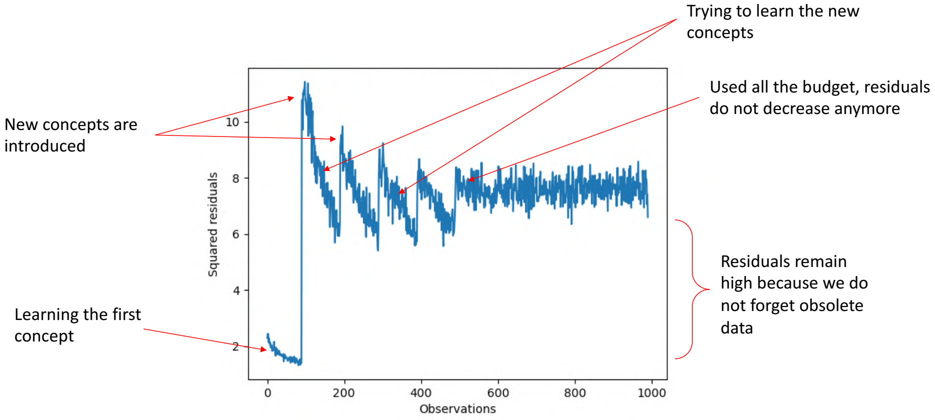


Figure 4.2. Learning curve on an abruptly drifting data stream (random sampling with $\alpha = 10\%$, 100 simulation runs).

- *Adjusting sampling rate.* When a drift occurs, it is crucial to collect a significant amount of data to facilitate the learning of the new model.
- *Forgetting obsolete data.* Attempting to learn new concepts without disregarding obsolete data will hinder the model from achieving optimal predictive performance.
- *Balancing sampling budget.* In the presence of drifts, it is necessary to judiciously balance the sampling budget, discerning when the data stream is in a stationary phase and when it is transitioning to a new concept.

4.2 Problem statement

We now introduce the problem of detecting localized concept drifts in data streams. In the previous section, we demonstrated what happens to the learning curve when the underlying regression model changes over time. However, in that scenario, we assumed that all the model coefficients were altered, meaning that all incoming data points from the stream would report significantly higher residuals, regardless of their specific location within the design space. In reality, it is possible that only some coefficients of the model are affected, or that the new concept is revealed only in certain regions of the covariates. For instance, the effect of changing a material in production or replacing a component might only lead to different behaviors when the

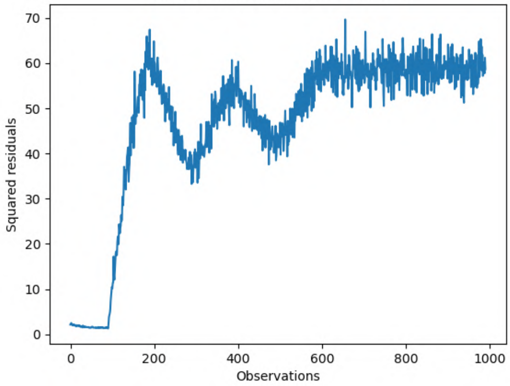


Figure 4.3. Learning curve on a gradually drifting data stream (random sampling with $\alpha = 10\%$, 100 simulation runs).

temperature or pressure rises above a certain levels. Similarly, for a large product or batch of products, the change might only be observable in specific locations. This phenomenon is illustrated in Figure 4.4, where the model is altered only in the region where $x \in [1, 2]$. The original function is from Gramacy and Lee [74]. This observation underlines the necessity for detection methodologies specialized for localized drifts. For any observation $x \in [0.5, 1] \cup [2, 2.5]$, the residual pre and post-drift would remain the same (except for inherent noise). Therefore, if one were to employ random sampling across the feature space, the detection of the drift could easily be missed. An aggregated measure of residuals, such as the sum of squares or absolute values, would likely show minimal deviation from zero, particularly if a significant number of points were sampled from unaffected regions. This could misleadingly indicate that a model update is not necessary, potentially compromising the predictive performance of the model in use. Thus, it becomes evident that a more nuanced approach is required for effectively identifying and responding to concept drifts, particularly those with a localized nature. Another example is depicted in Figure 4.5. Here, the concept drift appears in a region where the response was previously flat. This example also modifies a function proposed by Gramacy and Lee [74]. A different visualization of the same drifting function is presented in Figure 4.6. These examples highlight the necessity of an adaptive sampling strategy that can recommend optimal sampling locations for prompt drift identification, followed by an appropriate model update. In Figure 4.7, we see an additional example that represents the sequential nature of the problem, where a drift gradually increases in magnitude over time.

Since we are dealing with data streams, we assume that at each time step t, we need to estimate the function over the entire feature space. At each time step, we are also given a finite budget for performing quality inspections along the domain $\mathcal{X}$ to double-check the validity of the model. Thus, the core idea is to inform the location

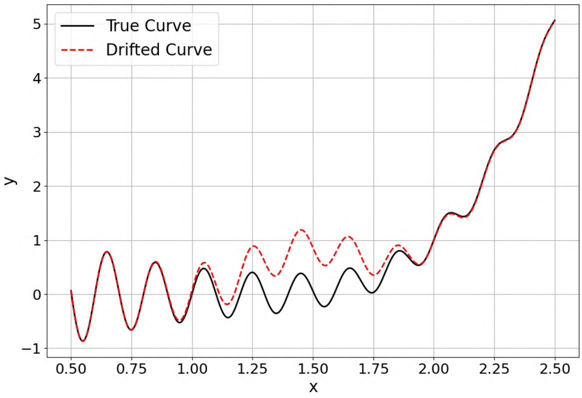
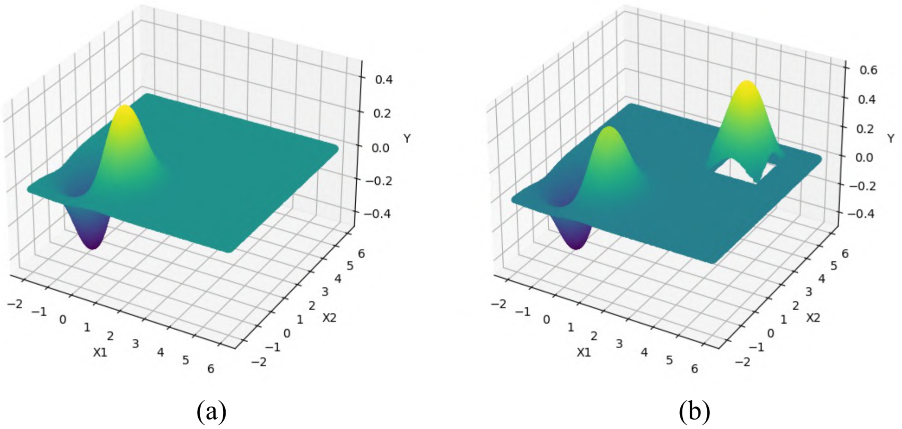


Figure 4.4. Effect of localized concept drift on a 1D function. The underlying model is locally altered from time t_1 to time t_2, indicating $P_{t_1}(y|x) \neq P_{t_2}(y|x)$ only when $x \in [1, 2]$.

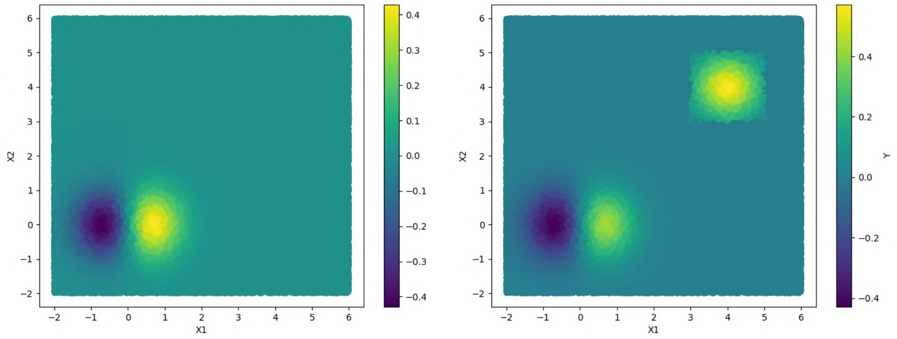


(a)

(b)

Figure 4.5. Effect of localized concept drift on a 2D function. Subplot (a) shows the original function, and subplot (b) shows the drifted function (in the region where x_1 and x_2 are between 3 and 5).

of the sampling at time t_{i+1} using the information available at time t_i. Intuitively, if we are at time t_1, we might start the process by collecting data randomly or using a space-filling design [75] to initialize the model and obtain some initial samples to measure its accuracy. Then, from successive time steps, we can use the model and the newly collected observations to adapt the sampling strategy in an attempt to be more sensitive to potential model alterations.



(a) (b)

Figure 4.6. Effect of localized concept drift on a 2D function (contour plot). Subplot (a) shows the original function, and subplot (b) shows the drifted function (in the region where x_1 and x_2 are between 3 and 5).

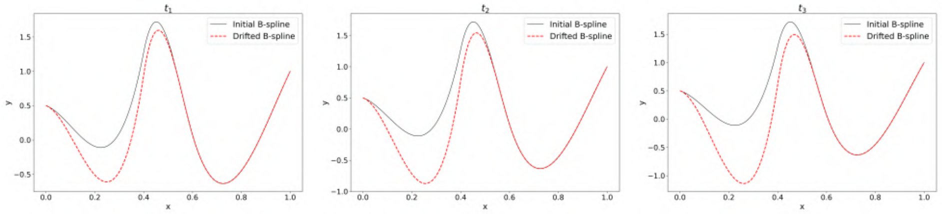


Figure 4.7. Observing concept drift over time: the curve is changing in the same region but the magnitude of the coefficients changes gradually. These plots are obtained using a B-spline function where only one coefficient is gradually increased over time. We can see how the concept drift is mostly affecting the region where $x \in [0, 0.5]$.

4.3 Proposed methodology

The proposed methodology aims to assign a sampling probability to each point $\mathbf{x} \in \mathcal{X}$, based on some informativeness measure that encourages sampling in drifting regions. We investigate the performance of two different kinds of models. The first model is a polynomial [76], which can be described as

$$y = \beta_0 + \sum_{j=1}^{p} \beta_j x_j + \sum_{j=1}^{p} \sum_{k=j}^{p} \beta_{jk} x_j x_k + \epsilon$$

where y is the response variable, x_j are the predictors, $\beta_0, \beta_j, \beta_{jk}$ are the coefficients for the intercept, linear and interaction terms up to the pth predictor, and ϵ represents the error term. The second model is a Gaussian process regression (GPR) model. A Gaussian process (GP) can be generally described as a collection of random variables, any finite number of which have a joint Gaussian distribution [77]. In the regression case, the random variables represent the continuous value of a function $f(\mathbf{x})$ at location $\mathbf{x}$. A GP can be defined as

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \tag{4.2}$$

where $m(\mathbf{x})$ represents the mean function, usually assumed to be zero, and $k(\mathbf{x}, \mathbf{x}')$ represents the covariance function, also referred to as the kernel. The squared exponential (SE) covariance function is frequently utilized as a kernel in various models due to its highly smooth nature. The SE kernel between two points $\mathbf{x}$ and $\mathbf{x}'$ in the input space is given by:

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\ell^2}\right) \tag{4.3}$$

where ℓ represents the length scale parameter. The SE kernel might also include a scale factor σ^2 to control the variance. The Gram matrix $\mathbf{K}$ is the covariance matrix for a set of input points $\mathbf{x}_i$, with $i = 1, \ldots, n$, where $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.

Assuming these models have already been trained on historical data (or data collected from the first time steps), we suggest using the residuals observed up to the current time step, to assign a sampling probability to each point $\mathbf{x} \in \mathcal{X}$. This probability will then be used to inform the sampling decisions in successive time steps. To do this, we propose the use of KDE to estimate the probability density function (PDF) of the sampling locations explored up to the previous time step, which in this case corresponds to. This provides insight into the spread of the observations across the x domain. Areas with higher density values suggest regions where more observations were collected, and lower densities indicate regions with fewer observations. If observations are uniformly spread, the density remains constant across x. While this representation captures the distribution of observations, it does not inherently reflect the model performance or account for potential concept drifts. To integrate model performance into the density estimation, we suggest using residuals from the previous time step as weights in the KDE. By computing a weighted KDE of the data locations using squared residuals as weights, we are effectively creating a density estimation that magnifies regions of the input space where the model made larger errors. Given N data points $x_1, x_2, \ldots, x_N$, the weighted KDE is given by

$$\hat{f}(x) = \frac{\sum_{i=1}^{N} r_i^2 K(x - x_i)}{\sum_{i=1}^{N} r_i^2} \tag{4.4}$$

where r_i is the residual of the current model for the ith observation, and $K(x - x_i)$ is the standard Gaussian kernel

$$K(x - x_i) = \frac{1}{\sqrt{2\pi h^2}} e^{-\frac{1}{2}\left(\frac{x - x_i}{h}\right)^2} \tag{4.5}$$

Here, the standard deviation h denotes the bandwidth, a smoothing parameter that in this study was set to 0.05. In essence, KDE works by placing a Gaussian (or another kernel) at each data point. By summing up these Gaussians, we obtain a smooth and continuous representation of the data distribution. Using the residuals as weights means that regions with larger residuals will have a more pronounced representation in the KDE. This can be valuable when deciding where to sample next, especially if you want to focus on regions where the model previously performed poorly.

Once we have a PDF, we can use it to decide where to sample at the next time step. However, sampling from an estimated PDF, especially when it is non-parametric like our KDE, is not trivial. One possible approach is to use the MH algorithm, which is a kind of Markov chain Monte Carlo (MCMC) method. It can be particularly useful when we are interested in sampling from a target distribution, in this case represented by our KDE, for which we can evaluate the density but cannot directly sample from because the analytical form of the distribution is not tractable. The main idea behind the MH algorithm is to construct a Markov chain whose stationary distribution is the target distribution. By running this chain for a long duration and taking samples from it, we can approximate samples from our target distribution. The key steps are:

1. *Initialization*: start with an arbitrary point x_0 in the domain.
2. *Proposal Distribution*: at each step, propose a new point to be sampled x' using a proposal distribution $q(x'|x_t)$.
3. *Acceptance Criterion*:
   - Compute the acceptance ratio α as

     $$\alpha = \min\left(\frac{f(x')q(x_t|x')}{f(x_t)q(x'|x_t)}, 1\right) \tag{4.6}$$

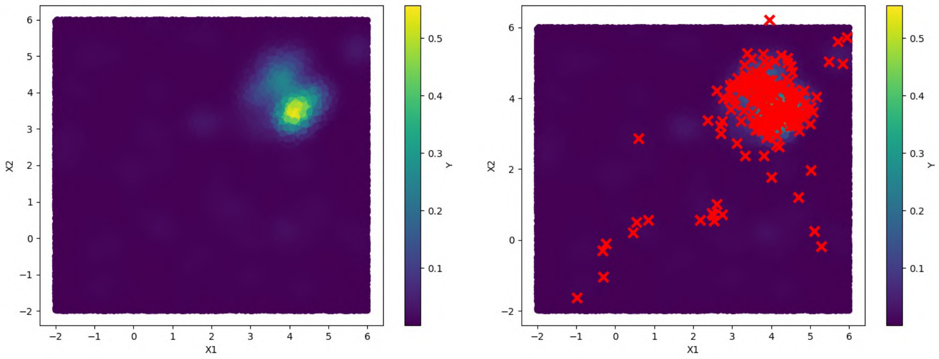
     where $f(x)$ is the value of the KDE at x.
   - Generate a random number $u \sim U(0,1)$.
   - Accept x' if $u \leq \alpha$.
4. *Repeat*: continue this process for a predefined number of iterations.

Here, we used a random walk MH algorithm, where $q(x'|x_t)$ is a Gaussian centered at the current point x_t. This means that at each step the proposal is obtained as $x' = x_t + \mathcal{N}(0, \sigma)$, where σ is a parameter to be tuned. The essential intuition behind MH is that even if our proposal distribution is not perfect, the acceptance criterion corrects for it, ensuring that over time we obtain samples that are representative of our target distribution.

To ensure this procedure ensures the discovery of drifts arising in new locations, it is necessary to balance exploitation and exploration. Indeed, the approach proposed so far is solely focused on exploiting the residuals from the previous time steps. While this is valuable, there is a risk of being short-sighted and missing concept drifts emerging in new areas. A more comprehensive strategy would balance both the exploitation of the high-residual locations and the exploration of new regions. To achieve this balance, we can employ the ϵ-greedy strategy, a method well-established in reinforcement learning. In the case of Q-learning with k actions, it selects its highest valued (greedy) action with fixed probability $(1 - \epsilon(k - 1)/k)$ and randomly selects among the other $k - 1$ actions with probability ϵ/k [78]. Here, we can use it to sample with probability ϵ at random from the entire domain, and with probability $1 - \epsilon$, from the MH algorithm based on residuals. This ensures that some fraction of the samples will always be exploratory.

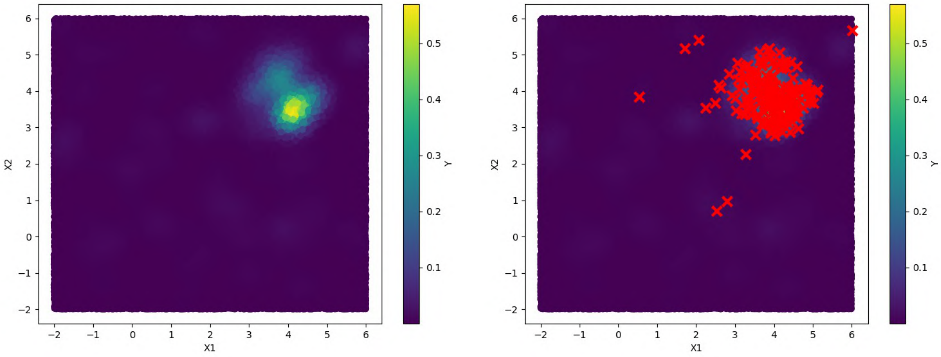
4.4 Key results

In Figure 4.8, we observe the PDF estimated using the weighted KDE approach with the polynomial model, along with the sampling locations suggested by the MH algorithm. For this estimation, we utilized 200 previously collected labeled observations from the drifting function depicted in Figure 4.5. The sampling strategy effectively recommends placing new samples in the region where the drift is occurring. In Figure 4.9, we present the outcomes of the sampling strategy, on the same function, based on the residuals obtained from the GPR model.



(a) (b)

Figure 4.8. KDE weighted by the residuals obtained from the polynomial model (a), and sampling locations suggested by the MH algorithm.



(a)

(b)

Figure 4.9. KDE weighted by the residuals obtained from the GPR model (a), and sampling locations suggested by the MH algorithm.

4.5 Conclusion

Despite the promising results obtained with the residual-weighted KDE, the approach we have presented is still in its initial, proof-of-concept stage. Moving forward, our aim is to expand this framework by implementing it in a more sequential manner. This advancement would facilitate an iterative enhancement of the PDF estimation through weighted KDE, effectively illustrating the gradual detection process of concept drifts. Furthermore, while our current strategy primarily addresses the sampling aspect in data streams, it is crucial to integrate a robust monitoring mechanism. This mechanism would be pivotal in accurately determining the precise moment to declare the occurrence of a concept drift. In pursuit of this objective, we are contemplating the adoption of an exponentially weighted moving average (EWMA) control chart. This method would allow for a dynamic and responsive monitoring system, capable of quickly identifying shifts in data stream patterns and facilitating timely interventions.

CHAPTER 5

Stream-based active distillation with object detection models

Object detection is a foundational task in computer vision with applications ranging from real-time surveillance to advanced robotics. In this chapter, we introduce the stream-based active distillation framework for object detection models proposed in Paper 6 (Appendix F). However, since the topic of object detection has not been extensively treated in the literature review (Paper 2) or in the introductory chapters, before delving into the proposed methodology we will briefly cover the core techniques, metrics, and strategies for fine-tuning object detection models. We also discuss the relevance of active learning in this context. Such an overview not only elucidates the foundational mechanics and inherent challenges of the domain but also paves the way for the stream-based active distillation methodology presented in the upcoming sections.

5.1 Preliminaries on object detection

Object detection enables the development of advanced models capable of identifying and locating objects of interest within images or videos. In recent years, the advancements in deep learning have led to significant progress in object detection, with state-of-the-art models achieving impressive performance on various benchmarks. Object detectors can be broadly divided into two classes, traditional computer vision techniques and deep learning models [79, 80].

5.1.1 Traditional approaches vs. deep learning-based approaches

In the early stages of computer vision, object detection was primarily dominated by traditional approaches that heavily relied on handcrafted features and ad-hoc algorithms. These methods laid the foundation for the field and solved some specific tasks effectively. Among these methods, the Viola-Jones algorithm [81] has long been the go-to method for face detection. It utilizes Haar-like features, which are divided into edge, line, and four-sided features, to efficiently capture local intensity variations in an image. The most relevant sub-regions of the image are then estimated using an AdaBoost classifier [82]. Viola-Jones achieved remarkable speed and accuracy in face detection and became widely adopted in various applications. Another popular traditional method is the histogram of orientated gradients (HOG), which relies on the gradients of the image to capture local edge or intensity variations. After a preprocessing step, HOG usually computes horizontal and vertical gradients using techniques like Robert, Prewit, or Sobel operators [83]. It then divides the image into small overlapping cells and computes a histogram of gradient orientations for each cell. These histograms are then concatenated to form the final feature vector used for classification. Deformable parts models (DPM) are another approach that has been highly effective in detecting objects with articulated structures, like humans or animals. Instead of treating the entire object as a single entity, DPMs represent objects as a collection of deformable parts. Each part is associated with its appearance model and spatial constraints that capture the typical spatial relationships among the parts. DPMs excel at handling variations in object pose and articulation. Due to their simplicity, ease of implementation, and interpretability, traditional approaches like the Viola-Jones algorithm, HOG, and DPMs have been widely employed by the computer vision community. However, it is important to notice that these approaches have some limitations, in particular:

- *Manual feature crafting.* One major drawback of traditional methods is that they heavily rely on handcrafted features, which require domain expertise and can be time-consuming to design. These features may not fully capture the complexities present in real-world data, limiting their ability to generalize well across diverse object variations.

- *Sliding window techniques.* Traditional approaches often use sliding window techniques to scan an image at multiple scales and locations to identify potential object regions. This process can be computationally expensive, especially when dealing with large-scale datasets or high-resolution images.

- *Limited robustness.* Traditional methods struggle with complex object configurations, occlusions, and variations in scale and viewpoint. They may not be well-suited for detecting objects in cluttered or challenging scenes.

Deep neural networks revolutionized the object detection domain, offering significant improvements in accuracy, speed, and robustness. The foundation of deep learning-based object detection lies in the ability to automatically learn hierarchical representations from data, eliminating the need for manual feature crafting that is prevalent in traditional approaches [84]. The main advantages of deep learning-based methods over traditional methods are:

- *Automatic feature learning.* Instead of relying on handcrafted features, deep learning methods learn features directly from the data, allowing them to adapt to a wide variety of object detection tasks and handle complex real-world data.
- *Unified framework.* These methods often operate in a unified framework where different components (feature extraction, object localization, and classification) are integrated and trained jointly, leading to superior performance and better optimization.
- *Robustness to variations.* Deep learning models are robust to a wider range of variations in object configurations, scales, and viewpoints. They can effectively handle occlusions and detect objects in cluttered or challenging scenes.
- *Scalability.* The computational efficiency and scalability of deep learning models, especially one-stage detectors, make them suitable for real-time object detection applications and large-scale datasets.

Despite these advantages, it should be noted that deep learning-based methods are data-intensive, requiring large labeled datasets and substantial computational resources for training. This motivates the use of intelligent sampling strategies based on active learning to reduce the need for unnecessarily large training sets.

5.1.2 Convolutional neural networks

Convolutional neural networks (CNNs) form the backbone of most of today's advanced object detection mechanisms. At the core of CNNs are convolutional layers, which employ filters (or kernels) to detect specific patterns or motifs in an image. These filters slide or 'convolve' across the input image, and at each position, a dot product is computed between the filter and the portion of the image it covers. If the filter and the image portion are similar, this dot product yields a high value, indicating a strong presence of the motif represented by the filter. CNNs can detect motifs anywhere in the input, allowing the system to recognize shapes or patterns irrespective of their position within the image [85]. For instance, consider the CNN depicted in Figure 5.1, which is designed to distinguish between the letters 'C' and 'D'. Intuitively, the letter 'C' can be characterized by the presence of open-end segments, while the letter 'D' has a closed curve with two corners. By employing filters that can detect these unique features, the CNN can classify the image based on the presence or absence of these detected motifs.

Convolution operations can be thought of as local feature detectors that are shift equivariant. If we shift an image and then apply a convolution operation, the result is the same as if you had first applied the convolution and then shifted its output. We can observe this property is shown in Figure 5.2. The weighted sums in Figures 5.1 and 5.2 are called feature maps, and they are derived by sliding the filter across the image pixel-by-pixel, producing darker outputs where there is a significant match with the filter. In real-world applications of CNNs, the filters or templates are not

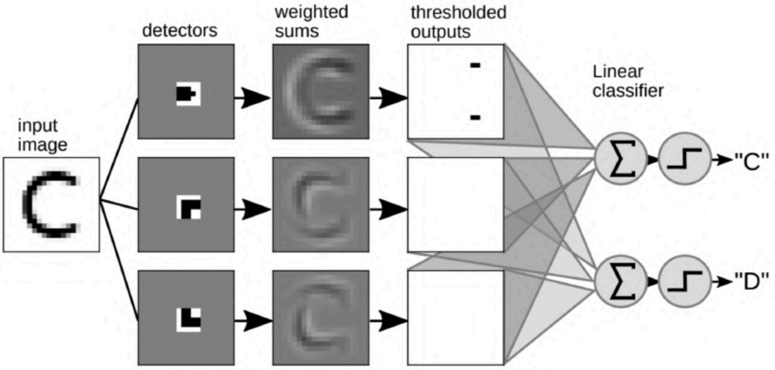


Figure 5.1. An illustrative example from [85] showcasing a hand-designed CNN for classifying images as either 'C' or 'D'. The detectors in this context represent filters that specifically seek out the presence of endpoints and corners in the image.

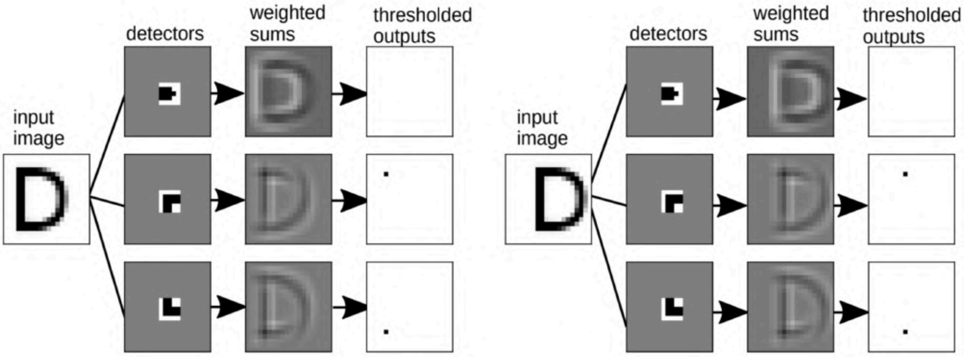


Figure 5.2. Shift equivariance property of the convolutional layers [85].

hand-designed as in the illustrative examples; instead, they are learned by the model during the training process. This is one of the primary advantages of deep learning: the ability of the model to learn appropriate features or patterns directly from the data without requiring manual feature engineering. In addition to the fundamental convolution operation, techniques such as padding and stride play important roles in the functioning of CNNs. Padding ensures the spatial dimensions of our image are retained post-convolution by adding a zero-value border around it. This way, the filter application preserves the original image size. On the other hand, stride dictates the movement of the filter across the image. Rather than the standard one-pixel shift, it allows for larger leaps, like two pixels at a time for a stride of two, making it possible to adjust the granularity of feature extraction. These parameters are crucial in fine-tuning CNN architectures for optimal depth and computational efficiency.

Following convolutional layers, pooling layers are often used to reduce the spatial dimensions of the feature maps obtained from convolutional layers. This downsampling serves a dual purpose: it reduces computational demands and also diminishes the risk of overfitting, ensuring the network remains both efficient and generalizable. Moreover, pooling introduces translational invariance to the architecture. This critical attribute ensures the network remains consistent in its recognition even if objects within the image undergo slight positional shifts or distortions. Among the various types of pooling, max pooling is the most commonly employed. For each segment of the feature map, it only retains the highest value, contrasting with average pooling which retains the average value of a segment. These pooling operations often use a sliding window mechanism that traverses the feature map, with the stride of this window determining the degree of downsampling. In essence, while convolutional layers focus on feature extraction and recognizing intricate patterns, pooling layers compactly represent these patterns, ensuring resilience to minor spatial alterations, which is foundational for robust tasks like object detection.

In summary, the combination of convolutional layers and pooling layers creates an intricate map of feature representations at multiple levels of abstraction, laying the foundations for the object detection task [86].

5.1.3 Object detection metrics

Before diving into the complexities of object detection models, it is necessary to understand the metrics used to evaluate their performance. Unlike simple classification or regression tasks, object detection not only classifies objects but also locates them within an image using bounding boxes. This spatial aspect necessitates unique evaluation metrics. Here, we introduce foundational metrics that offer insights into the accuracy and robustness of object detectors.

A foundational metric in object detection is the intersection over union (IoU), also known as the Jaccard Index. It quantifies the overlap between two bounding boxes: the predicted box and the actual (ground truth) box. Specifically, IoU calculates the ratio of the area where the two boxes overlap (the intersection) to the area covered by both boxes combined (the union)

$$IoU = \frac{Area(Predicted \cap Ground\,Truth)}{Area(Predicted \cup Ground\,Truth)} \tag{5.1}$$

As Figure 5.3 illustrates, varying overlaps yield different IoU scores. For model evaluation, a threshold (often, IoU $>$ 0.5) determines whether a predicted box accurately detects an object (true positive) or misidentifies it (false positive). Adjusting this threshold lets us balance precision and recall, two key metrics in detection tasks.

Beyond IoU, Average precision (AP) offers a holistic view of an object detector's performance. AP evaluates both precision and recall across varying IoU thresholds, effectively measuring a model's consistency in detection accuracy. By charting precision against recall for IoU thresholds ranging usually from 0.5 to 0.95, and then

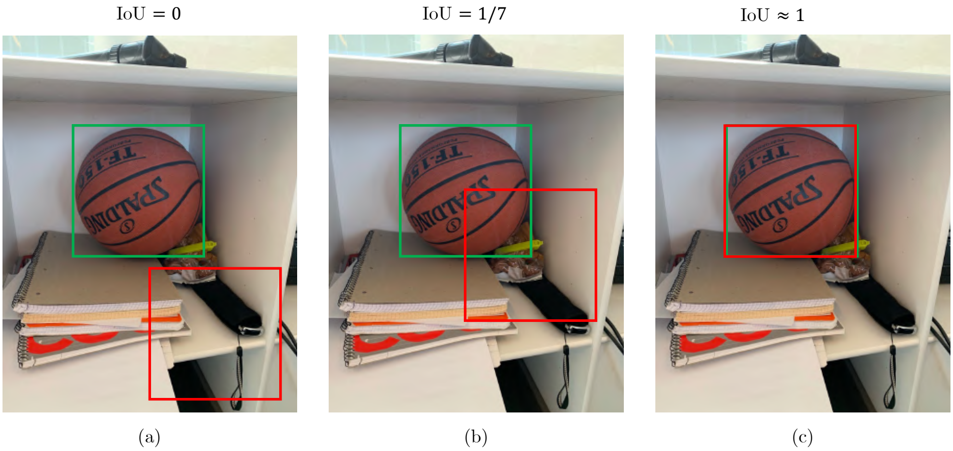


Figure 5.3. Different IoU scores obtained on the same image. In this example, the green bounding box represents the ground truth, and the red bounding box represents the prediction.

calculating the area under this curve, AP provides a singular metric capturing detection efficacy across diverse scenarios. Mean AP (mAP) is an extension of AP that calculates the average AP across multiple object categories or classes. In multi-class object detection tasks, there is a separate AP computed for each class, and the mAP is the average of these AP values. It is a useful metric for evaluating the overall performance of an object detection system when dealing with multiple object categories.

5.1.4 YOLO object detector

Object detectors based on CNNs can be broadly divided into two categories: two-stage detectors, which initially propose potential object regions and subsequently refine these proposals in a separate phase; and one-stage detectors, which perform object location prediction and classification simultaneously in a single step [87]. The dual-phase approach of two-stage detectors tends to make them more accurate than one-stage detectors, but also more computationally intensive. The most popular two-stage detector is Faster R-CNN [88]. In its first phase, a region proposal network locates a predefined number of regions that may contain objects. These sparse region proposals are represented by bounding boxes, which usually have an associated score representing the probability of an object being within it. These bounding boxes are then used to obtain classification scores and the spatial offsets [89]. Faster R-CNN's two-phase nature allows high predictive accuracy, especially in scenes with intricate object configurations and occlusions. However, this comes at the cost of reduced speed during inference compared to one-stage methods. Instead, one-stage object detectors aim to streamline the detection process by predicting the class and bounding box of objects in a singular pass through the network. Their architecture is inherently faster, making them particularly suitable for real-time applications. A notable example is the single shot multibox detector (SSD), which operates on an array of feature maps extracted from the input image [90]. This multilayer approach enables the detection of objects across varying sizes. SSD employs default anchor boxes at each feature map cell, adjusting during training to match the actual objects in the images. While SSD is renowned for its speed, it can occasionally lack accuracy, especially for smaller objects. Another prominent one-stage detector is you only look once (YOLO) [91]. This is the model[1] we used in Paper 6 (Appendix F), for its high accuracy coupled with rapid inference capabilities. This choice was particularly pertinent for our objective of real-time monitoring of streams of images collected from closed-circuit television (CCTV) cameras. An essential component in our sampling strategy was the confidence score tied to the bounding boxes predicted by YOLO.

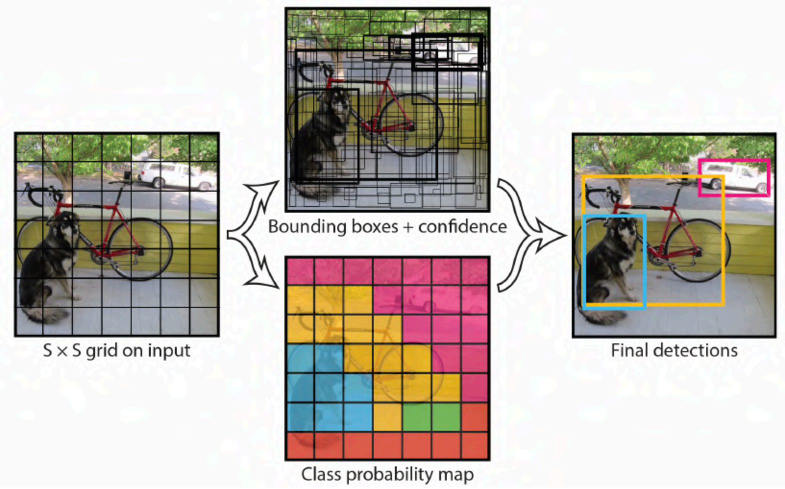


Figure 5.4. Grid-based prediction process [91].

The complete grid-based prediction framework utilized by YOLO is depicted in Figure 5.4. It works by dividing the image into an $S \times S$ grid. A grid cell is given the job of identifying the object and drawing a bounding box around it only if the center of the object falls within it. Other cells might also make predictions, but the cell where the object center lies has the primary responsibility. Each grid cell predicts B bounding boxes, confidence scores for those boxes, and C class probabilities. It should be noted that both S and B are typically chosen based on the nature of the dataset and the specific requirements of the application. For datasets where objects are generally large and centered in the image, a smaller S might be sufficient, while

[1] we used YOLOv8, the latest available version available at the time.

for datasets with many small objects or overlapping objects, a larger S and B might be more appropriate to capture all the objects accurately. In the original YOLO paper [91], the authors chose $S = 7$ and $B = 2$, meaning that the image is divided into a 7×7 grid, and each cell predicts two bounding boxes. Each bounding box is characterized by five predictions: x, y, w, h, and the confidence. The coordinates (x, y) represent the center of the bounding box, while w and h are, respectively, the width and height relative to the whole image. The confidence score associated with a bounding box is intended to encapsulate two core notions, how confident the model is that the box contains an object and how accurate the model believes its predicted box is. Formally, they defined it as $P(\text{Object}) * IoU_{pred}^{truth}$. If there are no objects in the grid cell, the confidence should be zero. Conversely, the confidence score should be equal to the IoU between the predicted box and the ground truth. Additionally, every grid cell predicts C conditional class-specific probabilities $P(Class_i|Object)$ using a softmax activation over the network outputs for the C class predictions.

Combining the four bounding box coordinates, the confidence, and the class probabilities, the model produces $B \times 5 + C$ values for each grid cell. Thus, throughout the whole grid, the model produces a tensor of shape $S \times S \times (B \times 5 + C)$. For each predicted bounding box, the model confidence score is multiplied by the class probabilities. This results in $B \times C$ class-specific confidence scores for each grid cell. These scores represent the likelihood that the bounding box contains an object of each specific class. Often, many of the predicted bounding boxes have low confidence scores, indicating that they likely don't contain any object. To reduce the number of bounding boxes, a confidence threshold is applied, and boxes with scores below this threshold are discarded. Then, non-maximum suppression (NMS) is applied to further prune the bounding boxes. This process involves sorting all remaining bounding boxes by their confidence scores and taking the box with the highest score and removing any other box that has a high overlap (measured by IoU) with it. These two steps are repeated until all boxes have either been kept or discarded. The NMS process ensures that we are left with only the most confident bounding box for each detected object. After thresholding and NMS, we are left with a set of bounding boxes that the model believes most accurately represents the objects in the image. Each of these boxes is associated with a class label (determined by the class with the highest class-specific confidence score for that box) and a confidence score (the aforementioned class-specific confidence score for the determined class). This extraction process transforms the dense output tensor $S \times S \times (B \times 5 + C)$ into a sparse list of bounding boxes, each with an associated class label and confidence score.

5.1.5 Fine-tuning object detection models

When fine-tuning object detection models, there are generally two approaches:

1. *Layer freezing.* In this approach, we take a pre-trained model (usually trained on a large dataset like COCO or ImageNet) and freeze some of its layers, typically the early layers. This means that the weights of these layers will not be updated

during fine-tuning. The rationale is that the early layers capture generic features (e.g., edges, textures) which are common across many tasks. Thus, there is often no need to retrain them. Only the later layers, which are more task-specific, are fine-tuned on the new dataset. A common strategy is to replace the last few layers (e.g., the softmax layer) with new layers that are initialized randomly and trained on the new dataset.

2. *Initialization.* In this approach, we initialize the model with weights from a pre-trained model, but we do not freeze them. All layers are fine-tuned using the new data. This method allows the model to adjust all of its weights based on the new data. This can be especially useful if the new dataset is quite different from the dataset the model was originally trained on.

Which approach to choose depends on several factors. If the dataset is small, we are more prone to overfitting, so freezing early layers might be beneficial. However, if the new data is significantly different from the original dataset, fine-tuning all layers might yield better results. It should be noted that fine-tuning all layers requires more computational resources than just training a few layers. If we are looking for the highest accuracy possible and have a large dataset, fine-tuning all layers might be the best approach. On the other hand, if we need a quick and versatile solution, transfer learning with layer freezing might suffice. In practice, a combination of both approaches could also be used. For example, one might start with layer freezing, and if results are not satisfactory, proceed to fine-tune all layers. In the stream-based active distillation framework proposed in PAPER 6 (Appendix F) we fine-tuned all the layers of the models, given the high dissimilarity of the street views from the dataset used for the pre-training (COCO).

5.1.6 Relevance of active learning for object detection

The development of sophisticated object detection models presents two specific challenges that are intrinsically related to active learning. The first one is the necessity for these models to be trained on voluminous annotated datasets, which can be expensive and time-consuming to obtain. In particular, the average price to obtain human-annotated images ranges from \$0.25 to \$7 per image [92]. This can easily make the development costs skyrocket if we consider that some models require thousands of labeled examples to achieve satisfactory performances. The second challenge revolves around the generalization of these models to unseen data when they are deployed to new environments. The idea of a ‘one model fits all’ solution is now becoming old-fashioned, ushering in a renewed interest in finding efficient ways to fine-tune models for specific scenes or applications. However, fine-tuning object detection models when obtaining labels is expensive becomes a difficult task. In light of this, the ensuing chapter delves deep into real-time sampling strategies that synergize active learning with knowledge distillation, aiming to substantially reduce the needed number of (pseudo) labeled examples for fine-tuning.

5.2 Problem statement

The problem addressed in Paper 6 can be seen as an extension of the methodology developed in Chapter 3, applied to scenarios involving object detection models. The significant innovation is the formulation of the stream-based active distillation framework, as illustrated in Figure 5.5. In this framework, the primary goal is to fine-tune student object detectors using pseudo-labels provided by a large, general-purpose model. Typically, student models are compact versions of the teacher model that can be deployed with reduced resource requirements. The key distinctions from the stream-based active learning framework presented in Figure 3.1 include:

- The learning model guiding the data collection scheme and being fine-tuned is a pre-trained object detection model and not a linear regression model.
- The oracle providing labels for model updates is not a perfect entity like a human annotator, but a large pre-trained model. Therefore, the labels from the oracle should be regarded as imperfect pseudo-labels, not ground truth.
- The model is not updated at each iteration. Frames selected for fine-tuning are chosen in a streaming fashion, but the model is updated only when a batch of frames is accumulated. This approach aligns with the batch-mode active learning paradigm [93], which indicates that updating large deep neural networks each time a new labeled example is made available is inefficient.

5.3 Proposed methodology

In Paper 6, we develop a sampling strategy that leverages the confidence scores provided by the student model at inference time. It is important to consider that the limited resource we aim to minimize is the number of pseudo-labels requested from the teacher model. We can freely ask the student model to detect objects in incoming frames without incurring any cost. As discussed in Section 5.1.4, for each image, the model generates bounding boxes for identified instances, accompanied by a confidence score for each box. Therefore, we can select the most informative frames by evaluating the confidence of the student model in detecting objects in each frame. A direct approach might be to adopt a least confidence strategy, under the assumption that the model would benefit most from data points with high uncertainty (a typical strategy in active learning [37]). However, since we are not working with a perfect oracle, this approach risks encountering confirmation bias [94]. This issue arises when a data point uncertain for the student is likely also uncertain for the teacher, leading to inaccurate pseudo-labeling. Therefore, we propose querying frames with high confidence scores, hoping they will be accurately pseudo-labeled by the teacher. We prioritize a simpler but correctly labeled training set over a more complex but inaccurately labeled one. The key contributions of the paper are:

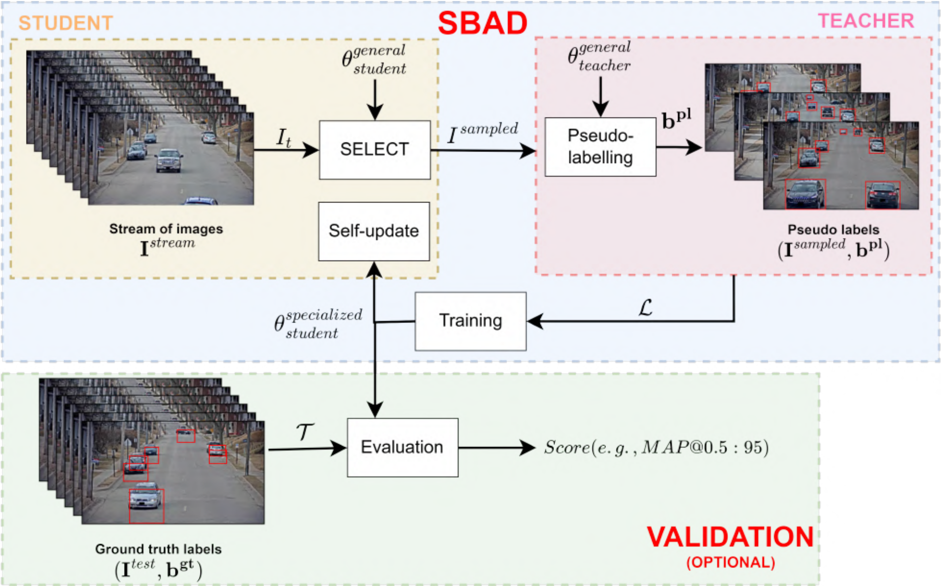

STUDENT
SBAD
TEACHER
$\theta^{general}_{student}$
I_t
SELECT
$I^{sampled}$
$\theta^{general}_{teacher}$
Pseudo-labelling
$\mathbf{b}^{\mathbf{pl}}$
Stream of images
$\mathbf{I}^{stream}$
Self-update
Pseudo labels
$(\mathbf{I}^{sampled}, \mathbf{b}^{\mathbf{pl}})$
$\theta^{specialized}_{student}$
Training
$\mathcal{L}$
$\mathcal{T}$
Evaluation
$Score(e.g., MAP@0.5:95)$
Ground truth labels
$(\mathbf{I}^{test}, \mathbf{b}^{\mathbf{gt}})$
VALIDATION
(OPTIONAL)

Figure 5.5. Stream-based active distillation framework (from Paper 6).

- The introduction of the stream-based active distillation framework, conceptualized for fine-tuning object detection models with pseudo-labeled frames selected in real-time.

- An assessment of the effectiveness of confidence-based sampling strategies (top confidence and least confidence), compared to more straightforward methods such as random sampling and the N-first approach, which samples the initial N images in the stream.

- The application of this framework to a real-world dataset sourced from CCTV cameras in the United States.

5.4 Key results

The most notable result from Paper 6 is the enhanced learning efficiency observed when implementing the top confidence sampling strategy. As shown in Figure 5.6, this approach enables the student model to closely match the performance of the teacher model using only 250 pseudo-labeled frames. This outcome is particularly significant as it suggests the possibility of fine-tuning a compact model to achieve performance comparable to that of a larger model without requiring human annotation.

The plot also shows that the initial performance of the models is lower than that of the original pre-trained student model before fine-tuning. This initial decrease can

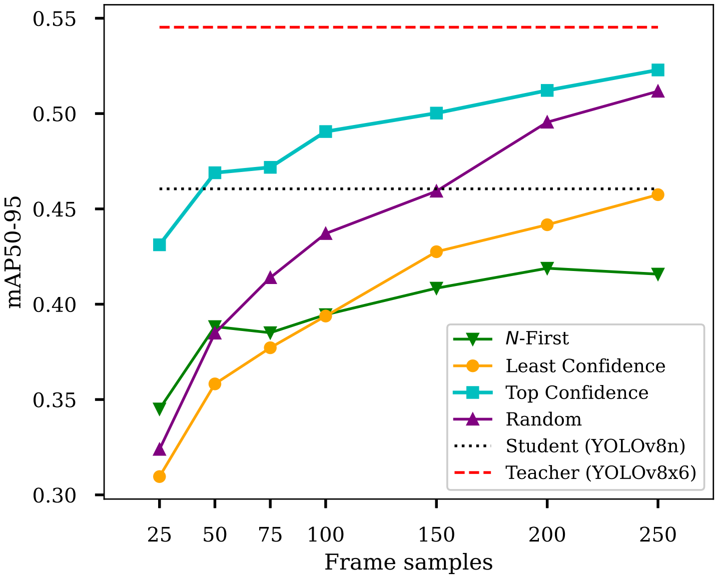


Figure 5.6. Learning curves of the stream-based active distillation framework (from Paper 6).

be attributed to the use of the initialization approach rather than the layer freezing method for fine-tuning, as discussed in Section 5.1.5. Consequently, in the early stages of the learning process, where the models are fine-tuned with a limited number of frames, there is a higher risk of overfitting to this small training set. However, the strategy proves to be effective as the number of frames increases.

5.5 Additional results

In this section, we provide additional results and analyses that are not included in Paper 6.

Confidence aggregation operators. For object detection models, the presence of multiple objects in the same image significantly increases the complexity of determining an image-level confidence score. Indeed, different aggregation operators can be used when performing active learning with object detection, in order to compute a unique score for each image from the individual detection metrics. Let us assume that, on the image i, an object detector identified the set of bounding boxes $\{b_i\}$ for each class c. Then, aggregating by maximum, we can define the confidence of the model with respect to the image i as

$$\max_c \max_{j \in \{b_{ic}\}} p(j) \tag{5.2}$$

where $p(j)$ is the probability or confidence score that the bounding box j belongs to the class c. Aggregation by max has been used by Roy et al. [95]. Additionally, aggregation functions like sum and average can also be used [96].

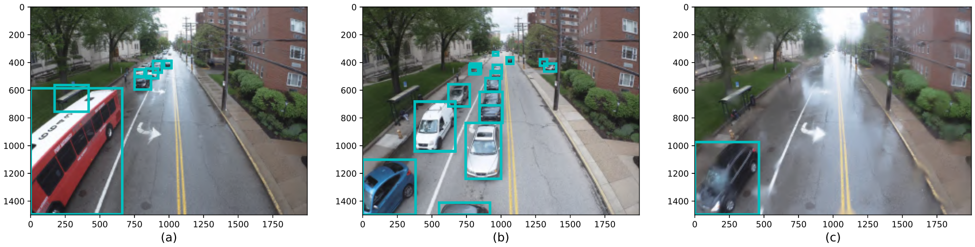


Figure 5.7. Most confident images according to different aggregation functions: max (a), sum (b), and average (c).

In Figure 5.7, we observe the impact of employing different aggregation metrics to determine the most confident image for an object detector. The images refer to the first camera of the WALT dataset [97]. When utilizing the max as the aggregation function, the selected image is the one where the model identifies a single object, with a substantially high confidence score (e.g., the bus). However, this approach may overlook other objects for which the model is not as confident (e.g., the bench). Alternatively, following the suggestion of [96], when we use the sum as the aggregation function, images with numerous objects receive high scores, regardless of the individual bounding box confidence scores. This approach might not accurately represent the confidence in each object but focuses on the total number of objects in the image. Finally, using the average as the aggregation function yields two possible outcomes. It can either result in an image containing multiple objects with a high average score or as observed in this case, it may select images with only one object that has a remarkably high confidence score. In summary, the choice of aggregation metric has a substantial impact on the selection of the most confident image, and each metric has its advantages and limitations when considering the confidence scores of objects in the image.

Confirmation bias. Confirmation bias refers to a situation where a student performance declines due to the training on incorrect pseudo labels provided by the teacher. This inaccuracy arises from the disparity between the teacher predictions for unlabeled frames and the actual ground truth labels. When the student is fine-tuned using these incorrect pseudo labels, it becomes more confident in making incorrect predictions. Within the stream-based active distillation framework, the confidence level plays a pivotal role in the selection of samples to be pseudo-labeled by the student. Specifically, opting to sample challenging data points where the student is least confident proves counterproductive. Indeed, as pointed out by Baykal et al. [98], hard instances for the student are often hard to predict correctly by the teacher, making the pseudo labels for these points more likely to deviate from the actual labels. This discrepancy can mislead the student during training. Given this insight, we explore the effectiveness of the least confidence and top confidence approaches, in providing accurately labeled training sets for the student models. In particular, we argue that least confidence will provide the student with less accurate training sets. To validate our hypothesis, we manually labeled the pseudo-labeled training sets and computed their mAP scores.

Table 5.1. Training set reliability (mAP)

Camera	**Top confidence**	**Least confidence**
1	0.586	0.199
2	0.589	0.284

Table 5.1 reports the quality of the pseudo labels provided by the teacher according to the different sampling strategies. These results are obtained by evaluating the performance of the teacher when 96 frames are sampled by the student according to the two sampling strategies. It can be seen how the gap between the learning curves obtained by the student with the two sampling strategies is highly supported by the quality of the pseudo labels used for the fine-tuning. Indeed, we can confirm that when the students query examples that are highly uncertain by minimizing the confidence associated with the selected images, the teacher is not able to provide accurate labels. Figure 5.8 shows a specific example that explains the poor performance of least confidence on the first camera. This particular scene (e.g., dark with few to no cars) is deemed highly uncertain by the students and, when it is queried, the teacher model that has not been fine-tuned for this camera is not able to detect the parked cars on the right. This missed detection is one of the examples highlighting the poor performance of least confidence.

To provide an additional view on confirmation bias, we explore the performance of the student model on the test set when the training set is labeled by teachers of different sizes, and by a human annotator. To avoid introducing biases in the evaluation, the annotator for the training set and the test set were the same. Table 5.2 provides an overview of the percentage improvement that is observed by switching from least confidence to top confidence. The YOLOv8 models are shown in decreasing order of size. On average, we can see that the smaller the model becomes the more the confirmation bias is pronounced. This reinforces the idea that when querying uncertain images for the students, imperfect oracles might provide inaccurate pseudo-labels. However, it should be noted that even in the case of a human annotator there remains a clear improvement with the use of top confidence. This could be explained by the fact that the high-confidence images selected with this strategy tend to be frames taken during the day, in broad light, when more instances are present in the images. This allows the model to get more easily specialized by seeing more objects.

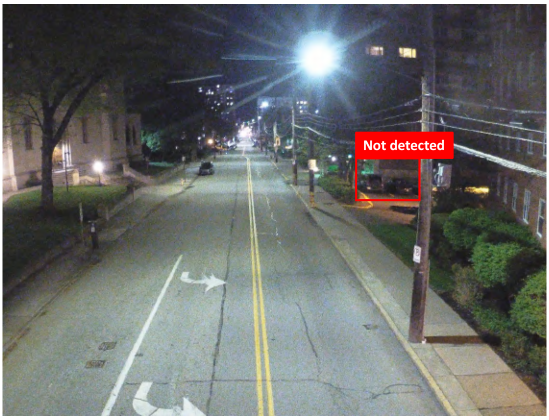


Figure 5.8. An example of hard-to-label image sampled using least confidence.

Table 5.2. Performance Difference (% mAP)

Oracle	Cam 1	Cam 2	Average
Human	21.40%	32.67%	27.03%
YOLOv8x6	34.62%	36.30%	35.46%
YOLOv8l	40.14%	33.79%	36.97%
YOLOv8m	53.27%	33.88%	43.58%
YOLOv8s	62.96%	38.63%	50.80%
YOLOv8n	110.25%	42.13%	76.19%

Clustering-based distillation. In the extension of this paper, we are exploring how to improve the efficiency of the stream-based active distillation framework when many cameras are considered. In particular, we are investigating the transferability of the trained models between different cameras and scene clustering in order to reduce the number of models required. Additionally, we are also trying to examine the relationship between model size and detection accuracy.

5.6 Discussion

The stream-based active distillation framework presents a compelling approach for fine-tuning object detection models, especially when faced with limited operational resources, such as storage and hardware constraints. However, several aspects merit further exploration to enhance this approach:

- *Theoretical foundations.* While the results are promising, the proposed approach lacks robust theoretical underpinnings that explain why the top confidence approach yields better performances compared to other strategies.

- *Diverse sampling strategies.* The current assessment primarily focuses on simple model confidence-based sampling strategies. Ideally, more intricate strategies that also consider the information at the image level and the temporal correlation between images could be explored to minimize redundant queries.
- *Varied datasets.* Although the methods are tested on real-world data, examining their performance across datasets from different contexts, such as sports scenes or other CCTV scenarios, would provide broader insights into their applicability and effectiveness.

CHAPTER 6

Conclusion

6.1 Summary

This thesis has primarily concentrated on advancing the field of stream-based active learning, a critical area in machine learning that addresses the challenge of label scarcity in data streams. Our exploration has been centered around developing and refining sampling strategies that can be used to prioritize data labeling in data streams. These methodologies diverge from traditional pool-based active learning, adapting to the unique characteristics of data streams where data points arrive sequentially and decision-making must be done in real-time. The key contributions of this thesis include:

- The development of the CDO algorithm tailored for stream-based active learning with linear regression models. This approach showcases a significant reduction in labeling costs while maintaining model accuracy.

- Addressing specific challenges in stream-based active learning, such as robustness against outliers and effective feature selection, to enhance the applicability and reliability of active learning in dynamic data environments. Additionally, we explored the use of methods to approximate the learning curve, with the potential to move from a fixed-budget setting to a performance-based stopping criterion.

- Conducting exploratory studies on the behavior of regression models under the influence of concept drift in data streams, laying a foundation for future in-depth research in this area. In particular, we propose a residual-weighted adaptive sampling strategy that showed promising results in detecting localized concept drifts.

- Introducing a stream-based active distillation framework, a novel approach to efficiently train object detection models using strategically selected pseudo-labeled frames.

These findings significantly contribute to the field of stream-based active learning, offering practical solutions and novel insights for handling label scarcity in data streams.

6.2 Limitations

Reflecting on the broader implications and challenges encountered in this thesis, several key themes and considerations emerge, shaping the future trajectory of research in stream-based active learning.

- *Adaptability and scope of models.* A recurring theme in our work is the balance between model specificity and generality. While tailored solutions for linear models showed promise, their applicability to diverse, real-world scenarios with complex data structures remains a question. This highlights the need for model-agnostic approaches that can adapt to varying data landscapes and model complexities.

- *Methodological complexities.* The introduction of advanced techniques like KDE and robust estimators significantly enhanced our methodologies. However, these additions also brought about increased computational complexities and nuanced parameter tuning challenges. This raises a crucial point about the trade-off between methodological sophistication and practical usability, especially in fast-paced industrial environments.

- *Defining and addressing data anomalies.* Our work touched upon the critical aspect of outliers and concept drifts, which are pivotal in stream-based scenarios. However, the multifaceted nature of these anomalies, ranging from simple outliers to complex drift patterns, calls for a more detailed understanding and approach. Future research could benefit from a deeper dive into the nature of these anomalies and the development of more adaptive, context-aware strategies.

- *Theoretical foundations and practical implications.* While our approaches have shown promising results, the underlying theoretical foundations need further strengthening. This is particularly true for strategies like top confidence in active distillation, where empirical success prompts a deeper theoretical inquiry. Additionally, the practical implications and feasibility of deploying these strategies in real-world settings remain areas ripe for exploration.

- *Diversity in data and application scenarios.* Our investigations were constrained by the types and sources of data used. Diversifying the data sources and application scenarios, such as exploring different industrial domains or incorporating varying types of streaming data, can provide more comprehensive insights into the applicability and robustness of the proposed methods.

In summary, this thesis provides novel stream-based active learning approaches, while also opening avenues for substantial future research. The challenges and limitations encountered not only underscore the complexities inherent in this field but also highlight the dynamic nature of machine learning research, where every solution brings new questions and every finding leads to unexplored pathways.

6.3 Future research directions

Looking ahead, several promising research directions emerge that can further enhance and expand upon the work presented in this thesis:

1. *Advanced adaptive sampling techniques.* Developing more sophisticated adaptive sampling strategies that can intelligently respond to concept drifts in data streams, thereby improving the model's adaptability and predictive power.

2. *Clustered active distillation frameworks.* Investigating the balance between model specialization and the diversity of models in clustered active learning setups, with a focus on scene similarity and the effects of varying teacher model sizes.

3. *Expanding to complex model architectures.* Extending the proposed methodologies to accommodate more complex, non-linear models, thereby broadening the scope and applicability of stream-based active learning.

4. *Temporal dynamics and redundancy reduction.* Incorporating the temporal aspect of data streams to enhance the efficiency of the learning process and minimize redundancy in data selection.

5. *Cost-Sensitive Learning Approaches:* Integrating cost considerations into the sampling strategy, reflecting the variable nature of labeling costs in real-world scenarios.

6. *Deployment and real-world evaluations.* Applying and rigorously testing stream-based active learning methodologies in practical industrial contexts to assess their scalability, robustness, and economic viability.

By pursuing these future research directions, the field of stream-based active learning is poised to make significant strides, offering innovative solutions to address label scarcity and enhancing the performance and applicability of machine learning models in various real-world applications.

Appendices

APPENDIX A

Paper 1 – Sampling strategies for industrial applications through active learning

This paper is awaiting publicatin and is not included in NTNU Open

APPENDIX B

Paper 2 – Active learning for data streams: a survey

Machine Learning (2024) 113:185–239
https://doi.org/10.1007/s10994-023-06454-2

Active learning for data streams: a survey

Davide Cacciarelli[1,2] · Murat Kulahci[1,3]

Received: 10 May 2023 / Revised: 13 September 2023 / Accepted: 10 October 2023 /
Published online: 20 November 2023

Abstract

Online active learning is a paradigm in machine learning that aims to select the most informative data points to label from a data stream. The problem of minimizing the cost associated with collecting labeled observations has gained a lot of attention in recent years, particularly in real-world applications where data is only available in an unlabeled form. Annotating each observation can be time-consuming and costly, making it difficult to obtain large amounts of labeled data. To overcome this issue, many active learning strategies have been proposed in the last decades, aiming to select the most informative observations for labeling in order to improve the performance of machine learning models. These approaches can be broadly divided into two categories: static pool-based and stream-based active learning. Pool-based active learning involves selecting a subset of observations from a closed pool of unlabeled data, and it has been the focus of many surveys and literature reviews. However, the growing availability of data streams has led to an increase in the number of approaches that focus on online active learning, which involves continuously selecting and labeling observations as they arrive in a stream. This work aims to provide an overview of the most recently proposed approaches for selecting the most informative observations from data streams in real time. We review the various techniques that have been proposed and discuss their strengths and limitations, as well as the challenges and opportunities that exist in this area of research.

Keywords Stream-based active learning · Online active learning · Data streams · Online learning · Unlabeled data · Query strategies · Selective sampling · Concept drift · Experimental design · Bandits

Editor: Joao Gama.

✉ Davide Cacciarelli
dcac@dtu.dk

Murat Kulahci
muku@dtu.dk

[1] Department of Applied Mathematics and Computer Science, Technical University of Denmark, Kgs. Lyngby, Denmark

[2] Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway

[3] Department of Business Administration, Technology and Social Sciences, Luleå University of Technology, Luleå, Sweden

1 Introduction

The deployment of machine learning models in real-world applications is often reliant on the availability of significant amounts of annotated data. While recent advancements in sensor technology have facilitated the collection of larger amounts of data, this data is not always labeled and ready for use in training models. Indeed, the process of obtaining labeled observations for supervised learning models can be cost-prohibitive and time-consuming, as it often requires quality inspections or manual annotation. In such cases, active learning proves to be a valuable strategy to identify the most informative data points for use in training, thereby reducing the overall cost of labeling and improving the performance of the model. Over the years, a plethora of active learning approaches have been proposed in the literature, each with its own benefits and limitations. These approaches seek to strike a balance between the cost of labeling and the quality of the model by selectively choosing the most informative observations for querying. By carefully selecting the most informative observations, active learning helps to minimize the amount of labeled data required and streamlines the learning process, contributing to its overall efficiency.

While several surveys have been published on pool-based active learning (Aggarwal et al., 2014; Settles, 2009; Fu et al., 2013; Kumar & Gupta, 2020), which involves selecting a fixed set of observations from a pool of unlabeled data, the dynamic and sequential nature of many real-world problems often renders these approaches impractical. This has led to growing interest in the online variant of active learning, also referred to as stream-based active learning, which involves continuously selecting and labeling observations as they arrive in a stream, allowing for real-time adaptation to changing data distributions. Lughofer (2017) provided a review of online active learning approaches with a focus on fuzzy models. However, since its publication, numerous other online active learning approaches have been proposed, and to the best of our knowledge, no other surveys have been published to synthesize these developments. Moreover, surveys purely focusing on online learning from data streams (Lu et al., 2018; Tieppo et al., 2022; Lima et al., 2022; Hoi et al., 2021) discuss methods that assume a complete availability of labels, which is not the case in many real-world applications. The aim of this review is to fill this gap by providing a comprehensive overview[1] of the most recently developed query strategies for online active learning. It is worth noting that in certain cases, stream-based active learning is narrowly defined as the act of selecting the most informative observations from a data stream to fit a predictive model. Instead, the act of determining which observations to query while making predictions is referred to as online selective sampling (Hanneke & Yang, 2021). In this work, we cover and examine all the methods that address the crucial problem of selecting the most informative data points to label from a data stream in an online fashion. We will present the techniques that have been proposed so far, discussing their strengths and limitations, as well as the challenges and opportunities that exist in this field. In addition, we will provide an overview of evaluation strategies for online active learning algorithms and highlight some real-world applications. Finally, we will identify potential future research directions in this area.

[1] We conducted a search on SCOPUS and Google Scholar using the following keywords: “on-line active learning”, “online active learning”, “stream-based active learning”, “single pass active learning”, “online selective sampling”, “sequential selective sampling”, and “active learning” combined with “data stream”. Each paper was reviewed individually to determine its relevance to online active learning. We eliminated irrelevant papers and manually added some papers that did not contain these keywords but used online active learning methods or were relevant to our discussion. Additionally, we included related papers that were necessary to understand the bigger picture from the references of the reviewed strategies.

This survey comprehensively explores various facets of active learning, encompassing both theoretical foundations and practical challenges. By delving into this review, we aim to shed light on pertinent research questions, including:

1. *Query strategy.* What sampling strategy should be used to maximize learning efficiency in a streaming context?
2. *Timing of queries.* When and how often should data points be queried to balance learning and resource constraints?
3. *Model updates.* When should predictive models be updated and how can they adapt to changing data distributions and concept drift?
4. *Scalability.* How can active learning methods be made scalable and efficient for high-velocity data streams?
5. *Evaluation.* What are appropriate evaluation metrics for assessing the performance of stream-based active learning algorithms?

The structure of this paper is as follows. In Sect. 2, we provide an overview of active learning, including the main instance selection criteria, an overview of the main active learning scenarios, and the connection between active learning and semi-supervised learning. Section 3 represents the core of the review, with a brief overview of how online active learning approaches have been classified, followed by a detailed description of the state-of-the-art approaches. In Sect. 4, we examine evaluation strategies for online active learning algorithms. Section 5 highlights real-world applications and challenges. Section 6 provides a summary of the most common online active learning methods and highlights potential directions for future research. Finally, Sect. 7 provides conclusions and summarizes the key contributions of the review.

2 Preliminaries on active learning

In supervised learning, we seek to learn a function that can predict the output variable, also known as response, given a set of input variables, also known as covariates. This function is often learned by training a model on a labeled dataset that consists of a large number of input–output pairs. However, obtaining labeled examples is not always straightforward, and it may not be possible or practical to label all the available data. In these cases, active learning can be used to select a subset of the data for labeling in order to improve the performance of the model, when there is a budget constraint on the number of unlabeled observations that can be queried. Indeed, there are many examples of how a classification or regression model can achieve a performance that is similar to what can be achieved when all the labels are available, using only a small fraction of the available observations.

2.1 Instance selection criteria

The main challenge in active learning is deciding which data points to label. There are many strategies for selecting data points in active learning, and most of them can be associated with one of these groups:

- *Uncertainty-based query strategies.* These approaches focus on selecting data points that the model is least confident about, in order to reduce its uncertainty (Lu et al., 2016; Tong & Koller, 2002). When using classification models, the most widely used is the margin-based query strategy, where data points close to the decision boundary are selected (Roth & Small, 2006; Balcan et al., 2007).

- *Expected error or variance minimization.* These strategies estimate the future error or variance, when a newly labeled example is made available, and try to minimize it directly (Cohn et al., 1996; Roy & Mccallum, 2001).
- *Expected model change maximization.* This strategy involves selecting data points that would have the greatest impact on the estimate of the current model parameters if they were labeled and added to the training set (Cai et al., 2013).
- *Disagreement-based query strategies.* These approaches focus on selecting data points where there is disagreement among multiple models or experts (Hanneke, 2014; Wang, 2011; Steve & Liu, 2014; Sheng et al., 2008). One of the most common approaches that use an ensemble of models is query by committee (Seung et al., 1992; Freund et al., 1997; Burbidge et al., 2007), which uses an ensemble of models to identify instances where the models have conflicting predictions.
- *Diversity- and density-based approaches.* These methods exploit the structural information of the instances and try to select data points that are diverse and representative of the overall distribution of the data. One example of this approach is the use of Mahalanobis distance to seek observations that are far from the currently labeled data points (Ge, 2014; Cacciarelli et al., 2022a). Clustering may be applied to label representative data points (Nguyen & Smeulders, 2004; Min et al., 2020; Ienco et al., 2013), and graph-based methods can be employed to explore the structure information of labeled and unlabeled data points (Zhang et al., 2020b) or to build upon the semi-supervised label propagation strategy (Long et al., 2008).
- *Hybrid strategies.* These are active learning algorithms that combine multiple instance selection criteria (Donmez et al., 2007; Huang et al., 2014). For example, by combining margin-based sampling with clustering the learner can select the most uncertain observations within different areas of the input space.

By considering these different strategies, one can select the most appropriate approach for a given problem based on the characteristics of the data and the specific requirements of the application.

2.2 Active learning scenarios

Active learning can be broadly categorized into three macro scenarios, based on how the unlabeled instances are supplied to the learner and then selected to be labeled by an oracle. Regardless of the particular query strategy being employed, these macro scenarios provide a framework for understanding the flow of information and the decision-making steps involved in active learning. These scenarios serve as a high-level categorization of different methods for approaching the active learning problem, each with its own set of advantages and disadvantages depending on the specific use case. Understanding these macro scenarios is crucial for selecting the appropriate active learning technique for a particular problem and for comparing different active learning algorithms. In the next subsections, each of the three macro scenarios will be discussed.

2.2.1 Membership query synthesis active learning

This scenario represents the case when the learner is given complete freedom to ask for the label of any data point belonging to the input space or for a synthetically generated one.

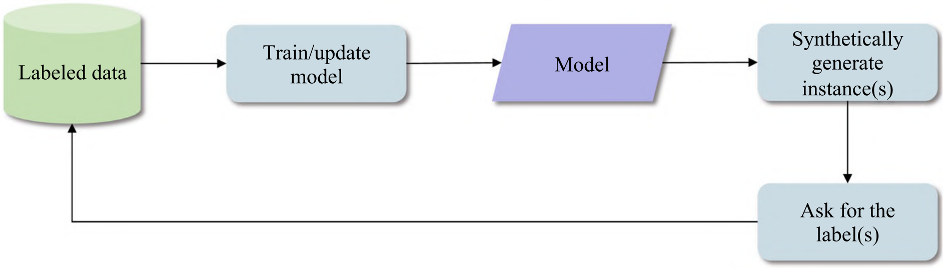


Fig. 1 Membership query synthesis active learning

Some examples of membership query synthesis active learning include image classification, where the learner can generate modified versions of existing images to be labeled, or object detection, where the learner can generate new instances by combining and transforming existing instances. In natural language processing (NLP) tasks such as text classification or sentiment analysis, the learner might generate synthetic examples in the form of sentences or paragraphs that cover a wider range of variations in the language. Also, in speech recognition, the learner might generate synthetic speech samples in different accents, pronunciations, or speaking styles in order to improve the recognition accuracy. However, as highlighted by Baum and Lang (1992) and Settles (2009), the main drawback of this strategy is that it could generate unlabeled examples for which no labels can be associated by a human annotator (e.g., a mixture between a number and a letter). A general flowchart for this scenario is reported in Fig. 1, where the scheme is repeated until a budget constraint on the requested labels is met, or a stopping criterion on the achieved performance is satisfied.

In the context of deep active learning (Ren et al., 2022), the membership query synthesis scenario can be addressed by using generative models. For instance, generative adversarial networks (GANs) have been used to generate additional instances from the input space that may provide more informative labels for the learner (Goodfellow et al., 2014). This can be done by using GANs for data augmentation, as GANs are capable of generating diverse and high-quality instances (Zhu & Bento, 2017). Another approach is to combine the use of variational autoencoders (VAEs) (Kingma & Welling, 2013) and Bayesian data augmentation, as demonstrated by Tran et al. (Tran et al., 2019, 2017). The authors used VAEs to generate instances from the disagreement regions between multiple models, and Bayesian data augmentation to incorporate the uncertainty of the generated instances in the learning process.

2.2.2 Pool-based active learning

Pool-based active learning is one of the most widely studied scenarios in the machine learning literature. The goal is to select the most informative subset of observations from a closed, static set of unlabeled data points. The majority of the proposed pool-based active learning approaches have been developed for classification tasks (Cai et al., 2013), with image classification being a common application in computer vision (Li & Guo, 2013), as manually labeling large image datasets can be a challenging task.

The flowchart in Fig. 2 provides an overview of pool-based active learning sampling schemes, where k represents the number of unlabeled instances whose label is queried at each round. Traditional machine learning models that do not require substantial computational resources to train are typically associated with a choice of k equal to one (Vahdat et al.,

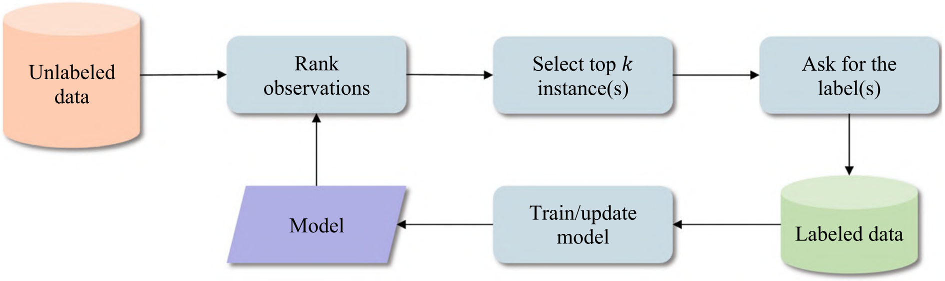


Fig. 2 Pool-based active learning

2019). This allows a timely update of the instance selection criteria, avoiding the redundant labeling of similar data points. However, larger values of k have also been used in practice, such as the analysis performed by Ge (2014) for values ranging from 5 to 30 or the approach used by Cai et al. (2013) to add 3% of the total number of observations to the training set each time. Using a higher k value may be more practical when working with large models, as repeated training can be computationally expensive and challenging. To this extent, batch mode active learning is generally considered to be a more efficient and effective option for image classification or detection tasks compared to the one-by-one query strategy, as the latter can be resource-intensive and time-consuming when working with large neural networks (Ren et al., 2022). This is because re-training the model with just one new data point with high input dimensionality may not result in significant improvement (Ren et al., 2022). In general, the choice of k may be problem- or model-specific, as it represents a trade-off between computational efficiency and the risk of querying redundant labels.

To enhance pool-based active learning, many approaches combine uncertainty-based instance selection criteria with acquisition functions such as entropy (Shannon, 1948; Wu et al., 2022), mutual information (Haussmann et al., 2020), or variation ratio (Schmidt et al., 2020). Entropy is commonly used as an acquisition function in active learning because it provides a way to measure the uncertainty of the model predictions for a given data point. The entropy of a probability distribution is a measure of the amount of disorder or randomness in the distribution. In the context of active learning, the entropy of a model's predicted class probabilities for a data point can be used as a measure of the model's uncertainty about the correct class label for that data point. Acquiring examples with the highest uncertainty is one way to select data points for annotation, but it is not the only way. Mutual information and variation ratio can also be used on the predictions obtained with the current model, in order to seek a diverse set of data points for which the predictions are the most uncertain. For a more comprehensive discussion on pool-based active learning, readers are referred to the surveys (Aggarwal et al., 2014; Settles, 2009; Fu et al., 2013; Kumar & Gupta, 2020).

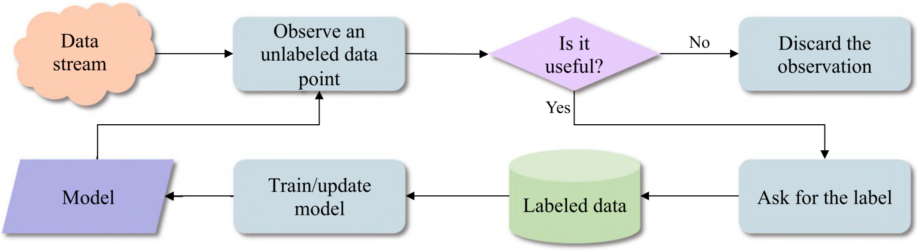


Fig. 3 Single-pass online active learning

2.2.3 Online active learning

In this type of active learning, we cannot greedily select the most informative observations from a static pool, as the instances are generated in a continuous stream and cannot be stored in their entirety before a decision is made. This is similar to the famous statistical puzzle known as the secretary problem (Freeman, 1983), where a hiring manager must make a hiring decision for each applicant as they are interviewed, without the benefit of seeing all applicants first. In general, online active learning is a crucial scenario for various real-world applications where the ability to make a sampling decision in real-time is of utmost importance. A few examples are:

- *Chemical or manufacturing processes.* In these applications, a learner is tasked with predicting the quality of the final product but may only have a short timeframe to make the sampling decision, to avoid traceability issues, particularly in high-volume production (Schmitt et al., 2013; Lieber et al., 2012). Also, tasks like predictive maintenance and visual inspection might benefit from a real-time selection of new examples to be labeled and included in the training set (Rožanec et al., 2022).
- *Video streaming and clinical trials.* In these cases, a decision must be made on the fly, as users arrive or volunteers appear sequentially, and there may not be enough time to accumulate a pool of potential users or patients (Fowler et al., 2023; Riquelme, 2017).
- *Text classification:* In NLP, online active learning can be used for tasks such as sentiment analysis and spam detection, where the learner continuously learns from new incoming data points which need to be labeled to update the model in real-time and improve accuracy (Kranjc et al., 2015).
- *Fraud detection.* To effectively detect fraudulent activities, the learner must continuously select new examples to label so that it can continuously update its decision-making process (Carcillo et al., 2018, 2017).
- *Online customer service.* Online customer service agents can use online active learning to improve their performance by continuously learning from customer interactions. To do this, the learner must continuously select new examples to label or customer information to obtain, so that it can predict the best response based on past interactions and improve its accuracy over time (Zheng & Padmanabhan, 2006).
- *Marketing.* Online active learning could also be applied in the field of marketing to select informative examples in real-time and continuously optimize customer targeting and personalization (Carnein & Trautmann, 2019; Jamil & Khan, 2016).

One of the defining features of online active learning strategies is their data processing capabilities. Figures 3 and 4 provide a visual representation of the two main approaches;

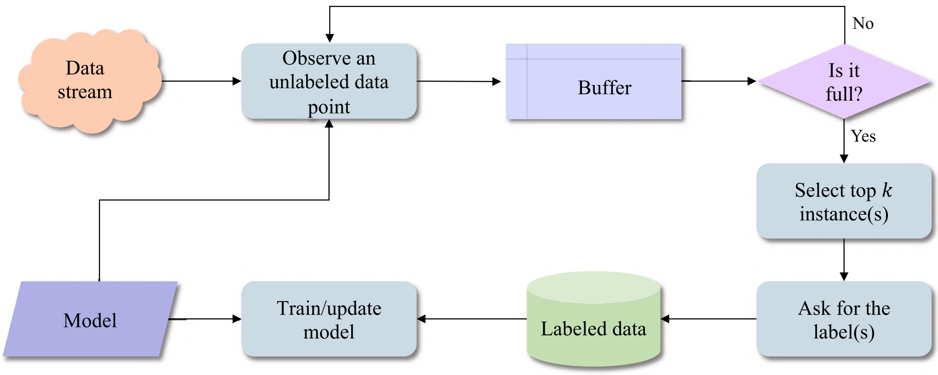


Fig. 4 Window- or batch-based online active learning

single-pass and window-based. Single-pass algorithms observe and evaluate each incoming data point on the fly, whereas window-based algorithms, also referred to as batch-based methods, observe a fixed-size chunk of data at a time. In this approach, the learner evaluates the entire batch of data and selects the top k observations as the most informative ones to be labeled. This approach is referred to as best-out-of-window sampling. The specific value of k and the dimensionality of the buffer can vary based on the storage capabilities of the system and the computational time required to update the model. Window-based methods are useful in situations where data is generated in large quantities and the algorithm does not have a tight constraint on the time available for decision-making. In contrast, single-pass methods are necessary when the algorithm needs to make a decision immediately after observing a specific data point.

Another critical property in the design of an effective online active learning strategy is the assumption made about the data stream distribution. One important difference to consider is whether the data stream is stationary or drifting. A stationary data stream is characterized by a stable data generating process where the statistical properties of the data distribution that remain constant over time. Conversely, a drifting data stream is marked by changing statistical properties of the data distribution over time, potentially due to alterations in the underlying data generating process. The distinction between stationary and drifting data streams is significant because it affects the performance of the active learning strategies. Online active learning strategies that have been developed for stationary data streams may lead to suboptimal performance when applied to drifting data streams. This is because concept drift can alter the scale of the informativeness measure of unlabeled data points or even urge a complete change of the model, with the acquisition of more observations to accommodate the new concept. Therefore, it is important to accurately assess the nature of the data stream distribution in the design of an active learning strategy. A failure to do so can result in a suboptimal performance and a reduced ability to effectively leverage the strengths of active learning. Another important property to consider when designing an active learning strategy is the label delay or verification latency. This refers to the time needed by the oracle to provide the label when it is requested by the learner. In some cases, there may be a delay L in the oracle providing the label after it has been requested. This property must be taken into account when designing a sampling strategy as there may be redundant label requests for similar instances if this issue is not properly addressed. Label delay can be classified into null latency, intermediate latency, or extreme latency (Souza et al., 2018). The case with null latency, or immediate availability of the label upon request, is commonly used in the stream mining community, but may not be realistic for many practical applications. Extreme latency, where labels are never made available to the learner, is closer to an unsupervised learning task. Intermediate latency assumes a delay $0 < L < \infty$ in the availability of the labels from the oracle.

Finally, the training efficiency of the online active learning algorithms should also be taken into consideration. There are two main training approaches in active learning; incremental training and complete re-training. Incremental training involves updating model parameters with a small batch of new data, without starting the training process from scratch (Polikar et al., 2001; Wu et al., 2019; Shilton et al., 2005; Istrate et al., 2018). This approach allows the model to learn from new data while preserving its existing knowledge. This can be achieved through fine-tuning the model parameters with the new data, or by using techniques such as elastic weight consolidation, which prevent previous knowledge from being erased. Complete re-training, on the other hand, involves training a new model from scratch using the entire labeled data collected so far. This approach discards the previous knowledge of the model and starts anew, which may result in the loss of knowledge learned from previous data. Complete re-training is typically used when the amount of new data is substantial, the previous model is no longer relevant, or when the model architecture needs to be altered. It is important to note that the choice of training approach in online active learning algorithms can have a significant impact on the overall performance and effectiveness of the model.

2.3 Connection between active learning and semi-supervised learning

Semi-supervised learning is a field of research that is closely related to active learning, as both methods are developed to deal with limited labeled data. While active learning aims to minimize the amount of labeled data required to train a model, semi-supervised learning is a technique that trains a model using a combination of labeled and unlabeled data. Active learning can be considered a special case of semi-supervised learning, as it allows the model to actively select which data points it wants to be labeled, rather than relying on a fixed set of labeled data. In the context of online learning, Kulkarni et al. (2016) conducted a study that provided an overview of semi-supervised learning techniques for classifying data streams. These techniques do not address the primary question of active learning, which is *when to query*, but they are useful in exploiting the information contained in the unlabeled data points and in addressing issues related to model update and retraining in limited labeled data environments. It is also worth noting that semi-supervised learning can be used in combination with active learning to improve the data selection strategy. By leveraging the strengths of both methods, it is possible to achieve better performance and more efficient learning compared to using either method alone.

Semi-supervised learning approaches can be distinguished into three categories, unsupervised preprocessing, wrapper methods, and graph-based methods. Unsupervised preprocessing refers to the use of unsupervised learning techniques, such as dimensionality reduction (Cacciarelli & Kulahci, 2023), clustering, or feature extraction, to preprocess the entire dataset, labeled and unlabeled, before it is fed to the supervised model (Frumosu & Kulahci, 2018). The goal is to transform the data into a more useful representation that can be learned more easily by a supervised model and can support the sampling of more informative data points. This strategy can also help reduce the dimensionality of the learning problem, thus improving the model parameter estimation when only a few queries can be made. Related to the online active learning problem, Rožanec et al. (2022) used a pre-trained network to extract salient features from unlabeled images before starting the sampling routine. Similarly, Cacciarelli et al. (2022a) used an autoencoder trained on all the available unlabeled data points to improve the performance of online active learning for linear regression models.

Wrapper methods, on the other hand, use one or more supervised learners that are trained on labeled data and pseudo-labeled unlabeled data. There are two main variants of wrapper methods, self-training and co-training. Self-training uses a single supervised model that is trained on labeled data, and pseudo-labels are used for the data points with confident predictions. Co-training, on the other hand, extends self-training to multiple supervised models, where two or more models exchange the most confident predictions to obtain pseudo-labels. Pseudo-labels can be very beneficial in label-scarce environments, but one must be mindful of the confirmation bias issue, where the model might rely on incorrect self-created labels. This problem has been extensively analyzed by Baykal et al. (2022) in the active distillation scenario, which is a strategy where a smaller model, known as the student model, is trained to mimic the behavior of a larger pre-trained model, known as the teacher model (Hoang et al., 2021; Kwak et al., 2022). In this context, confirmatory bias refers to the student model tendency to reproduce the predictions of the teacher model, even when the teacher predictions are incorrect. This can happen when the student model is trained to mimic the teacher model output too closely, without considering the underlying errors. To mitigate this, active distillation techniques use sample selection methods that encourage the student model to learn from data points where the teacher model makes errors, rather than just reproducing the teacher model predictions. In the more general active learning framework, confirmation bias might also refer to the tendency of an active learning algorithm to select examples that confirm its current hypothesis, rather than selecting examples that would challenge or improve it.

Finally, graph-based methods construct a graph on all available data and fit a supervised model, where the loss comprises a supervised loss and a regularization term that penalizes the difference between the labels predicted for connected data points. In the online active learning scenario, the graph structure can be used to model the similarity between data points, and the active learning algorithm can select the examples to label based on their position on the graph, such as selecting examples that are in low-density regions or are distant from other labeled examples.

3 Online active learning approaches

In this review, we present a taxonomy of online active learning strategies into four categories:

1. *Stationary data stream classification approaches.* These methods are designed to tackle online classification tasks, where the model is updated on the fly using newly labeled examples selected from a stream of data that does not change significantly over time. These methods are particularly useful in scenarios where the data distribution is relatively stable, such as quality control in industrial processes, where stationarity is often ensured by control actions taken at regular intervals and continuous maintenance of the components of the system (Bisgaard & Kulahci, 2011). Another example is represented by human activity recognition using wearable devices, where data is collected over time from wearable devices such as fitness trackers to identify patterns of activity like walking, running, or sleeping. This scenario would fall into this category because the data stream is relatively stable, and the model can be updated in real-time as new labeled examples become available (Miu et al., 2015).
2. *Drifting data stream classification approaches.* These online active learning strategies are specifically designed to handle classification tasks in dynamic environments where the

data distribution constantly changes. These approaches are designed to adapt to changes in the data distribution in order to maintain high classification accuracy. Some real-world applications might be fraud detection or intrusion detection. In financial fraud detection, fraudsters often change their methods to evade detection, so a classification model used for fraud detection must be able to adapt to new patterns of fraud as they emerge or to new customer habits (Zhang et al., 2022). In real-time intrusion detection, computer networks detection systems must be able to detect new forms of cyberattacks as they appear, so the classification models used must be able to adapt to changes in the data distribution over time (Nixon et al., 2021). This scenario would fall into this category because the data stream is constantly changing, and the model must be able to adapt to changes in the data distribution over time to maintain high accuracy.

3. *Evolving fuzzy system approaches.* These approaches are based on a type of fuzzy system that can adapt and change over time, in response to new data or changes in the environment (Gu et al., 2023). In traditional fuzzy systems, the rules and membership functions that define the system are fixed and do not change over time. Evolving fuzzy systems, on the other hand, are able to adapt their rules and membership functions based on new data or changes in the environment. This is particularly useful in applications where the data or the environment is non-stationary and evolves over time, such as in control systems for autonomous vehicles, where we must be able to adapt to changes in the environment, such as traffic patterns, road conditions, and weather (Naranjo et al., 2007; Wang et al., 2015).
4. *Experimental design and bandit approaches.* These methods, mostly related to regression models, actively select the most informative data points to improve model predictions. This category includes online active linear regression and sequential decision-making strategies like bandit algorithms or reinforcement learning. These methods adaptively select the most promising options in a given situation. An example is given by online advertising, where a model is used to select the most promising advertisements to display to users based on their browsing history and other factors (Avadhanula et al., 2021). This scenario would fall into this category because the model must adaptively select the most promising options in real-time based on the information available at that time. Also, in clinical trials, a model is used to select the most promising patients to enroll in a clinical trial based on their medical history and other personal information. Finally, in drug development studies (Réda et al., 2020), online active learning can be used to select the most promising compounds for further testing and development, based on their potential efficacy and safety.

This categorization provides a comprehensive overview of the different types of online active learning strategies and how they can be applied in various scenarios. While the simplest active learning strategy, random sampling, is available and involves selecting data points randomly from the stream for annotation, we will primarily focus on more specialized strategies designed to address scenarios where informed decisions are crucial due to resource constraints or where the data distribution is non-stationary.

Figure 5 depicts a general framework illustrating the essential components shared by the various categories of online active learning algorithms. The accompanying callouts highlight key options utilized by these methods. The following sections will provide an in-depth analysis of these strategies. For a more detailed flowchart regarding the drift detection and adaptation process, please refer to Lu et al. (2018), Lima et al. (2022).

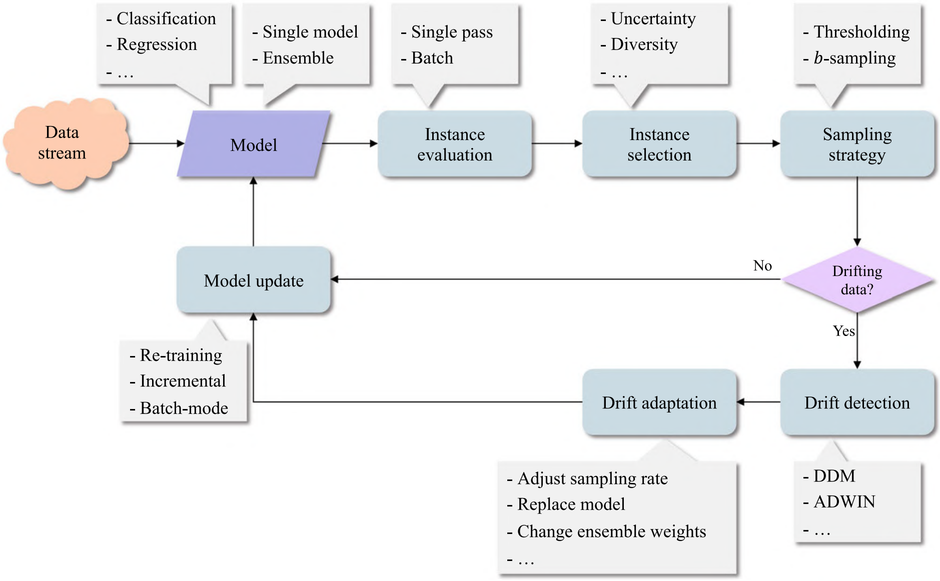


Fig. 5 Online active learning: general framework

3.1 Stationary data stream classification approaches

In online active learning, a commonly employed strategy is to request labels for data points that are considered to be informative enough based on a pre-determined threshold. This threshold can be established through a variety of techniques, depending on the instance selection criterion used to evaluate the informativeness of the unlabeled observations. Another method, sometimes referred to as b-sampling, is to calculate the probability that a data point will be queried by adjusting the parameter of a Bernoulli random variable, as proposed by Cesa-Bianchi et al. in one of the pioneering studies on online active learning (Cesa-Bianchi et al., 2004, 2006). They used a linear predictor characterized by the weight vector $\mathbf{w} \in \mathbb{R}^d$ and, at each time step t, after observing the current data point $\mathbf{x}_t$, the binary output $y \in \{-1, +1\}$ is predicted using

$$\widehat{y}_t = \mathrm{SGN}\left(\mathbf{w}_{t-1}^\top \mathbf{x}_t\right) \tag{1}$$

where $\mathbf{w}_{t-1}$ is the weight vector estimated with the previously seen labeled examples $(\mathbf{x}_1, y_1)$, $\ldots, (\mathbf{x}_{t-1}, y_{t-1})$. The value $\mathbf{w}_{t-1}^\top \mathbf{x}_t$ is the margin, $\widehat{p}_t$, of $\mathbf{w}_{t-1}$ on the instance $\mathbf{x}_t$. If the learner queries the label y_t, a new weight vector is estimated using the newly added labeled example $(\mathbf{x}_t, y_t)$ with the regular perceptron update rule (Rosenblatt, 1958) as in

$$\mathbf{w}_t = \mathbf{w}_{t-1} + M_t y_t \mathbf{x}_t \tag{2}$$

where M_t represents the indicator function of the event $\widehat{y}_t \neq y_t$. If the label is not requested, the model remains unchanged, and we have $\mathbf{w}_t = \mathbf{w}_{t-1}$. At each time step t, the learner decides whether to query the label of a data point $\mathbf{x}_t$ by drawing a Bernoulli random variable $Z_t \in \{0, 1\}$, whose parameter is given by

$$P_t = \frac{b}{b + |\widehat{p}_t|} \tag{3}$$

where $b > 0$ is a positive smoothing constant that can be tuned to adjust the labeling rate. In general, as $\widehat{p}_t$ approaches 0, the sampling probability P_t converges to 1, suggesting that the labels are requested for highly uncertain observations. The sampling scheme introduced by Cesa-Bianchi et al. (2004) is referred to as selective sampling perceptron, and it is reported in Algorithm 1.

Algorithm 1 Selective sampling perceptron

```
Require: a data stream $\mathbf{S}$, an initial model $\mathbf{w}_0 = (0, \dots, 0)^\top$, a time horizon $T$, a sampling budget $B$, a
  parameter $b$.
  $t \leftarrow 1$                                                    ▷ Timestamp
  $c \leftarrow 0$                                                    ▷ Labeling cost
  while $c \leq B, t \leq T$ do
    Observe an incoming data point $\mathbf{x}_t \in \mathbf{S}$ and set $\widehat{p}_t = \mathbf{w}_{t-1}^\top \mathbf{x}_t$
    Predict the label $\widehat{y}_t = \mathrm{SGN}(\widehat{p}_t)$
    Draw a Bernoulli random variable $Z_t$ of parameter $P_t = b/(b + |\widehat{p}_t|)$
    if $Z_t = 1$ then                                                 ▷ Sampling decision
      Ask for the true label $y_t$ and update the model
      $c \leftarrow c + 1$                                            ▷ Pay for the label
    else
      Discard $\mathbf{x}_t$
    end if
    $t \leftarrow t + 1$
  end while
```

A similar approach to the one proposed by Cesa-Bianchi et al. (2004) was investigated by Dasgupta et al. (2005), who presented one of the first thresholding techniques for online active learning. They suggested setting a threshold on the margin, with the idea of sampling data points $\mathbf{x}_t$ with a value of $|\widehat{p}_t|$ lower than a given threshold Γ. The threshold is initially set at a high value and iteratively divided by two until enough misclassifications occur among the queried points. The linear classifier is updated using the reflection concept [60] to give more focus to recent data points. Sculley (2007) built on the works of Cesa-Bianchi and Dasgupta to analyze the online active learning scenarios for real-time spam filtering. The author compares two models, a perceptron and a support vector machine (SVM), and tries three different instance selection criteria, the fixed thresholding approach by Dasgupta et al. (2005), the Bernoulli-based approach by Cesa-Bianchi et al. (2004), and a newly developed logistic margin sampling. The perceptron is updated as per Dasgupta et al. (2005), while the SVM is retrained on all available labeled observations each time a new data point is added. According to the logistic margin sampling strategy, the sampling decision is taken by drawing a Bernoulli random variable $Z_t \in \{0, 1\}$ with a parameter given by

$$P_t = e^{-\gamma|\widehat{p}_t|} \tag{4}$$

As in the traditional b-sampling approach introduced by Cesa-Bianchi et al. (2004), this sampling strategy depends on the uncertainty, meant as the distance from the prediction hyperplane. The main difference between the two strategies is the shape of the resulting sampling distribution, which can be observed in Fig. 6.

The selective sampling perceptron approach has also been investigated by Lu et al. (2016), who proposed an online passive-aggressive active learning variant of the algorithm. Similarly to the b-sampling approach, at each time step t, a Bernoulli random variable $Z_t \in \{0, 1\}$ is drawn to decide whether to query the label of the current data point $\mathbf{x}_t$ or not. In this case, the parameter of Z_t is given by

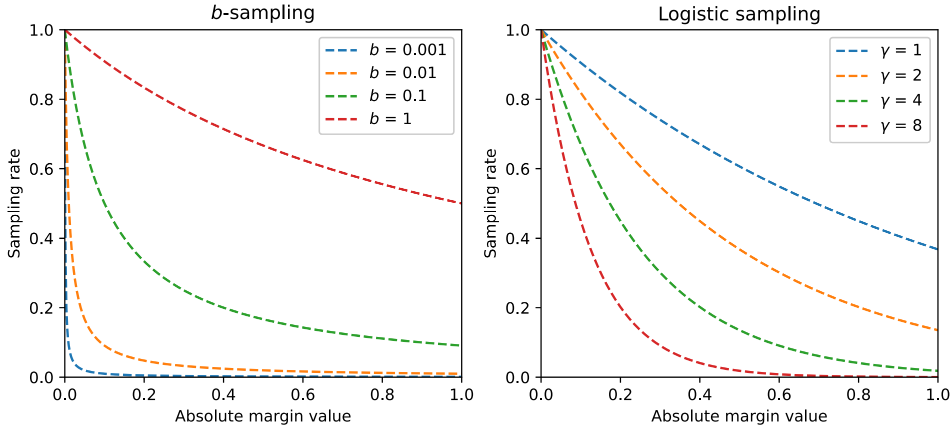


Fig. 6 Shape of the sampling distributions for b-sampling (**a**) and logistic sampling (**b**), for different values of b and γ

$$P_t = \frac{\delta}{\delta + |\widehat{p_t}|} \tag{5}$$

where $\delta \geq 1$ is a smoothing parameter. Besides not allowing the smoothing parameter to assume a value lower than 1, the sampling distribution is the same as the one governed by the parameter in Eq. 3. The main difference lies in the passive-aggressive approach used for updating the weight vector. Indeed, while the traditional perceptron update, shown in Eq. 2, only uses misclassified examples to update the model, the passive-aggressive approach updates the weight vector $\mathbf{w} \in \mathbb{R}^d$ whenever the current loss $\ell_t(\mathbf{w}_{t-1}; (\mathbf{x}_t, y_t))$ is nonzero (Crammer et al., 2006). The new parameter $\mathbf{w}_t$ is found using

$$\mathbf{w}_t = \mathbf{w}_{t-1} + \tau_t y_t \mathbf{x}_t \tag{6}$$

where τ_t represents the step size, and can be computed according to three different policies

$$\tau_t = \begin{cases} \ell_t(\mathbf{w}_{t-1}; (\mathbf{x}_t, y_t)) \,/\, \|\mathbf{x}_t\|^2 \\ \min\left(\kappa, \ell_t(\mathbf{w}_{t-1}; (\boldsymbol{x}_t, y_t)) \,/\, \|\boldsymbol{x}_t\|^2\right) \\ \ell_t(\mathbf{w}_{t-1}; (\mathbf{x}_t, y_t)) \,/\, \left(\|\mathbf{x}_t\|^2 + 1/2\kappa\right) \end{cases} \tag{7}$$

where κ is a penalty cost parameter. Passive-aggressive algorithms are known for their aggressive approach in updating the model, which is motivated by the fact that traditional perceptron updates might waste data points that have been correctly classified but with low prediction confidence.

A related issue to the update of the weight vector $\mathbf{w}_t$ was emphasized by Bordes et al. (2005), who noted that always picking the most misclassified example is a reasonable sampling strategy only when the training examples are highly confident. When dealing with noisy labels, this strategy could lead to the selection of misclassified examples or examples lying on the wrong side of the optimal decision boundary. To address this, they suggested a more conservative approach that selects examples for updating $\mathbf{w}_t$ based on a minimax gradient strategy.

In addition to confidence in the labels of the training examples, confidence in the model itself must be considered when the sampling strategy is based solely on model predictions. Hao et al. (2018b) pointed out that a margin-based sampling strategy may be suboptimal when the classifier is not precise, especially in the early rounds of active learning when the model performance may be poor due to limited training feedback, leading to misleading sampling decisions. This issue is also referred to as cold-start active learning (Houlsby et al., 2014; Yuan et al., 2020; Jin et al., 2022). To address this, Hao et al. (2018b) propose considering second-order information in addition to margin value when deciding whether or not to query the label of a data point $\mathbf{x}_t$. In general, first-order online active learning strategies only consider the margin value, while second-order methods also take into account the confidence associated with it. To do this, they assume that the weight vector of the classifier $\mathbf{w} \in \mathbb{R}^d$ is distributed as

$$\mathbf{w} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma) \tag{8}$$

where the values μ_i and $\Sigma_{i,i}$ encode the model knowledge and confidence in the weight vector for the ith feature w_i. The covariance between the ith and jth features is captured by the term $\Sigma_{i,j}$. The smaller the variance associated with the coefficient w_i, the more confident the learner is about its mean value μ_i. The objective of the proposed method is to take into account the confidence of the model when updating the model and making the sampling decision. With regards to the model update, when the true label y_t of $\mathbf{x}_t$ is queried, the Gaussian distribution in Eq. 8 is updated by minimizing an objective function based on the Kullback–Leibler divergence (Joyce, 2011) to ensure the updated model is not too different from the previous one. The sampling decision uses an additional parameter to the margin $\widehat{p}_t$, which is defined as

$$c_t = \frac{-\eta}{2\left(\frac{1}{\nu_t} + \frac{1}{\gamma}\right)} \tag{9}$$

where $\eta, \gamma > 0$ are two fixed hyper-parameters and ν_t represents the variance of the margin related to the data point $\mathbf{x}_t$. The intuition is that, when the variance ν_t is high, the model has not been sufficiently trained on instances similar to $\mathbf{x}_t$, and querying its label would lead to a model improvement. Then, a soft margin-based approach is employed by computing

$$\rho_t = |\widehat{p}_t| + c_t \tag{10}$$

If $\rho_t \leq 0$, the label is always queried as the model is extremely uncertain about the margin. Instead, when $\rho_t > 0$, the model is more confident, and the labeling decision is taken by drawing a Bernoulli random variable of parameter

$$P_t = \frac{\delta}{\delta + \rho_t} \tag{11}$$

where $\delta > 0$ is a smoothing parameter. Finally, Hao et al. (2018b) also introduced a cost-sensitive variant of the loss function, for dealing with class-imbalanced applications. For a comprehensive discussion on imbalanced data stream analysis, please see Aguiar et al. (2023).

The cold-start issue related to the application of active learning to imbalanced datasets has also been highlighted by Qin et al. (2021), who used extreme learning machines (Huang et al., 2006) and extended the active learning framework initially proposed by Yu et al. (2015) to the multiclass classification scenario. They highlighted the challenge of the lack of instances for certain classes in imbalanced datasets, which can seriously impact the predictive ability of the model for those classes. To address this issue, they propose a sampling strategy that considers both diversity and uncertainty. The diversity is calculated by computing pairwise

Manhattan distance between the unlabeled observations. The uncertainty of a data point $\mathbf{x}_t$ is computed by taking the difference between the largest two posterior probabilities as in

$$\operatorname{margin}\left(\mathbf{x}_t\right)=p\left(y=c_b \mid \mathbf{x}_t\right)-p\left(y=c_{sb} \mid \mathbf{x}_t\right) \tag{12}$$

where c_b and c_{sb} are the classes with the highest posterior probabilities. This approach is also referred to as best-versus-second-best margin and, as highlighted by Joshi et al. (2009), is a good indicator of uncertainty when a large number of classes are present in the data. It should be noted that the sampling strategy introduced by Qin et al. (2021) is not suited for single-pass active learning as it requires computing similarity and uncertainty measures for all the unlabeled observations in the current batch.

Another approach to deal with class imbalance in active learning was proposed by Ferdowsi et al. (2013), who used linear SVMs and a sampling strategy that switches between multiple instance selection criteria online. This approach, however, is limited to a pool-based setting and requires predicting an unsupervised evaluation score for all available unlabeled instances. The impact of the last queried observations on the scores associated with the unlabeled data points is evaluated, and a greedy approach is used to decide which instance selection criterion to trust. SVMs have also been used by Ghassemi et al. (2016), who proposed a differentially private approach to online active learning. The privacy concerns are tackled both during the instance selection and the training phase, by randomizing the strategy introduced by Tong and Koller (2002). The informativeness of a data point $\mathbf{x}_t$ is measured by its closeness to the current hyperplane $\mathbf{w}_t$ as in

$$c(t)=\exp\left(-d\left(\mathbf{x}_t, \mathbf{w}_t\right)\right) \in [0,1] \tag{13}$$

where the distance function $d\left(\mathbf{x}_t, \mathbf{w}_t\right)$ is defined as

$$d\left(\mathbf{x}_t, \mathbf{w}_t\right) \triangleq \frac{\left|\left\langle \mathbf{w}_t, \mathbf{x}_t\right\rangle\right|}{\left\|\mathbf{w}_t\right\|} \tag{14}$$

In the traditional framework, the label y_t is queried if we have $c(t) > \Gamma$, where Γ is a predefined threshold. It should be noted that $c(t) > \Gamma$ is equivalent to $d\left(\mathbf{x}_t, \mathbf{w}_t\right) \leq \log 1/\Gamma$, which means that the observation $\mathbf{x}_t$ is in a sampling region of width $2\log 1/\Gamma$ around $\mathbf{w}_t$. However, to avoid a deterministic decision process on the labeling and ensure privacy, some randomness needs to be introduced. This can be done in two ways. First, the labeling decision can be modeled as a Bernoulli random variable of parameter p if $c(t) < \Gamma$ or $(1-p)$ if $c(t) \geq \Gamma$, where $p < 1/2$. Another approach is based on the exponential mechanism introduced by McSherry and Talwar (2007). According to this strategy, the algorithm sets a constant probability of labeling data points within a sampling region defined by α, and a decaying probability for points outside of it. The selection strategy is represented by a Bernoulli of parameter

$$q(t)=\begin{cases} e^{-\alpha\epsilon/\Delta} & d\left(\mathbf{x}_t, \mathbf{w}_t\right) \leq \alpha \\ e^{-d\left(\mathbf{x}_t, \mathbf{w}_t\right)\epsilon/\Delta} & d\left(\mathbf{x}_t, \mathbf{w}_t\right) > \alpha \end{cases} \tag{15}$$

where $\epsilon > 0$ and $\Delta = (1-\alpha/M)M$. The authors assumed all data points belonging to the stream to be bounded in norm by M, $\|\mathbf{x}_t\| \leq M$ for $t=1,\ldots,T$. To tackle the privacy concerns while training, the authors propose two mini-batch strategies, to avoid the problem of slow convergence that may result from introducing noise according to the private stochastic gradient descent scheme (Bassily et al., 2014; Song et al., 2013; Duchi et al., 2013).

Two different approaches have been proposed by Ma et al. (2016) and Shah and Manwani (2020). Ma et al. (2016) proposed a query-while-learning strategy for decision tree classifiers. They used entropy intervals extracted from the evidential likelihood to determine the dominant attributes, which are ordered based on the information gain ratio. When a new data point $\mathbf{x}_t$ is observed, its label is queried only if there does not exist a dominant attribute. This will help to identify one and narrow the entropy interval. However, it should be noted that the authors consider a query while learning framework that only partially relates to to online active learning. Shah and Manwani (2020) investigated the online active learning problem for reject option classifiers. Given the high cost that is sometimes associated with a misclassification error, these models are given the option of not predicting anything, for example when dealing with a highly ambiguous instance. A typical application of reject option classifiers is in the medical field, when making a diagnosis with ambiguous symptoms might be particularly difficult. In this case, it could be more beneficial not to provide a prediction but suggest further tests instead. They proposed an approach based on a non-convex double ramp loss function ℓ_{dr} (Manwani et al., 2013), where the label of the current example $\mathbf{x}_t$ is queried only if it falls in the linear region of the loss given by $|f_t(\mathbf{x}_t)| \in [\rho_t - 1, \rho_t + 1]$, which is the region where the parameter would be updated. Here, ρ refers to the bandwidth parameter of the reject option classifier that determines the rejection region.

Fujii and Kashima (2016) investigated the problem of Bayesian online active learning. They provided a general framework based on policy-adaptive submodularity to handle data streams in an online setting. The authors distinguish between the stream setting, where the labeling decision can be made within a given timeframe, and the secretary setting, introduced in Sect. 2, where the labeling decision must be made immediately. The proposed framework can be applied in a variety of active learning scenarios, such as active classification, active clustering, and active feature selection. The framework is based on the concept of adaptive submodular maximization, which extends the idea of submodular maximization. A set function is considered to be submodular if it satisfies the property of diminishing returns, meaning that adding an element to a smaller set has a greater impact on the function value than adding the same element to a larger set. Adaptive submodular maximization allows the model to adapt to the changing distribution of data over time, by adjusting the set function to reflect the current state of knowledge. This leads to more efficient use of available data and improved performance.

So far, we discussed several single model approaches to active learning, which have shown promising results in various applications. However, it is important to note that single models have their limitations and can sometimes struggle to capture complex patterns and diverse representations present in the data. To address these limitations, researchers have proposed the use of ensembles or committees as an alternative (Krawczyk et al., 2017). An ensemble or committee refers to a group of multiple models that collaborate to produce a more robust and accurate prediction by combining their individual predictions. The models in an ensemble or committee can be trained on different subsets of the data or with varying hyperparameters, and the final prediction is typically made through either voting or weighted averaging. Ensembles or committees can also be regarded as a collection of models that work together to make a prediction, either by exchanging information or learning from one another. Among this class of methods, a common sampling strategy is represented by disagreement-based active learning. A framework to perform disagreement-based active learning in online settings was recently introduced by Huang et al. (2022). They characterized the learner by a hypothesis space $\mathcal{H}$ of Vapnik–Chervonenkis (VC) dimension d, which is composed of all the classifiers currently under consideration, and a Tsybakov noise model (Mammen & Tsybakov, 1999; Tsybakov, 2004). Each classifier $h \in \mathcal{H}$ is a measurable function mapping the observation $\mathbf{x}_t$ to binary output $y_t = \{0, 1\}$. The disagreement among two classifiers is given by $d(h_1, h_2) = \mathbb{P}[h_1(\mathbf{x}) \neq h_2(\mathbf{x})]$ and the disagreement region is defined as

$$D\left(h_1, h_2\right) = \{\mathbf{x} \in \mathcal{X} : h_1(\mathbf{x}) \neq h_2(\mathbf{x})\} \tag{16}$$

The online active learning strategy is represented by the policy $\pi = (\{v_t\}, \{\lambda_t\})$, where $\{v_t\}_{t\geq 1}$ is the map of the queried data points, and $\{\lambda_t\}_{t\geq 1}$ is the sequence of prediction rules. Over the time horizon T, the performance of the policy π is evaluated using the label complexity and the regret. The label complexity measures the expected number of labels queried, with respect to the stochastic process induced by π, and it is given by

$$\mathbb{E}[Q(T)] = \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left[v_t = 1\right]\right] \tag{17}$$

The regret, on the other hand, represents the expected number of excess classification errors with respect to h^*, and it is obtained as

$$\mathbb{E}[R(T)] = \mathbb{E}\left[\sum_{t\leq T: v_t = 0} \mathbb{1}\left[\lambda_t \neq y_t\right] - \mathbb{1}\left[h^*\left(\mathbf{x}_t\right) \neq y_t\right]\right] \tag{18}$$

The objective of the algorithm is to minimize the label complexity with a constraint on the regret. At the first round, the initial version space is the entire hypothesis space $\mathcal{H}$, while the initial region of disagreement is the whole input space $\mathcal{X}$. Then, at time step t, the learner updates the version space $\mathcal{H}_t$ using the M collected labels, and computes a new region of disagreement as

$$\mathcal{D}\left(\mathcal{H}_t\right) = \{\mathbf{x} \in \mathcal{X} : \exists h_1, h_2 \in \mathcal{H}_t, h_1(\mathbf{x}) \neq h_2(\mathbf{x})\} \tag{19}$$

If $\mathbf{x}_t \in \mathcal{D}\left(\mathcal{H}_t\right)$, then the label of the current data point is queried, otherwise a prediction is produced using an arbitrary classifier in $\mathcal{H}_t$. At the end of the iteration t, the set $\mathcal{Z}_t$ of M collected labeled examples is used to estimate the empirical error $\epsilon_{\mathcal{Z}_t}(h)$ of the classifiers in $\mathcal{H}$ and identify the best currently available classifier. Then, the version space is updated by removing all the suboptimal hypotheses whose empirical error exceeds the one obtained with h_t^* by a threshold $\Delta_{\mathcal{Z}_t}\left(h, h_t^*\right)$. The threshold regulates the trade-off between reducing label complexity by narrowing the region of disagreement and increasing the regret by eliminating good classifiers.

The disagreement concept was also used by Desalvo et al. (2021), while proposing an approach to online active learning for binary classification tasks based on surrogate losses. The overall framework is similar to the disagreement-based one used by Huang et al. (2022), with the main difference being the use of weak-labels to optimize the sampling strategy. At each time step t, the learner observes the unlabeled data point $\mathbf{x}_t$ and either decides to request its label or assigns a pseudo-label $\widehat{y}_t$. Then, the pseudo labels $\widehat{y}_t$ and the true labels y_t processed so far are used together to obtain an estimate of the empirical risk $\epsilon_{\mathcal{S}_t}(h)$, where $\mathcal{S}_t$ is obtained by combining the collected labeled examples $\mathcal{Z}_t$ with the pseudo-labeled ones $\widehat{\mathcal{Z}}_t$. This represents an example of combining active learning and semi-supervised learning, as highlighted in Sect. 2.3.

Loy et al. (2012) presented a Bayesian framework that leverages the principle of committee consensus to balance exploration and exploitation in online active learning. The aim of exploration is to discover new, previously unknown classes, while exploitation focuses on refining the decision boundary for known classes. To address the issue of unknown classes, the framework uses a Pitman-Yor Processes (PYP) prior model (Pitman & Yor, 1997) with a Dirichlet process mixture model (DPMM). A DPMM is a non-parametric clustering and classification model that models the data generating process using a mixture of probability distributions. Each data point is assigned to a cluster, which is associated with a probability distribution over the classes. The number of clusters is modeled using a Dirichlet process, which is a distribution over distributions that allows for an infinite number of clusters but ensures that the number of actual clusters is always finite. At each time step t, the learner samples two random hypotheses h_1 and h_2 from the model. Then, it computes the posterior probability of the current class c corresponding to k, $p\left(c = k \mid \mathbf{x}_t\right)$, for each of the two hypotheses. Finally, $h_i\left(\mathbf{x}_t\right) = \arg\max p\left(c \mid \mathbf{x}_t\right)$ is calculated for $i = 1, 2$. The label of the current data point is queried in two cases: first, if $h_1\left(\mathbf{x}_t\right) \neq h_2\left(\mathbf{x}_t\right)$, meaning the two hypotheses disagree, and second, if $h_i\left(\mathbf{x}_t\right) = K + 1 \forall i$, where K is the number of currently known classes, meaning the current data point belongs to a new class.

The DPMM has also been used by Mohamad et al. (2020), who proposed a semi-supervised strategy for performing active learning in online human activity recognition with sensory data. To account for the possibility of dealing with different sensor network layouts, the authors proposed pre-training a conditional restricted Boltzmann machine (Taylor & Hinton, 2009; Taylor et al., 2006) and used it to extract generic features from the sensory input. The instance selection strategy follows a Bayesian approach, in trying to minimize the uncertainty about the model parameters. To assess the usefulness of labeling the data point $\mathbf{x}_t$, they measure the discrepancy between the model uncertainty computed from the data observed until the time step t and the expected risk associated with y_t. This gives a hint of how the current label would impact the current model uncertainty. A dynamically adaptive threshold Γ is finally used to the determine whether the current expected risk is greater than the current risk.

A different kind of committee has been considered by Hao et al. (2018a). They proposed a framework for minimizing the number of queries made by an online learner that is trying to make the best possible forecast, given the advice received from a pool of experts. To do so, they adapted the exponentially weighted average forecaster (EWAF) and the greedy forecaster (GF) to the online active learning scenario. A comprehensive analysis of forecasters to perform prediction with expert advice can be found in the book by Cesa-Bianchi and Lugosi (2006). In general, at each time step t, the learner or forecaster has access to the predictions for the data point $\mathbf{x}_t$ made by the N experts, $f_{i,t}\left(\mathbf{x}_t\right) : \mathbb{R}^d \rightarrow [0, 1]$ with $i = 1, \ldots, N$. Based on these predictions, it outputs its own prediction p_t for the outcome y_t. Then, if the label is revealed, the predictions made by the forecaster and the experts are scored using a nonnegative loss function ℓ. The objective of the learner is to minimize the cumulative regret over the time horizon T, which can be seen as the difference between its loss and the one obtained with each expert i as in

$$R_{i,T} = \sum_{t=1}^{T} \left(\ell\left(p_t, y_t\right) - \ell\left(f_{i,t}\left(\mathbf{x}_t\right), y_t\right)\right) = \widehat{L}_T - L_{i,T} \tag{20}$$

The most simple approach to obtain a prediction p_t from the learner is to compute a weighted average of the experts predictions as in

$$p_t = \frac{\sum_{i=1}^{N} \omega_{i,t} f_{i,t}\left(\mathbf{x}_t\right)}{\sum_{i=1}^{N} \omega_{i,t}} \tag{21}$$

where $\omega_{i,t} \geq 0$ is the weight assigned at time t to the ith expert. With the EWAF, the weight for the ith expert are obtained using

$$\omega_{i,t} = \frac{e^{\eta R_{i,t-1}}}{\sum_{i=1}^{N} e^{\eta R_{i,t-1}}} \tag{22}$$

where η is a positive decay factor and $R_{i,t-1}$ is the cumulative loss of expert i observed until step t. The exponential decay factor η determines the weight given to the past losses, with more recent losses having a higher weight and older losses having a lower weight. Instead, the GF works by minimizing, at each time step, the largest possible increase of the potential function for all the possible outcomes of y_t. The potential function is the function that assigns a potential value to each expert, which captures the quality of an expert advice based on its past performance. Hao et al. (2018a) extended the EWAF and GF by proposing the active EWAF (AEWAF) and active GF (AGF). The key idea is that, while the standard EWAF and GF assume the availability of the true label y_t after each prediction, in the online active learning framework the loss ℓ can only be measured a limited number of times. To factor this in, a binary variable $Z_t \in \{0, 1\}$ is introduced to decide whether or not at round t the label is requested. Consequently, the cumulative loss suffered by the ith expert on the instances queried by the active forecaster is given by

$$\widehat{L}_{i,T} = \sum_{t=1}^{T} \ell\left(f_{i,t}\left(\mathbf{x}_t\right), y_t\right) \cdot Z_t \tag{23}$$

The sampling strategy is based on the determination of a confidence condition on the difference between the prediction p_t of the fully supervised forecaster and the prediction $\widehat{p_t}$ made by the active forecaster. For the active forecaster we have that $\widehat{p_t} = \pi_{[0,1]}\left(\overline{p}_t\right)$, where $\overline{p}_t$ depends on the chosen model. The AEWAF is based upon the observation that if we have

$$\max_{1 \le i,j \le N} \left| f_{i,t}\left(\mathbf{x}_t\right) - f_{j,t}\left(\mathbf{x}_t\right)\right| \le \delta \tag{24}$$

then $|p_t - \widehat{p_t}| \le \delta$, where δ is a tolerance threshold. This means that the prediction of the forecaster is close to the one obtained in the fully supervised setting if the maximum difference of advice between any two experts is not too large. This assumption might not hold in the presence of noisy or bad experts and, to tackle this problem, the authors proposed a robust variant of the AEWAF. The AGF uses instead a confidence condition based on the fact that if

$$\max_{1 \le i,j \le N} \left| f_{i,t}\left(\mathbf{x}_t\right) - \overline{p}_t\right| \le \delta \tag{25}$$

then $|p_t - \widehat{p_t}| \le \delta$. The general scheme for performing online active learning with expert advice is reported in Algorithm 2.

A similar framework, in conjunction with multiple kernel learning (MKL), has been investigated by Chae and Hong (2021). They propose an active MKL (AMKL) algorithm based on random feature approximation. In general, online MKL based on random feature approximation is a method for online learning and prediction that combines multiple kernel functions to improve the performance of a learning algorithm (Jin et al., 2010; Hoi et al., 2013). In MKL, multiple kernel functions are used to capture different aspects of the data, and the optimal combination of kernels is learned from the data. The online version of MKL based on random feature approximation is designed to handle data that arrives sequentially, and the learning algorithm is updated after each new data point. In kernel-based learning, the target function $f(\mathbf{x})$ is assumed to belong to a reproducing Hilbert kernel space (RKHS). In the proposed AMKL the learner uses an ensemble of N kernel functions. At each time step t, two main steps are implemented. First, each kernel function $\hat{f}_{i,t}\left(\mathbf{x}_t\right)$, with $i = 1, \ldots, n$, is optimized independently of the other kernel functions. This is referred to as local step. Then, in the global step, the learner seeks the best function approximation $\widehat{f_t}\left(\mathbf{x}_t\right)$ by combining the N kernel functions as in

Algorithm 2 Online active learning with expert advice

```
Require: a data stream $\mathbf{S}$, a loss function $\ell$, a time horizon $T$, a set of $N$ experts, a tolerance threshold $\delta$, a
  sampling budget $B$.
  $t \leftarrow 1$                                                              ▷ Timestamp
  $c \leftarrow 0$                                                              ▷ Labeling cost
  while $c \leq B, t \leq T$ do
      Observe an incoming data point $\mathbf{x}_t \in \mathcal{S}$
      Receive advide by experts $\{f_{i,t}(\mathbf{x}_t) : i = 1, \ldots, N\}$
      Generate prediction $\overline{p}_t$ for the label $y_t$ and set $\widehat{p}_t = \pi_{[0,1]}(\overline{p}_t)$
      Draw a Bernoulli random variable $Z_t$ of parameter $P_t = b/(b + |\widehat{p}_t|)$
      if Equation 24 or 25 is satisfied then                                    ▷ Sampling decision
          Discard $\mathbf{x}_t$
      else
          Ask for the true label $y_t$
          $c \leftarrow c + 1$                                                  ▷ Pay for the label
      end if
      $t \leftarrow t + 1$
  end while
```

$$\widehat{f_t}(\mathbf{x}_t) = \sum^{N} \widehat{v}_{i,t} \hat{f}_{i,t}(\mathbf{x}_t) \tag{26}$$

where $\widehat{v}_{i,t}$ refers to the weight for the ith kernel function at round t. Similarly to the case with expert advice, the weights are determined by minimizing the regret over the time horizon T, which is defined as the difference between the loss of the learner and the one obtained with the best kernel function $f^*_{i,t}$. To do so, the weights are computed based on the past losses ℓ as

$$\widehat{\omega}_{i,t} = \exp\left(-\eta_g \sum_{\tau \in \mathcal{A}_{t-1}} \ell\left(\hat{f}_{i,\tau}(\mathbf{x}_\tau), y_\tau\right)\right) \tag{27}$$

where $\eta_g > 0$ is a tunable parameter and $\mathcal{A}_{t-1}$ is an index of time stamps t indicating the instances for which has label has been requested, thus permitting to measure the loss. Then, the weights $\widehat{v}_{i,t}$ are obtained from $\widehat{\omega}_{i,t}$ as follows

$$\widehat{v}_{i,t} = \frac{\widehat{\omega}_{i,t}}{\sum_{i=1}^{N} \widehat{\omega}_{i,t}} \tag{28}$$

Finally, the instance selection criterion is based on a confidence condition, denoted by with $\delta > 0$, on the similarity of the learned kernel function, which is a similar to the condition used by Hao et al. (2018a) in the formulation of the AEWAF

$$\max_{1 \leq j \leq N} \sum_{i=1}^{N} \widehat{v}_{i,t} \ell\left(\widehat{f}_{i,t}(\mathbf{x}_t), \widehat{f}_{j,t}(\mathbf{x}_t)\right) \leq \delta \tag{29}$$

3.2 Drifting data stream classification approaches

Active learning strategies belonging to this category aim to tackle online classification tasks in time-varying data streams affected by distribution shifts. We can classify distribution shifts into three main categories, depending on whether they concern the feature space $\mathbf{x}$ or the output dimension y. A shift that only affects the input distribution $p(\mathbf{x})$, and not the conditional distribution $p(y \mid \mathbf{x})$, is referred to as covariate shift (Zhou et al., 2021; Wu et al., 2021; Li et

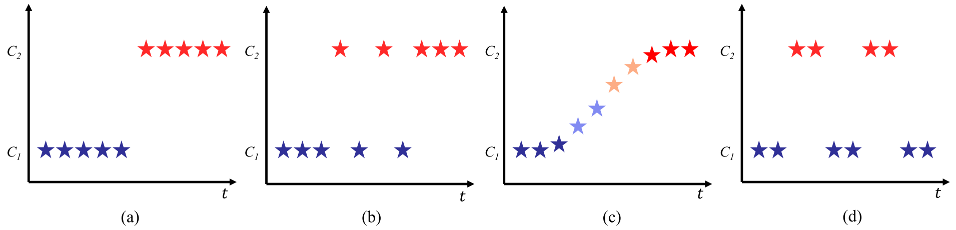


Fig. 7 Different types of drifts that can affect the data stream: abrupt drift (**a**), gradual drift (**b**), incremental drift (**c**), recurring concepts (**d**). C_1 and C_2 indicate the two concepts that might characterize the data distribution

al., 2021) or virtual drift (Baier et al., 2021). In these circumstances, for two different time steps, t_i and $t_{i+\Delta}$, we have that $p_{t_i}(\mathbf{x}) \neq p_{t_{i+\Delta}}(\mathbf{x})$ and $p_{t_i}(y \mid \mathbf{x}) = p_{t_{i+\Delta}}(y \mid \mathbf{x})$, meaning that the underlying model is not being altered by phenomena like class swaps or coefficient changes. Conversely, in the presence of a real concept drift (Baier et al., 2021; Suárez-Cetrulo et al., 2023), the conditional distribution changes, and we have $p_{t_i}(y \mid \mathbf{x}) \neq p_{t_{i+\Delta}}(y \mid \mathbf{x})$. In this scenario, the predictive performance of the fitted model dramatically deteriorates, and a model update or replacement becomes necessary. An example of this kind of distribution shift can be identified in the changes of the consumer behaviors over time, or following a major event as the COVID-19 pandemic (Zwanka & Buff, 2021). However, it should be noted that virtual drifts and real concept drifts often occur together (Tsymbal et al., 2008), leading to a situation where we have both $p_{t_i}(\mathbf{x}) \neq p_{t_{i+\Delta}}(\mathbf{x})$ and $p_{t_i}(y \mid \mathbf{x}) \neq p_{t_{i+\Delta}}(y \mid \mathbf{x})$ (Lu et al., 2018). Lastly, we can incur in a label distribution shift (Wu et al., 2021) when the shift only affects $p(y)$, leading to $p_{t_i}(y) \neq p_{t_{i+\Delta}}(y)$. This situation can be observed in many real-world scenarios where the target distribution changes over time. A typical example is the prediction of diseases like influenza, whose distribution can dramatically change depending on the season, or in the presence of sudden outbreaks.

Another key characteristic of distribution shifts is represented by the change rate, namely how fast the new concept or distribution is introduced into the data stream. To this extent, we can identify four kinds of drifts (Lu et al., 2018; Lima et al., 2022), which are illustrated in Fig. 7. A sudden or abrupt drift is a drift that can be immediately detected from two consecutive time steps, t_i and t_{i+1}. It refers to a sudden and clearly identifiable change in the data distribution. An example of this would be a sudden change in the weather, which would affect the behavior of customers at a retail store. The change is noticeable, and the model needs to be updated immediately. A gradual drift exhibits a transition phase, where a mixture or overlap between the two distributions p_{t_i} and $p_{t_{i+\Delta}}$ exists. In this case, the change is slower and more difficult to detect, making it challenging to update the model. An example would be a change in consumer behavior over time, which is hard to detect but can have a significant impact on a business. Another type of drift is the incremental drift, which has an extremely low transition rate, which makes it very difficult to detect changes between the data points observed in the transition period. This type of drift is often caused by changes in the data generating process that happen gradually over time, in small steps rather than all at once. An example would be changes in the types of products that are popular among customers, which happen gradually and are hard to detect. Finally, a data stream can also be affected by recurring concepts, which sequentially alternate over time. An example would be a retail store where the same types of products are popular at different times of the year, such as winter coats and summer dresses. The model needs to be able to detect and adapt to these recurring concepts in order to maintain good performance.

In online active learning for drifting data streams, some approaches address the presence of concept drifts by combining active learning strategies with drift detectors (Zhang et al., 2020a; Krawczyk et al., 2018). Drift detectors are algorithms that try to detect distribution shifts and identify when the context is changing. They can be divided into three macro-categories (Lu et al., 2018). The first group of methods is represented by the error-based drift detectors, which try to detect online changes in the error rate of a base classifier. Among these, one of the most commonly employed strategies is the drift detection method (DDM) proposed by Gama et al. (2004). Another popular approach is the adaptive window (ADWIN) strategy proposed by Bifet and Gavaldà (2007). The second class of drift detectors is called data distribution-based drift detection, and the third class is represented by multiple hypothesis testing strategies. While the first class contains the majority of the proposed approaches, it assumes that we are able to observe the labels of all the incoming data points to assess the error rate. Instead, the last two classes could be implemented even in an unsupervised manner. An exhaustive overview on unsupervised drift detection methods has been proposed by Gemaque et al. (2020). While the unsupervised nature of the data distribution-based and multiple hypothesis testing strategies make them ideal for the active learning scenario, it should be noted that real concept drifts can hardly be detected in a completely unsupervised fashion. Indeed, in a circumstance when the input distribution $p(\mathbf{x})$ remains unaltered while the underlying model relating the input variables $\mathbf{x}$ to the label y changes, it would not be possible to detect the change of concept without collecting labels. This is why Krawczyk et al. (2018) propose to apply an error-based drift detector to the few labels collected during the online active learning routine. To this extent, they use the ADWIN (Bifet & Gavaldà, 2007) method to detect drifts and decide when the current model needs to be updated or replaced. The proposed general framework for dealing with online active learning with drifting data streams is reported in Algorithm 3.

Moreover, the authors proposed the use of a time-variable threshold to balance the budget use over time. Their approach is based on the intuition that, when a new concept is introduced, more labeling effort will be required to quickly collect representative observations belonging to the new concept and replace the outdated model. This is obtained by adjusting a time-variable threshold to balance the budget use over time. Given a threshold Γ on the uncertainty of the classifier and a labeling rate adjustment $r \in [0, 1]$, the threshold is reduced to $\Gamma - r$ when ADWIN raises a warning and to $\Gamma - 2r$ when a real drift is detected. Thus, when allocating the labeling budget, the key requirement is that the labeling rate employed when a drift is detected should be strictly larger than the one used in static conditions. A similar thresholding idea has also been used by Castellani et al. (2022), who proposed an active learning strategy for non-stationary data streams in the presence of verification latency. They used a piece-wise constant budget function, where the labeling rate α is increased to α_{high} when a drift is detected and, after a while, reduced to α_{low}. Finally, the labeling rate is restored to its nominal value α. A visual representation of the labeling approach is shown in Fig. 8. The length of the time segments where the labeling rate is altered depends on the desired values for α_{high} and α_{low}, constraining the overall labeling rate to be equal to α.

The authors also tackled the verification latency issue by considering the spatial information of a queried point for which the label has not been made available yet by the oracle. In this way, it is possible to avoid oversampling from regions where many close points have a high utility, namely a low classification confidence. While assessing the utility of the incoming data points the authors use real and pseudo-labels by propagating the information contained in the already labeled observations, as suggested by Pham et al. (2022). The idea is to use the spatial information of the queried labels by estimating the still missing labels with a weighted majority vote of the label of its k-nearest neighbors labels, where the weight for

Algorithm 3 Online active learning with drifting data streams

```
Require: a data stream $\mathbf{S}$, a classifier $\Theta$, a drift detector $\Theta$, a sampling strategy $\Upsilon$, a labeling rate $\alpha$, a sampling budget $B$.
    $t \leftarrow 1$                                                         ▷ Timestamp
    $c \leftarrow 0$                                                         ▷ Labeling cost
    while $c \leq B$ and $t \leq |S|$ do
        Observe incoming data point $x_t \in \mathbf{S}$
        if $\Upsilon(x_t) =$ True then                                       ▷ Sampling decision
            Ask for the true label $y_t$
            $c \leftarrow c + 1$                                             ▷ Pay for the label
            Update classifier $\Psi$ with the labeled example $(\mathbf{x}_t, y_t)$
            Update drift detector $\Theta$ with the labeled example $(\mathbf{x}_t, y_t)$
            if drift warning = True then
                Start to train a new classifier $\Psi_{\text{new}}$
                Increase labeling rate $\alpha$
            else
                if drift detected = True then            ▷ A detection is always preceded by a warning
                    Replace $C$ with $C_{\text{new}}$
                    Further increase $\alpha$
                else
                    Return to initial labeling rate $\alpha$
                end if
            end if
            if $C_{\text{new}}$ exists then          ▷ Keeps being updated in the background until replacement
                Update classifier $C_{\text{new}}$ with the labeled example $(\mathbf{x}_t, y_t)$
            end if
        end if
        $t \leftarrow t + 1$
    end while
```

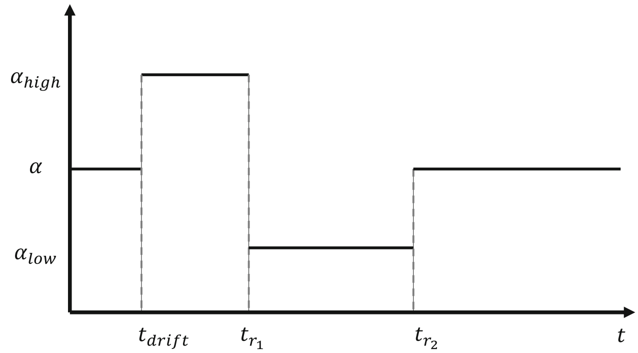


Fig. 8 Piece-wise constant budget function introduced by Castellani et al. (2022). The sampling rate α is increased to α_{high} when a drift is detected (t_{drift}), then reduced to α_{low} between t_{r_1} and t_{r_2}, before being restored to its nominal value

each nearest neighbor depends on the arrival time of the labels. The verification latency issue in online active learning with drifting data streams was also extensively analyzed by Pham et al. (2022). Consider the general case where at time t_i^x we draw an instance $\mathbf{x}_i$, and find it interesting enough to send it to the oracle, which will send back the label y_i only at time t_i^y, where $t_i^y > t_i^x$. Before the requested label arrives, we might encounter another instance similar to $\mathbf{x}_i$ and ask again for its label, since the learner could not update its utility function or threshold. Similarly, we might use outdated information when updating the policy in a future window. To tackle these issues, the authors propose a forgetting and simulating strategy to avoid using soon-to-be outdated observations and prevent redundant labeling. The instance selection is based upon the variable uncertainty strategy proposed by Zliobaite et al. (2014) and the balanced incremental quantile filter by Kottke et al. (2015). If we denote the current sliding window at time t_n^x as $\mathcal{W}_n = \left[t_n^x - \Delta, t_n^x\right)$ and use windows of fixed size Δ, we know that the sliding window that would be used for training when the label y_n related to $\mathbf{x}_n$ arrives will be given by $\mathcal{D}_n = \left[t_n^y - \Delta, t_n^y\right)$. The forgetting step is then implemented by discarding outdated labeled examples that are included in $\mathcal{W}_n$ but will not be included in $\mathcal{D}_n$. If a_i is a Boolean variable indicating whether the ith observation has been labeled, the set of instances selected to be forgotten is given by

$$O_n = \left[(\mathbf{x}_i, y_i)\, \forall i < n : a_i = 1 \wedge t_i^x, t_i^y \in \mathcal{W}_n \backslash \mathcal{D}_n\right]. \tag{30}$$

Similarly, there is a second set of observations, with time stamps $\mathcal{D}_n^+ = \mathcal{D}_n \backslash \mathcal{W}_n = \left[t_n^x, t_n^y\right)$, where there might be instances that have been queried but whose label is not currently available. To avoid losing such information and redundantly asking for the label of similar instances, the algorithm simulates incoming labels with a bagging approach by averaging across multiple utility estimations. They also consider an alternative simulation approach based on fuzzy labeling.

Similarly to Krawczyk et al. (2018), the ADWIN drift detector has also been used by Zhang et al. (2020a) while proposing a method for dealing with online active learning in environments characterized by concept drifts and class imbalance. The instance selection criterion is based on the predictive uncertainty, which they estimate using the best-versus-second-best margin value (Eq. 12), as they tackle a multi-class classification problem. An initial pool of n observations is passively collected from the stream to initialize the active learning strategy. Then, a threshold Γ_i is estimated for each class as in

$$\Gamma_i = \begin{cases} \frac{nm}{n_i L} & \frac{n}{n_i L} \geq 1 \\ m & \frac{n}{n_i L} < 1 \end{cases} \tag{31}$$

where $i = 1, \ldots, L$ is the number of classes and m is a pre-defined constant used to control the size of the threshold. The model is represented by an ensemble of N classifiers and, when ADWIN detects a concept drift, the classifier with the higher error is replaced with a newly trained one. Finally, the class imbalance issue is also taken into account in two ways, during the training of the ensemble with the use of class-specific weights, and during the active learning routine, by dynamically adjusting the threshold to select more observations belonging to the minority class.

Recently, Cheng et al. (2023) presented another approach to combine online active learning with drift detection. Their method involves segmenting the data stream $\mathbf{S}$ into fixed-length chunks and then detecting drifts by comparing the distributions of adjacent chunks. After a drift is detected, a multi-objective optimization problem is formulated in order to identify the most relevant and diverse data points within the current batch. For a data point $\mathbf{x}_t$, relevance is defined as its contribution to the new concept, and diversity as the Pearson correlation coefficient with other instances in the same region. Instead, Martins et al. (2023) proposed to sample the most uncertain data points from each chunk, using a meta-learning framework to fine-tune the threshold used for each window. This allows to reduce the need for labels while maintaining a steady adaptation to the new concepts.

Another window-based approach to perform active learning from data streams has been proposed by Zhu et al. (2007). The authors developed an ensemble E by partitioning the data stream $\mathbf{S}$ into chunks and then training each of the k models composing the ensemble E on

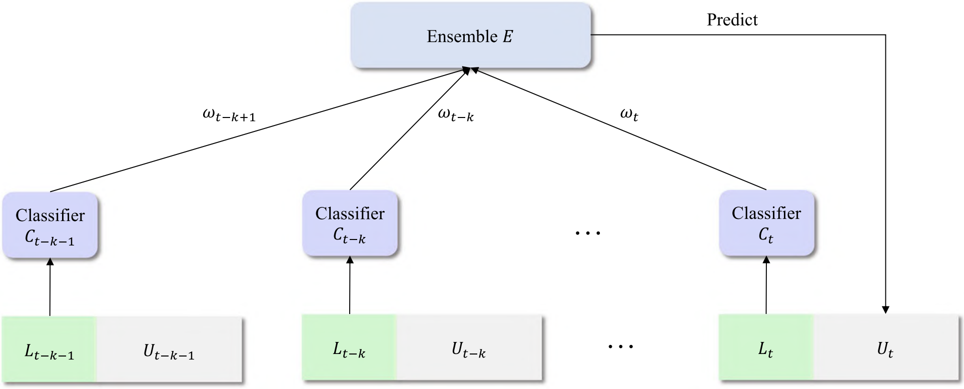


Fig. 9 Ensemble-based active learning framework for data streams proposed by Zhu et al. (2007)

a different chunk of data. In this way, even if the previous observations become unavailable, the models can be used when taking the sampling decision in order to take into account a global uncertainty measure, which is a more robust approach than treating each chunk as a static dataset. At time step t, the learner receives a data chunk $\mathbf{S}_t$, which is used to build the current classifier C_t. At this point, the ensemble is composed by C_t, together with the most recent $k-1$ classifiers, $C_{t-k+1}, \ldots, C_{t-1}$, trained on the labeled examples sampled from the previously observed data chunks, $L_{t-k+1}, \ldots, L_{t-1}$. At each iteration, the objective is to predict the remaining unlabeled data points from the current chunk, U_t. The ensemble-based active learning framework is depicted in Fig. 9. The instances selected to be queried are the ones with the largest ensemble variance, and the predictions are obtained by combining the predictions of the single classifiers using the weights $\omega_{t-k+1}, \ldots, \omega_{t-1}$. Finally, a weight updating rule is used to adapt to dynamic data streams.

Shan et al. (2019) and Zhang et al. (2018) developed online active learning strategies by building upon the pairwise classifiers strategy introduced by Xu et al. (2016). The pairwise strategy makes use of two models, a stable classifier C_s and a dynamic classifier C_d, and divides the data stream into batches as in Zhu et al. (2007). The prediction for an incoming data point $\mathbf{x}_t$ is obtained with a weighted average of the predictions obtained from the two classifiers as in

$$f_E(\mathbf{x}_t) = \omega_s f_{C_s}(\mathbf{x}_t) + \omega_d f_{C_d}(\mathbf{x}_t) \tag{32}$$

where ω_s and ω_d are the weights associated with the stable and the dynamic classifier, respectively. At time t, the stable classifier C_s is trained on the labeled portions of all the batches processed so far, $L_1, \ldots, L_{t-1}$. Conversely, C_d is trained exclusively on L_{t-1}. The key idea is that whenever the reactive classifier starts to outperform the stable classifier, the stable classifier is replaced by the reactive one, which is eventually reset. This replacement allows the learner to adapt to the drift and focus on the most recent instances, forgetting the seemingly obsolete data points. The main drawback of this approach is that it cannot effectively address gradual drifts as the replacement with the classifier trained on the most recent observations makes the learner forget about observations away from the current window. Hence, similarly to the approach of Zhu et al. (2007), Shan et al. (2019) proposed an extension of this approach, based on an ensemble of classifiers in trying to contemporarily address gradual drifts and abrupt drifts. In their strategy, the stable classifier learns from all the labeled instances and the reactive classifier is replaced by an ensemble of dynamic classifiers, trained on multilevel sliding windows to capture changes in the data stream at different time intervals. The instance selection approach combines random sampling and uncertainty sampling, where the latter is based on the margin value of the predictions obtained by the ensemble. It should be noted that the prediction f_E for the data point $\mathbf{x}_t$ is obtained as a weighted combination of the predictions obtained with the stable and dynamic classifiers as in

$$f_E(\mathbf{x}_t) = \omega_s f_{C_s}(\mathbf{x}_t) + \sum_{d=1}^{D} \omega_d f_{C_d}(\mathbf{x}_t) \tag{33}$$

The stable classifier has a constant weight $\omega_s = 0.5$ and plays a crucial role in trying to learn the overall trend and direction of concept drift. Conversely, the dynamic classifiers have gradually decaying weights, according to a damped sliding winding approach where each weight is initialized at $\frac{1}{D}$ and then reduced according to its creation time

$$\omega_d = \begin{cases} \omega_d\left(1 - \frac{1}{D}\right) & d = 1, \ldots, D-1 \\ \frac{1}{D} & d = D \end{cases} \tag{34}$$

The most recent classifiers are useful in detecting sudden concept drifts and have highest weights while the old dynamic classifiers have lower weights and can help to identify gradual drifts. The same pairwise strategy based on an ensemble composed by a stable classifier and D dynamic classifiers was used by Zhang et al. (2022). They modified the original strategy by introducing a reinforcement mechanism to adjust the weights ω_d according to the prediction performance and the class imbalance issue. The weights adjustment strategy is described by Algorithm 4. It should be noted that this procedure is only implemented after the true label y_t has been revealed by the oracle. The damped class imbalance ratio (DCIR) value is obtained by taking into account the number of observations for each class collected so far. This is expected to be useful when dealing with imbalanced classes. With regards to the instance selection criterion, the authors consider a hybrid strategy combining uncertainty sampling and random sampling, since approaches solely based on uncertainty could ignore a concept change that is not close to the boundary. Woźniak et al. (2023) recently proposed another ensemble-based active learning strategy where the data points to be labeled are selected from the current chunk using the budget labeling active learning strategy introduced by Zyblewski et al. (2020). According to this approach, the learner selects both random and informative data points, where the informativeness is determined using the support function threshold, which in the case of binary classification problems can be interpreted as a distance from the decision boundary.

Algorithm 4 Weight adjustment for dynamic classifiers

```
Require: a labeled observation $(\mathbf{x}_t, y_t)$, number of classes $K$, number of dynamic classifiers $D$, current
    weights $\omega_d$ with $d = 1, \ldots, D$, DCIR for each class DCIR$\kappa$ for $\kappa \in K$.
    if DCIR $[y_t] < \frac{1}{K}$ then                         ▷ Check if it belongs to the minority class
        for $d$ in $(1, D)$ do
            if $C_d(\mathbf{x}_t) = y_t$ then                  ▷ Check if the prediction made by $C_d$ is correct
                $\omega_d \leftarrow \omega_d\left(1 + \frac{1}{D}\right)$      ▷ Increase weight of classifier $C_d$
            else
                $\omega_d \leftarrow \omega_d\left(1 - \frac{1}{D}\right)$      ▷ Decrease weight of classifier $C_d$
            end if
        end for
    end if
```

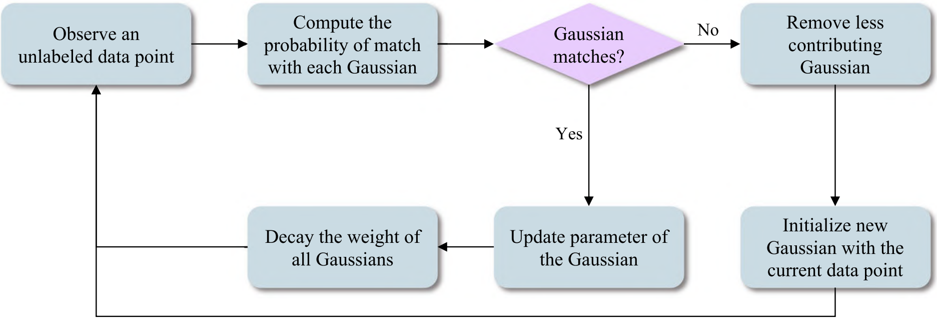

Observe an unlabeled data point
Compute the probability of match with each Gaussian
Gaussian matches?
No
Remove less contributing Gaussian
Yes
Decay the weight of all Gaussians
Update parameter of the Gaussian
Initialize new Gaussian with the current data point

Fig. 10 Main steps of the growing Gaussian mixture model used by Mohamad et al. (2018)

Another way to perform online active learning in time-varying data streams is to use clustering-based approaches. Halder et al. (2023) extended the framework based on stable and dynamic classifiers by introducing a clustering step that aims to train the new stable classifier C_s on the most informative and representative instances from each data block. Similarly, Ienco et al. (2013) investigated a clustering-based approach in a batch-based scenario, where only a fraction of the incoming block of observations can be labeled. They extend the pre-clustering approach (Nguyen & Smeulders, 2004), which had been previously studied in the pool-based scenario, to the stream-based case. The sampling strategy takes into account an extra-cluster metric, to sort the clusters, and an intra-cluster one, to sort the observations within each cluster. When a new batch arrives, observations are clustered, and clusters are sorted based on the homogeneity of the clusters, which is measured taking into account the number of (predicted) classes within each cluster. If a cluster is balanced in the number of expected classes, it is regarded to as an uncertain cluster that covers a more difficult area of the input space. Within each cluster, the certainty of an observations is determined by its representativeness, namely the distance from the centroid, and the uncertainty, meant as the maximum a posterior probability among all the predicted classes for $\mathbf{x}_t$. When the clusters and observations are ranked, the learner starts to iteratively ask the observations label in an alternate fashion. To sample the most representative data points from each batch, Zhang et al. (2023) suggested the use of density-peak clustering and recognize the incomplete clusters in the dynamic feature space through the altitude of these data points. This allows to query the observations belonging to those regions in the following iterations.

Recently, Yin et al. (2023) proposed an adaptive data stream classification method based on microclustering. After initializing micro-clusters from the initial training data, they collected new labels using a mixed strategy that combines random sampling with a class-weighted margin score. Then, the micro-cluster learning model is dynamically updated to adapt to the presence of concept drifts.

Another approach that tries to exploit the clustering nature of the incoming observations has been proposed by Mohamad et al. (2018), with the use of bi-criteria active learning algorithm that considers both density in the input space and label uncertainty. The density-based criterion makes use of the growing Gaussian mixture model proposed (GGMM) by Bouchachia and Vanaret (2014), which is used to find clusters in the data and estimate its density. This model creates a new cluster when a new data point $\mathbf{x}_t$ has a Mahalanobis distance greater than a given closeness threshold from the nearest cluster, among the currently available ones. A flowchart describing the main steps of the GGMM is depicted in Fig. 10.

A Bayesian logistic regression model is used for addressing the label uncertainty criterion and the concept drift. As the classifier parameters $\mathbf{w}_t$ are assumed to evolve over time, the model is incrementally updated using a discrepancy measure, which is computed as the difference between the uncertainty of the model in $\mathbf{x}_t$ before and after the true label y_t is added to the training set. The query strategy follows the b-sampling approach, in trying to sample, with high probability, the observations that contribute the most to the current error. The combination of density and uncertainty is also employed by Liu et al. (2021), who proposed a cognitive dual query strategy for online active learning in the presence of concept drifts and noise. The local density measure is used to obtain representative instances and the uncertainty criterion aim to select data points where the classifier is less confident. The cognitive aspect takes into account Ebbinghaus's law of human memory (Ebbinghaus, 2013) to determine an optimal replacement policy. The proposed strategy tries to tackle both gradual and abrupt drifts. The drift is generally considered as a change in the underlying joint probability distribution from one time step t to another, namely $p_t(\mathbf{x}, y) \neq p_{t+1}(\mathbf{x}, y)$. The local density of an observation $\mathbf{x}_t$ is defined by the number of times that $\mathbf{x}_t$ is the nearest neighbor of other instances (Ienco et al., 2014). Since we are in an online framework, the authors proposed to measure the local density using a sliding window model, referred to as a cognition window. Based on the concept of memory strength, the model determines when the current window is full and needs to be updated. Finally, the labeling decision is taken by using two thresholds, one for the local density and one for the classifier uncertainty.

A different sliding window-based online active learning strategy is the one proposed by Kurlej and Woźniak (2011). The authors proposed a sliding window approach based on a nearest neighbors classifier. The reference set for the k-nearest neighbors model is a window, and it is updated in two ways: in a first-in-first-out manner or using the examples selected by the active learning strategy. Since the reference set is updated over time, this method can effectively deal with concept drift and time-varying data streams. The sampling strategy is also based on two criteria. The first one is similar to the margin-based approaches, an instance is queried if it has a low distance from two observations belonging to different classes. The second criterion, similar to the greedy sampling strategy, seeks observations that have a large minimum distance from the observations in the current reference set. Both criteria are implemented by setting a threshold on the distances.

A simpler approach for taking into account the time-varying aspect of evolving data stream is to force the model to focus on the most recent observations. Along these lines, Chu et al. (2011) propose a framework based on a Bayesian probit model and a time-decay variant. Online Bayesian learning is used to maintain a posterior distribution of the weight vector of a linear classifier over time $\mathbf{w}_t$, and the time-decay strategies are employed to tackle the concept drift and give more importance to recent observations. They also propose an online approximation technique that can handle weighted examples, which is based upon Minka (2001). They tested different sampling strategies, built upon an online probit classifier. The instance selection criteria are based on entropy, function-value, and random sampling.

3.3 Evolving fuzzy systems approaches

An alternative way to take into account the time-varying nature of evolving data streams is the use of evolving fuzzy systems (EFS) (Lughofer, 2011), which are soft computing techniques that can efficiently deal with novelty and knowledge expansion. EFS are self-developing, self-learning fuzzy rule-based or neuro-fuzzy systems that self-adapt both their parameters and their structure on-the-fly. They try to mimic human-like reasoning by modeling it with a dynamically developing fuzzy rule-based structure and implementing it utilizing data streams using a formal learning process. The basic rule structure of a fuzzy model is given by

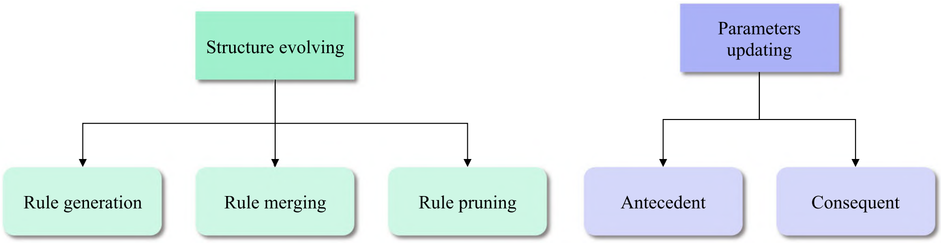


Fig. 11 Learning modules of an EFS (Ge & Zeng, 2020)

$$\begin{aligned} \text{Rule}_i : \textbf{if } (x_1 \textbf{ is } X_{i1}) \textbf{ and } \dots \textbf{ and } (x_n \textbf{ is } X_{in}) \\ \textbf{then } (y_i = a_{i0} + a_{i1}x_1 + \cdots + a_{in}x_n) \end{aligned} \tag{35}$$

where Rule $_i$ with $i = 1, 2, \dots, R$ is one of several fuzzy rules in the current rule base; $x_j (j = 1, 2, \dots, n)$ are input variables; y_i denotes the output of the ith fuzzy rule; X_{ij} denotes the jth prototype (focal point) of the ith fuzzy rule; a_{ij} denotes the jth parameter of the ith fuzzy rule. For a more thorough discussion on EFS and their use in online learning, please see (Lughofer, 2017, 2011; Ge & Zeng, 2020; Gu et al., 2022). The main components of an EFS are shown in Fig. 11. The two key components of an EFS are the structure evolving scheme, which contains the rule generation and simplification modules, and the parameters updating scheme. The rule generation module defines when a new rule needs to be added to the current model. The rule merging and pruning steps simplify the models by removing redundant rules and combining two rules when their similarity is larger than a given threshold. The parameter updating modules are used to keep track of the model evolution. These learning modules are used to update the EFS every time a new labeled example $(\mathbf{x}_t, y_t)$ is made available.

The first single-pass active learning approach based on the use of evolving classification models has been proposed by Lughofer (2012). The proposed algorithm is based on two key concepts, conflict and ignorance. The former is related to an incoming data point lying close to the boundary between any two classes; the latter considers the distance of the incoming observation from the currently labeled training set, in the feature space. This suggests that the data point falls within a region that has not been thoroughly explored by the learner. Later on, Lughofer and Pratama (2018) also proposed the first online active learning approach for evolving regression models. Similarly to their previous work (Lughofer, 2012), the authors consider the ignorance about the input space in the instance selection criterion. Moreover, they also consider the uncertainty in the model outputs and in the model parameters. The predictive uncertainty is assessed in terms of confidence intervals using locally adaptive error bars. The error bars are inspired by Škrjanc (2009) and the authors propose a new merging approach for dealing with the case of overlapping fuzzy rules. The uncertainty in the model parameters is instead evaluated using the A-optimality criterion, which will be discussed in Sect. 3.4 together with other alphabetic optimality criteria. Instead of leveraging the uncertainty about the output, Pratama et al. (2015) set a dynamic threshold based on the variable uncertainty strategy introduced by Zliobaite et al. (2014) while trying to address the what-to-learn question in the training of a recurrent fuzzy classifier. The key idea is that the model is iteratively retrained using data points that fall within rules with low support, which were formed using the smallest amount of observations. Recently, Lughofer and Škrjanc (2023) proposed an online active learning strategy for fuzzy models based on three criteria.

- D-optimality in the consequent space to reduce parameter uncertainty, as in Cacciarelli et al. (2022b).

- Overlap degree in the antecedent space to reduce the number of data points lying in the overlap regions of two different rules.
- Novelty content in the antecedent space, indicating the required knowledge expansion through rule evolution.

A different kind of threshold, based on the spherical potential theory, has been suggested by Subramanian et al. (2014), with the proposal of a meta-cognitive component that evaluates the novelty content of incoming data points. This is done using a knowledge measure represented by the spherical potential, which has been thoroughly investigated in kernel-based approaches (Hoffmann, 2007). The spherical potential is used to set a threshold and decide whether to add a new rule to capture the knowledge in the current sample. It should be noted that the authors also used a threshold based on the prediction error, which could not be used with scarcity of labels. The prediction error is assessed using the hinge loss error function (Suresh et al., 2008; Zhang, 2004).

Fuzzy models have also been used to solve computer vision tasks. Weigl et al. (2016) analyze the visual inspection quality control case, which is also considered by Rožanec et al. (2022). They assess the usefulness of the images in a single-pass manner, but the instances that are selected to be queried are accumulated in a buffer, which is later on assigned to an oracle for labeling. Choosing the size of the buffer represents a trade-off problem between timely updating the classifier and requiring continuous interventions from a human annotator. The active learning strategy works by setting a threshold on the certainty of the model with regards to the incoming data points. The authors take into account two model classes, a random forest classifier and an evolving fuzzy classifier. When using random forest, certainty is computed using the best-versus-second-best margin score. Instead, when using evolving fuzzy classifiers, the sample selection criterion takes into account the conflict and ignorance concepts as in Lughofer (2012).

Finally, Cernuda et al. (2014) combine the use of fuzzy models with a sampling approach inspired by the multivariate statistical process control literature. Indeed, using a latent structure model, they propose a query strategy based on the Hotelling T^2 and the squared prediction error (SPE) statistics, which have been extensively used in anomaly detection problems (Cacciarelli & Kulahci, 2022; Gajjar et al., 2018; Vanhatalo & Kulahci, 2016; Vanhatalo et al., 2017). Ge (2014) used these statistics for pool-based active learning in conjunction with a principal component regression model. The key idea is to use the Hotelling T^2 and the SPE statistics to measure the distance between the currently labeled training set and a new unlabeled data point. A high value in one of the two statistics would most likely suggest that the new observation is violating the current model, and thus its inclusion in the training set could bring some valuable information. Similarly, Cernuda et al. (2014) use the Hotelling T^2 and the SPE statistics with a partial least squares model. Then, when a new observation is added to the training set, they retrain a TS fuzzy model using a sliding window approach.

3.4 Experimental design and bandit approaches

Optimal experimental design (Karlin & Studden, 1966) is a research field that is closely related to active learning. It deals with the design of experiments that allow for efficient estimation of model parameters or improved prediction performance while minimizing the number of required labeled examples, also referred to as the number of runs N. Many optimality criteria have been developed in thriving to strike a balance between efficient use of resources and ensuring good performance of the model. The traditional framework of optimal experimental designs focuses on linear regression models of the form

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{36}$$

where, given d input variables, y is a $N \times 1$ vector of response variables, $\mathbf{X}$ is a $N \times d$ model matrix, $\boldsymbol{\beta}$ is a $d \times 1$ vector of regression coefficients, and $\boldsymbol{\varepsilon}$ is a $N \times 1$ vector representing the noise, with covariance matrix $\sigma^2\mathbf{I}$. If the matrix $\mathbf{X}^\top\mathbf{X}$ is of full rank, an ordinary least square (OLS) estimator for $\boldsymbol{\beta}$ can be obtained using

$$\widehat{\boldsymbol{\beta}} = \left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{y} \tag{37}$$

In general, design optimality criteria leverage the information contained in the moment matrix, which is defined as $\mathbf{M} = \mathbf{X}^\top\mathbf{X}/N$. The matrix $\mathbf{X}^\top\mathbf{X}$ plays a crucial role in the estimation of the model coefficients $\boldsymbol{\beta}$, and it is important to perceive information about the design geometry. Indeed, with Gaussian noise characterized by $\boldsymbol{\varepsilon} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}\right)$, we know that

$$\widehat{\boldsymbol{\beta}} \mid \mathbf{X} \sim \mathcal{N}\left(\boldsymbol{\beta}, \left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\sigma^2\right) \tag{38}$$

and we can define a $100(1-\alpha)\%$ confidence ellipsoid related to the solutions of $\boldsymbol{\beta}$ using

$$\frac{(\mathbf{b}-\widehat{\boldsymbol{\beta}})^\top\left(\mathbf{X}^\top\mathbf{X}\right)(\mathbf{b}-\widehat{\boldsymbol{\beta}})}{ds^2} \leq F_{\alpha,d,N-d} \tag{39}$$

where s^2 represents the residual mean square, $F_{\alpha,d,N-d}$ is the $100(1-\alpha)$ percentile derived from the Fisher distribution, and $\mathbf{b}$ indicates all the possible vectors that could be the true model parameter $\boldsymbol{\beta}$. The ellipsoid can also be expressed as $(\mathbf{b}-\widehat{\boldsymbol{\beta}})^\top\left(\mathbf{X}^\top\mathbf{X}\right)(\mathbf{b}-\widehat{\boldsymbol{\beta}}) \leq C$, where $C = ds^2F_{\alpha,d,N-d}$. The volume of this ellipsoid is inversely proportional to the square root of the determinant of $\mathbf{X}^\top\mathbf{X}$, and the length of its axes is proportional to $1/\lambda_i$, where λ_i represents the ith eigenvalue of $\mathbf{X}^\top\mathbf{X}$, with $i = 1, \ldots, d$. The so-called alphabetic optimality criteria pursuit efficient designs by exploiting these properties (Kiefer, 1959). The most commonly employed optimality criteria for good parameter estimation are A-, D- and E-optimality:

- *A-optimality.* This criterion pursues good model parameter estimation by minimizing the sum of the variances of the regression coefficients. Knowing that the coefficients variances appear on the diagonal of the matrix $\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}$, it can be shown that an A-optimal design is given by a design $\mathcal{D}^*$ that satisfies $\min_{\mathcal{D}} \operatorname{tr}[\mathbf{M}(\mathcal{D})]^{-1} = \operatorname{tr}\left[\mathbf{M}\left(\mathcal{D}^*\right)\right]^{-1}$.
- *D-optimality.* This criterion takes into account both the variance and covariance of the regression coefficients, directly minimizing the total volume of the confidence ellipsoid (Myers et al., 2016). A D-optimal design is given by a design $\mathcal{D}^*$ that satisfies $\max_{\mathcal{D}} |\mathbf{M}(\mathcal{D})| = |\mathbf{M}\left(\mathcal{D}^*\right)|$ (John & Draper, 1975).
- *E-optimality.* This strategy tries to shrink the ellipsoid by minimizing the maximum eigenvalue of the covariance matrix.

The geometrical intuition behind these criteria is illustrated, in the two-dimensional case, in Fig. 12.

Finally, there are also optimality criteria that focus on developing models with good predictive properties. Within this class, *G-optimality* represents a criterion that is used to seek protection against the worst-case prediction variance in a region of interest $\mathcal{R}$. This is achieved by solving

$$\min_{\mathcal{D}}\left[\max_{\mathbf{x}\in\mathcal{R}} v(\mathbf{x})\right] \tag{40}$$

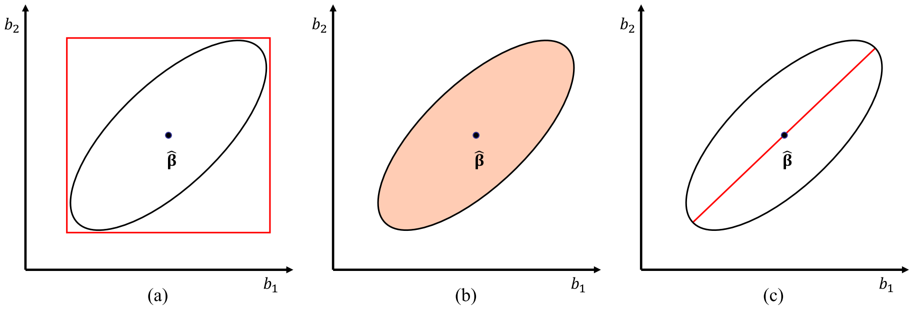


Fig. 12 Confidence ellipsoid around the model parameters and optimality criteria: A-optimality (**a**) shrinks the hyperrectangular enclosing the confidence ellipsoid (Asprey & Macchietto, 2002; Galvanin, 2010), D-optimality (**b**) aim to shrink the total volume of the ellipsoid, and E-optimality (**c**) tries to reduce the length of the longest axis (Jamieson, 2018)

where $v(\mathbf{x})$ represents the scaled prediction variance of the current model in the data point $\mathbf{x}$, which can be computed as

$$v(\mathbf{x}) = N\mathbf{x}^{(m)\mathrm{T}} \left(\mathbf{X}^{\top}\mathbf{X}\right)^{-1} \mathbf{x}^{(m)} \tag{41}$$

where $\mathbf{x}^{(m)}$ represents the data point where the variance is being estimated, expanded to the model form. It should be noted that G-optimality can be highly influenced by anomalous observations, as it protects against the highest possible variance over all the region $\mathcal{R}$. This issue can be tackled by using *I*- or *V-optimality*, which estimate the overall prediction variance over $\mathcal{R}$ by integrating or averaging, respectively. For a more extensive discussion on optimal designs, please see Montgomery (2012) or Myers et al. (2016).

The use of optimality criteria has proven to be highly beneficial in offline experimental design, allowing practitioners to pre-determine the location of each design point with ease. However, these methods require modification to be applied in a stream-based scenario where data points arrive sequentially. A common approach for obtaining a near-optimal design with streaming observational data is represented by thresholding. Riquelme (2017) proposed a thresholding algorithm for online active linear regression, which is related to the A-optimality criterion. Their approach uses a norm-thresholding algorithm, where only observations with large, scaled norms are selected. The design is augmented with the observations $\mathbf{x}$ whose norm exceeds a threshold Γ given by

$$\mathbb{P}(\|\mathbf{x}\| \geq \Gamma) = \alpha \tag{42}$$

where α is the ratio of observations we are willing to label out of the incoming data stream. Another approach related to the A-optimality criterion was proposed by Fontaine et al. (2021), who studied online optimal design under heteroskedasticity assumptions, with the objective of optimally allocating the total labeling budget between covariates in order to balance the variance of each estimated coefficient. Cacciarelli et al. (2022b) further extended the thresholding approach introduced by Riquelme (2017) by proposing a conditional D-optimality (CDO) algorithm. The terms conditional refers to the fact the design is marginally optimal, given an initial set of labeled observations to be augmented. The main steps of the CDO approach are reported in Algorithm 5. The authors exploited the connection between D-optimality and prediction variance previously highlighted by Myers et al. (2016). The sampling strategy selects observations by setting a threshold Γ given by

$$\mathbb{P}\left(\mathbf{x}_t^\top\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{x}_t \geq \Gamma\right) = \alpha \tag{43}$$

where $\mathbf{X}$ is the current set of labeled observations and $\mathbf{x}_t$ is the data point that is currently under evaluation. The threshold is estimated using kernel density estimation (KDE) on a set of j unlabeled observations, which are taken passively from the data stream without querying any label. This provides an initial set of data, referred to as warm-up set, that can be used to estimate the covariance matrix and the threshold.

Algorithm 5 Online active learning using CDO

```
Require: an initial random design $\mathbf{X}$, a data stream $\mathbf{S}$, a warm-up length $j$, a sampling rate $\alpha$, a budget $B$
  $t \leftarrow 1$                                                        ▷ Timestamp
  $c \leftarrow 0$                                                        ▷ Labeling cost
  Set $\mathbf{W} = \varnothing$                                          ▷ Warm-up set to estimate $\mathbf{\Sigma}$ and $\Gamma$
  while $t \leq j$ do
    Observe incoming data point $\mathbf{x}_t \in \mathbf{S}$
    Select $\mathbf{x}_t$ : $\mathbf{W} = \mathbf{W} \cup \mathbf{x}_t$
    $t \leftarrow t + 1$
  end while
  Estimate the covariance matrix $\Sigma$ of $\mathbf{W}$ and perform eigendecomposition $\mathbf{\Sigma} = \mathbf{U}\Lambda\mathbf{U}^\top$
  Whiten the initial design by computing $\mathbf{Z} = \Lambda^{-1/2}\mathbf{U}^\top\mathbf{X}$
  Whiten the warm-up observations by computing $\mathbf{V} = \Lambda^{-1/2}\mathbf{U}^\top\mathbf{W}$
  Estimate $\Gamma$ using KDE on $\mathbf{V}$ with the desired sampling rate $\alpha$ using Equation 43 with $Z$ and $V$
  while $c \leq B$ and $t \leq |\mathbf{S}|$ do
    Observe incoming data point $\mathbf{x}_t \in \mathbf{S}$
    Whiten $\mathbf{x}_t$ by computing $\mathbf{z}_t = \Lambda^{-1/2}\mathbf{U}^\top\mathbf{x}_t$
    if $\mathbf{z}_t^\top(\mathbf{Z}^\top\mathbf{Z})^{-1}\mathbf{z}_t \geq \Gamma$ then
      Ask for the label $y_i$ and augment the labeled dataset: $\mathbf{Z} \leftarrow \mathbf{Z} \cup \mathbf{z}_t$
      $c \leftarrow c + 1$                                                ▷ Pay for the label
      Update threshold $\Gamma$ to measure the prediction variance of the enlarged design
    else
      Discard $\mathbf{x}_t$
    end if
    $t \leftarrow t + 1$
  end while
```

Cacciarelli et al. (2023) also investigated how the presence of outliers affect the performance of online active linear regression strategies. They showed how the design optimality-based sampling strategies might be attracted to outliers, whose inclusion in the design eventually degrades the predictive performance of the model. This issue can be tackled by bounding the search area of the learner with two thresholds, as in

$$\mathbb{P}\left(\Gamma_1 \leq \mathbf{x}_t^\top\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{x}_t \leq \Gamma_2\right) = \alpha \tag{44}$$

where the choice of Γ_2 represents a trade-off between seeking protection against outliers and exploring uncertain regions of the input space.

The norm-thresholding approach was also extended by Riquelme et al. (2017a) to the case where the learner tries to estimate uniformly well a set of models, given a shared budget. This scenario is similar to a multi-armed bandit (MAB) problem where the learner wants to estimate the mean of a finite set of arms by setting a budget on the number of allowed pulls (Ruan et al., 2020; Audibert & Munos, 2010; Jamieson & Nowak, 2014; Soare et al., 2013). The authors propose a trace upper confidence bound (UCB) algorithm to simultaneously estimate the difficulty of each model and allocate the shared labeling budget proportionally to these estimates. UCB is a common algorithm used in MAB problems to balance exploration and exploitation (Carpentier et al., 2015; Garivier & Moulines, 2008), which takes into account the predicted mean value and the predicted standard deviation, weighted by an adjustable parameter (Thompson et al., 2022). This allows to balance the exploitation of data points with a high predicted value and the exploration of areas with high uncertainty.

In general, MAB problems can be seen as a special case of sequential experimental design, where the goal is to sequentially choose experiments to perform with the aim of maximizing some outcome. The typical framework of a MAB problem can be regarded as an optimization problem where the learner must identify the option or arm with the highest reward, among a set of available arms characterized by different reward distributions. Both MAB and active learning paradigms involve a sequential decision-making process where the learner aims to maximize a reward or improve model accuracy by selecting an arm to pull or a data point to label, respectively, and receiving feedback (in the form of a reward or label request) for each selection. There are two main approaches to tackle MAB problems:

- *Regret minimization.* This approach is coherent with the objective of maximizing the cumulative reward observed over many trials. In this case, the learner must balance exploration, namely trying out different arms to learn more about the reward distributions, with exploitation, i.e., using current knowledge to choose the most promising arm. These kinds of algorithms strike a balance between learning a good model and obtaining high rewards. A few examples might be treatment design, online advertising and recommender systems.
- *Pure exploration.* In this case, we are interested in finding the most promising arm, with a certain confidence or given a fixed budget on the number of pulls. To do so, the objective is to learn a good model while minimizing the number of measurements or labels required. This scenario is suggested in circumstances where, due to safety constraints, we are not given complete freedom to change the variable levels and we are mostly interested in understanding the underlying model governing the system. Possible examples include drug discovery or soft sensor development (Fortuna et al., 2007; Shi & Xiong, 2018; Chan et al., 2018; Tang et al., 2018).

The pure exploration approach is particularly useful when coupled with the study of linear bandits, which are a type of contextual bandit algorithms that assume a linear relationship between the features of the context and the expected reward of each arm. In this type of problem, when an arm $\mathrm{x} \in \mathcal{X}$ is pulled, the learner observes a reward $r(\mathbf{x})$ that depends on an unknown parameter $\boldsymbol{\theta}^* \in \mathbb{R}^d$ according to the linear model

$$r(\mathbf{x}) = \mathbf{x}^\top \boldsymbol{\theta}^* + \varepsilon \tag{45}$$

where ε is a zero-mean i.i.d. noise. This is similar to active linear regression in that, in both cases, the learner aims to select the most informative data points to learn about the underlying model or system (Audibert & Munos, 2010; Jamieson & Nowak, 2014). Soare et al. (2014), investigated this problem, in the offline setting, using the G-optimality criterion and a newly proposed $\mathcal{XY}$-allocation algorithm. Jedra and Proutiere (2020) proposed a fixed-confidence algorithm for the same problem, while Azizi et al. (2022) analyzed the fixed-budget case, extending the framework to the case where the underlying model is represented by a generalized linear model (Filippi et al., 2010). An interesting variant of this problem is presented in the study of transductive experimental designs. A transductive design is a problem where we can pull arms from a set $\mathcal{X} \in \mathbb{R}^d$, with the objective of identifying the best arm or improve the predictions over a separate set of observations $\mathcal{Z} \in \mathbb{R}^d$, which is given, in an unlabeled form, beforehand. A practical example of this case is when we are trying to infer the user preferences over a set of products, but we can only do that by pulling arms from a limited set of free trials. Alternatively, we might be interested in estimating the efficacy of a drug over a certain population, while doing experiments on a population with different characteristics. This problem has been tackled with an active learning approach by Yu et al. (2006), with the idea of exploiting unlabeled data points in $\mathcal{Z}$ while evaluating the informativeness of the data points in $\mathcal{X}$. The transductive case of sequential experimental design has been explored by Fiez et al. (2019), but instead of performing active learning, they were interested in inferring the best reward over $\mathcal{Z}$, only pulling the arms in $\mathcal{X}$. Finally, this has been extended to the online scenario by Camilleri et al. (2021), balancing the trade-off between time complexity and label complexity, namely between the number of unlabeled observations spanned and the number of labels queried in order to stop the learning procedure and declare the best-arm.

In addition to MAB, reinforcement learning-based approaches can also be applied to active learning in order to optimize a decision-making policy that balances the exploration of uncertain data with the exploitation of information learned from previous observations. This can be particularly useful in applications where the goal is to maximize the expected cumulative reward over time, such as in robotics or game playing. Compared to MAB, reinforcement learning-based approaches offer a more general and flexible framework for active learning, allowing for a wider range of problem formulations and feedback signals (Menard et al., 2021; Fang et al., 2017; Rudovic et al., 2019). One approach to combining active learning and reinforcement learning is through modeling the sampling routine as a contextual-bandit problem, as proposed by Wassermann et al. (2019). In this approach, the rewards are based on the usefulness of the query behavior of the learner. The key intuition behind the use of reinforcement learning in online active learning is that the learner gets feedback after the requested label, based on how useful the request actually was. In contrast to the traditional active learning view, where most of the effort is dedicated to the instance selection phase, the learner is penalized ex-post for querying useless instances. The learner gets a positive reward ρ^+ if it asks for the label when it would have otherwise predicted the wrong class, and a negative reward ρ^- when querying was unnecessary as the model would have predicted the right label. The contextual bandit problem is implemented by building an ensemble of different models, with each expert suggesting whether to query or not based on whether its prediction certainty exceeds a threshold Γ. The models are assigned a decision power based on how past suggestions were rewarded and how coherent they were with the other experts' suggestions. When an observation is sent to the oracle for labeling, the reward is computed, and the objective function of the learner is to maximize the total reward over a time horizon T.

Another reinforcement learning-based approach has been proposed by Woodward and Finn (2017). They considered the case where at each time step t the learner needs to decide whether to predict the label of the unlabeled data point $\mathbf{x}_t$ or pay to request its label y_t. The reinforcement learning framework is used to find an optimal policy $\pi^*(s_t)$ that takes into account the cost of asking for a label and the cost of making an incorrect prediction, where s_t represents the state that is given in input at the time$t$ to a policy $\pi(s_t)$ that outputs the suggested action a_t. The authors approximate the action-value function using a long short-term memory (LSTM) neural network with a linear output layer. The optimal policy is determined by maximizing the long-term reward, after assigning a reward to a label request R_{req}, a correct prediction R_{corr}, and an incorrect prediction R_{inc}. It should be noted that R_{corr} and R_{inc} should be negative rewards, as they are associated with costly actions.

4 Evaluation strategies

The use of active learning approaches is becoming increasingly common in machine learning, allowing models to be trained more efficiently by selecting the most informative examples for labeling. To evaluate the performance of these approaches, it is typical to compare them to a passive random sampling strategy by generating learning curves that plot the model performance (e.g., accuracy, F1 score, or root mean square error) on a holdout test set over the number of labeled examples used for training. Learning curves are a useful tool for comparing the asymptotic performance of different strategies and their sample efficiency, with the slope of the curve reflecting the rate at which the model performance improves with additional labeled examples. A steeper slope indicates a more sample-efficient strategy. When multiple sampling strategies are being compared, a visual inspection of the learning curves may not be sufficient, and more rigorous statistical tests may be necessary. Reyes et al. (2018) recommend the use of non-parametric statistical tests to analyze the effectiveness of active learning strategies for classification tasks. The sign test (Steel, 1959) or the Wilkinson signed-ranks test (Wilcoxon, 1945) can be used to compare two strategies, while the Friedman test (Friedman, 1940), the Friedman aligned-ranks test (Hodges & Lehmann, 1962), the Friedman test with Iman-Davenport correction (Iman & Davenport, 1980), or the Quade test (Quade, 1979) can be used when evaluating more than two strategies. These statistical tests can provide insight into whether the difference in performance between the active learning and passive random sampling strategies is statistically significant.

Algorithm 6 Prequential evaluation for online active learning

```
Require: an initial model $\mathbf{w}_0$, a data stream $\mathbf{S}$, a budget $B$, an active learning strategy $Q$.
  $t \leftarrow 1$                                            ▷ Timestamp
  $\mathbf{P} \leftarrow \emptyset$                           ▷ Storing predictions
  while $c \leq B$ and $i \leq |\mathbf{S}|$ do
    Observe the data point $\mathbf{x}_t \in \mathbf{S}$
    Predict the label $\widehat{y}_t$ and store it: $\mathbf{P} \leftarrow \mathbf{P} \cup \widehat{y}_t$
    if $Q(\mathbf{x}_t) = \texttt{True}$ then                ▷ Sampling decision
      Ask for the true label $y_t$ and update the model
      $c \leftarrow c + 1$                                    ▷ Pay for the label
    else
      Discard $\mathbf{x}_t$
    end if
    $t \leftarrow t + 1$
  end while
```

Overall, the use of learning curves and statistical tests can provide valuable insights into the effectiveness and efficiency of different active learning strategies. By understanding the statistical significance of differences in performance between these strategies, researchers can make informed decisions about which approaches are more effective for a particular task or dataset. Furthermore, the choice of the evaluation scheme is crucial when assessing the performance of active learning approaches. If we use an evaluation scheme based on a holdout test set, at each learning step t the performance of the model is assessed using the same test set. This can be a reasonable approach if we are dealing with a stationary data stream, which does not evolve over time. Under these assumptions, using the same test set we might be able to better assess the prediction improvement as more labeled examples are included in the design. However, this approach might not be ideal when dealing with drifting data streams. In these circumstances, a prequential evaluation scheme can be more

Table 1 Evaluation strategies

Evaluation strategy	Works
Holdout test set	Desalvo et al. (2021), Wassermann et al. (2019), Rožanec et al. (2022), Narr et al. (2016), Ferdowsi et al. (2013), Bordes et al. (2005), Suzuki et al. (2021), Ghassemi et al. (2016), Qin et al. (2021), Woodward and Finn (2017), Riquelme et al. (2017b), Cacciarelli et al. (2022b), Cacciarelli et al. (2023), Manjah et al. (2023)
Prequential/test-then-train	Zhang et al. (2022), Pham et al. (2022), Castellani et al. (2022), Chu et al. (2011), Zhang et al. (2018), Krawczyk et al. (2018), Xu et al. (2016), Mohamad et al. (2020), Weigl et al. (2016), Ienco et al. (2013), Zhang et al. (2020a)

useful to monitor the evolution of the prediction error over time (Suárez-Cetrulo et al., 2021; Cerqueira et al., 2020; Tieppo et al., 2022; Cacciarelli & Boresta, 2021). In online learning, prequential evaluation is also referred to as test-then-train approach, and it involves using each incoming instance first to measure the prediction error, and then to be included in the training set (Suárez-Cetrulo et al., 2023). The main steps of the test-then-train approach are reported in Algorithm 6. The key idea is that at each time step t, we first test the model by making a prediction, then we decide whether to query the true labels and finally we update our model.

An in-depth analysis and discussion between the use of a holdout test set and the prequential evaluation scheme for streaming data has been provided by Gama et al. (2009, 2013), who suggested the use of a prequential evaluation scheme with forgetting mechanisms. For scenarios with imbalanced data streams, a specialized prequential variant of the area under the curve metric has been proposed by Brzezinski and Stefanowski (2015, 2017). From an implementation perspective, Bifet et al. (2010) developed an open source software suite called MOA for data stream mining, which includes both the holdout and prequential strategies. This framework has found widespread application in the evaluation of online active learning strategies, as evidenced by the studies conducted by Liu et al. (2021), Shan et al. (2019), Weigl et al. (2016), Zhang et al. (2020a), Alabdulrahman et al. (2016).

In Table 1, we categorize the studies based on the experimental protocols they employed to evaluate the sampling strategies. The table exclusively includes approaches where the evaluation strategy was explicitly defined. In most cases, when assessing active learning methods in the context of drifting data streams, a prequential approach is favored. Conversely, for scenarios where the methods are ill-suited to handle concept drifts, holdout test sets tend to be the preferred choice. In approaches not featured in the table, the evaluation strategies exhibited some variations or lacked explicit specification. For instance, in the work by Fujii and Kashima (2016), their evaluation strategy involved training models on the queried data and subsequently testing them with the entire dataset. This approach differs from the conventional test-then-train paradigm since, in this case, models are tested on data they encountered during training, at least in part. Another example is found in Zhu et al. (2007), who utilized a window-based approach, assessing prediction accuracy across all observations in the current batch. On a different note, Hao et al. (2018a) employed the per-round regret metric, which quantifies the loss difference between the forecaster and the best expert at each iteration of the active learning process. In some instances, none of the previously mentioned methods were employed, as the analysis took a more theoretical perspective. This is exemplified by the works of Dasgupta et al. (2005), Chae and Hong (2021), Huang et al. (2022). Lastly,

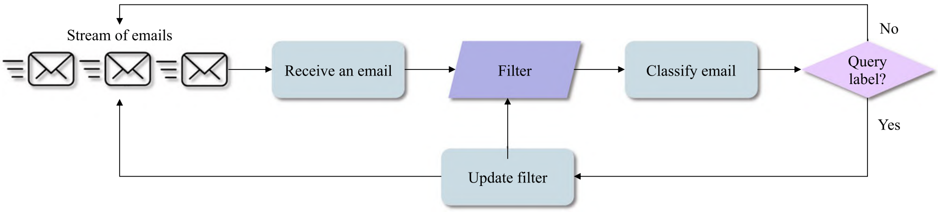


Fig. 13 Low-cost active spam filtering (Sculley, 2007)

bandit algorithms employed a distinct evaluation approach, often aiming to identify the most promising arm with a fixed confidence or budget. In the fixed confidence setting, performance typically hinges on comparing label complexity to problem dimensionality or the number of arms pulled, as observed in Fiez et al. (2019). Alternatively, regret or error metrics were evaluated against the required number of trials, as demonstrated in the studies by Riquelme et al. (2017a), Sudarsanam and Ravindran (2018), Fontaine et al. (2021).

5 Real-world applications and challenges

5.1 Applications

Online active learning has been recognized as a powerful technique in scenarios where data is arriving at a high velocity, labeling data is expensive, and it is infeasible to store all the unlabeled data before making a decision about which observations to query to update the model. In particular, these techniques have proven particularly useful in dynamic and ever-evolving environments, where models need to adapt to new data in real-time, by selectively querying the most informative instances. One of the first real-world applications of online active learning has been presented by Sculley (2007), who investigated the scenario of low-cost active spam filtering (Fig. 13) where a filter is updated online by selecting the most informative emails in real time. Another application of online active learning in the field of IT has been recently presented by Zhang et al. (2020a). They analyzed the scenario of network protocol identification and proposed a method (presented in Sect. 3.2) to select the most representative instances on the fly and adapt the model to dynamic data distributions.

Computer vision is another interesting area where online active learning can be applied. Deep learning models require a large amount of annotated data, making manual annotation of thousands of images one of the most challenging aspects of model development. However, it is important to note that the most effective deep active learning methods proposed so far are not easily adaptable to a stream-based setting. Many of these methods involve clustering or measuring pairwise similarity among image embeddings (Sener & Savarese, 2017; Agarwal et al., 2020; Ash et al., 2019; Citovsky et al., 2021; Prabhu et al., 2020), which cannot be easily done in a single-pass manner. As a result, most online applications of active learning in computer vision rely on the use of traditional models with uncertainty-based sampling. Narr et al. (2016) analyze the stream-based active learning problem for the classification of 3D objects. They used a mondrian forest classifier (Lakshminarayanan et al., 2014), which is an efficient alternative of random forest for the online learning scenario, and selected images with high classification uncertainty to be labeled. Rožanec et al. (2022) used online active learning to reduce the data labeling effort while performing vision-based process monitoring. Initially, features are extracted from the images using a pre-trained ResNet-18 model (He et al., 2015) and then, using the mutual information criterion (Kraskov et al., 2004), only $\sqrt{n}$ features (Hua et al., 2005) are retained to fit an online classifier, where n is the total number of observations in the training set. The authors combine a simple active learning strategy based on model uncertainty with five streaming classification algorithms, including Hoeffding tree (Hulten et al., 2001), Hoeffding adaptive tree (Bifet & Gavaldà, 2009), stochastic gradient tree (Gouk et al., 2019), streaming logistic regression, and streaming k-nearest neighbors. Recently, Saran et al. (2023) proposed a novel approach to streaming active learning with deep neural networks. Given a neural network with f with parameters θ, last-layer parameters θ_L, and the cross-entropy function ℓ, they compute the gradient representation of the data point $\mathbf{x}_t$, which is given by

$$g(\mathbf{x}_t) = \frac{\partial}{\partial \theta_L} \ell\left(f(\mathbf{x}_t; \theta), \widehat{y}_t\right) \tag{46}$$

where $\widehat{y}_t = \text{argmax}\, f(\mathbf{x}_t; \theta)$. Then, the data points to be included in the batch for training the model are chosen by using a probability p_t proportional to the contribution of the current example to the covariance matrix of the examples collected so far, as in

$$p_t \propto \det\left(\widehat{\Sigma}_t + g(\mathbf{x}_t)g(\mathbf{x}_t)^\top\right) \tag{47}$$

where $\widehat{\Sigma}_t$ is the covariance matrix of the data points that have been selected to be included int he current batch, up to the time step t.

Online active learning has also been explored for object detection tasks. Manjah et al. (2023) proposed a stream-based active distillation (SBAD) framework by combining the concepts of active learning and self-supervision as described in Sect. 2.3. The SBAD framework enables the deployment of scalable deep-learning models as it does not rely on human annotators and takes into account the imperfection of the oracle when distilling knowledge from a large teacher model to a lightweight student. Indeed, the authors suggest setting a threshold on the confidence of the images and only querying images with high confidence in trying to avoid confirmation bias. The threshold is determined using a warm-up phase, similarly to the approach proposed by Cacciarelli et al. (2022b) presented in Algorithm 5. The SBAD pipeline for model development and evaluation is reported in Fig. 14.

The problem of performing active learning for object detection with streaming data has also been explored by Beck et al. (2023). In the case of a camera placed on an autonomous vehicle, the collected data encompasses various scenarios, including clear weather, foggy conditions, and rainy weather, all of which require the model to perform effectively. However, the frequency of these scenarios can vary significantly. In situations where one scenario is prevalent, a passive sampling strategy could tend to sample very few examples from the most rare slices. Instead, the proposed streamline approach by attempts to smartly allocate the budget to obtain more observations from the slices where the model is under-performing. The case of autonomous cars was also considered by Yan et al. (2023), who used a diversity-based online active learning strategy to reduce false alarm rate and learn unseen faults.

Another interesting industrial application has been recently presented by Ghiasi et al. (2023). They proposed a deployable framework that combines a thermodynamics-based compressor model and a Gaussian Process-based surrogate model with an online active learning module. The objective of the study was to minimize the power absorbed by the machine during the boil off process of centrifugal compressor. In the proposed framework, the simulator, the surrogate model, and the optimizer interact in real time to determine the new experimental points.

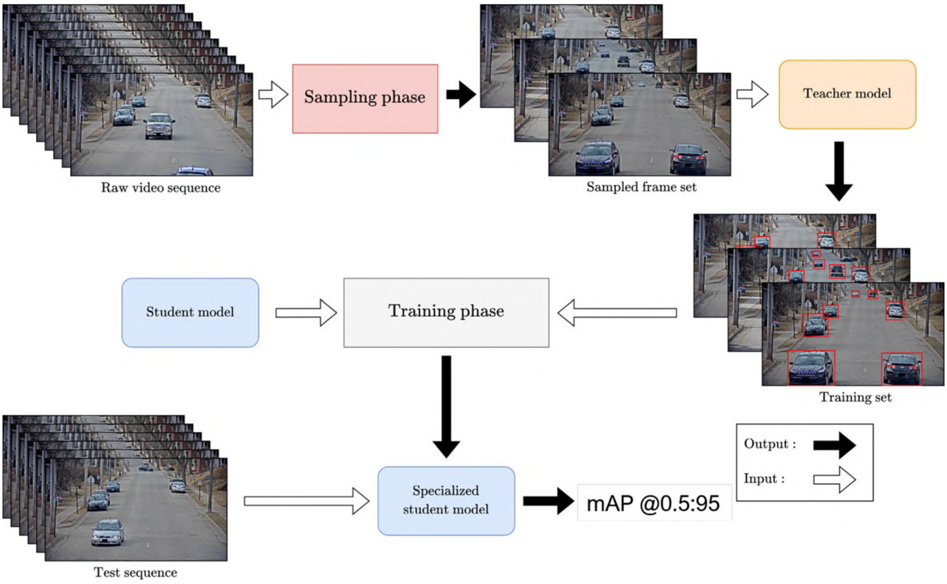


Fig. 14 SBAD framework (Manjah et al., 2023): sampling, fine-tuning and evaluation. The sampling is performed in a single-pass manner via thresholding

5.2 Challenges

When applying online active learning strategies to real-world problems, there are several potential issues to consider, including:

- *Algorithm scalability.* Online active learning algorithms need to be efficient and scalable to handle large datasets and high-velocity data streams. As the amount of data grows, the computational demands of active learning can become prohibitive, making it difficult to deploy in practice. The time required to make the sampling decision needs to be lower than the feed rate of the process being analyzed. If the algorithm is too slow, it may require a buffer, which reduces the benefits of online active learning.
- *Labeling quality.* Most online active learning strategies rely heavily on the quality of labeled data, which can be challenging to ensure in real-world scenarios. Human annotators may make errors, introduce biases, or interpret labeling instructions differently. For this reason, in real-life situations, it may be necessary to consider oracle imperfections like in the knowledge distillation case (Baykal et al., 2022). Another difficult aspect related to labeling quality is the delay or latency, which has been described in Sect. 2.2.3.
- *Data drift.* In real-world settings, data distributions may shift over time, making it challenging for models to adapt and continue providing accurate predictions. Changes in the data distribution may also affect the quality of the labeled data, as the criteria for selecting informative instances may become less effective. Methods from Sects. 3.2 and 3.3 should be used when dynamic and ever-changing behaviors are expected.
- *Model interpretability.* Besides simply asking for the most informative instances from a modeling perspective, it might be useful to provide additional information on why a particular instance is beneficial for improving the performance of the current model. In fields like healthcare and manufacturing this might help practitioners to improve their understanding of the underlying problem.

- *Evaluation.* When developing active learning methods from a research perspective, the different query strategies are evaluated assuming the ground-truth labels to be available for a held-out test set, or for the data stream being analyzed. However, in real life, the key motivation behind active learning is label scarcity and thus it might be difficult to thoroughly assess the effectiveness of the deployed sampling strategy.
- *Human-computer interaction.* In the context of active learning for data streams, the synergy between human labelers and computer systems plays a pivotal role in the labeling process. While the majority of online active learning methods focus on querying the most informative data points in real-time, we can distinguish between two distinct labeling scenarios:

    1. *Real-time annotation.* In most of the presented works, it is assumed that labels are immediately available when a data point is queried from the stream. This immediate access to true labels enables an optimized active learning routine, as the model can be promptly updated and can recommend exploration of new regions based on up-to-date information. However, this approach poses some implementation challenges that need to be addressed with the use of advanced data annotation tools (Feuz & Cook, 2013).
    2. *Postponed annotation.* There are cases where we must allow for a delay between data querying and labeling. For instance, methods that consider verification latency (Castellani et al., 2022; Pham et al., 2022) take into account the possibility of delayed labels. This is particularly relevant in situations where a physical quality inspection or medical treatment must occur before the label is revealed. Another example is in the training of deep neural networks, where real-time sampling from a data stream is necessary due to memory constraints (Manjah et al., 2023), but the labeling and model update phase may occur when a batch is collected, following a batch-mode active learning strategy (Ren et al., 2022).

6 Summary and future directions

This survey outlines the challenge of conducting active learning with data streams and investigates different approaches for selecting the most informative data points in real-time.

Table 2 provides a summary of the relevant state-of-the-art approaches, highlighting their main properties and settings. Our examination reveals that existing research has predominantly concentrated on creating online classification models, which can operate with both stationary and drifting data streams. However, there has been comparatively limited effort devoted to online active linear regression or dedicated to constructing online regression models in general.

We believe that there are several promising directions for future research in this field. First, we recommend further investigation into online active learning strategies specifically designed for regression models. Given the limited work in this area, there is a need for more advanced methods that can be applied to nonlinear models, beyond linear models or linear bandits. For example, there has been a recent spark of interest toward the use of Bayesian optimization for active learning in nonlinear regression problems (Mohamadi & Amindavar, 2020; Riis et al., 2022). Additionally, model-agnostic methods that can be applied to a variety of regression models could be valuable as they would provide a more

Table 2 Online active learning strategies: summary based on data processing capabilities, assumptions about the data stream, task of the model and model characteristics

Data processing	Data stream	Task	Model	Work(s)
Single-pass	Stationary	Classification	Single model	Cesa-Bianchi et al. (2004), Cesa-Bianchi et al. (2006), Dasgupta et al. (2005), Sculley (2007), Lu et al. (2016), Hao et al. (2018b), Ghassemi et al. (2016), Shah and Manwani (2020), Mohamad et al. (2020), Saran et al. (2023), Rožanec et al. (2022), Woodward and Finn (2017)
			Ensemble	Huang et al. (2022), Desalvo et al. (2021), Loy et al. (2012), Hao et al. (2018a), Chae and Hong (2021)
		Regression	Single model	Riquelme (2017), Fontaine et al. (2021), Cacciarelli et al. (2022b), Cacciarelli et al. (2023), Cacciarelli et al. (2022a)
		Object detection	Single model	Manjah et al. (2023)
	Drifting	Classification	Single model	Krawczyk et al. (2018), Castellani et al. (2022), Pham et al. (2022), Yin et al. (2023), Mohamad et al. (2018), Liu et al. (2021), Kurlej and Woźniak (2011), Chu et al. (2011)
			Ensemble	Zhang et al. (2020a), Shan et al. (2019), Zhang et al. (2018), Zhang et al. (2022)
	Evolving	Classification	Single model	Lughofer (2012), Pratama et al. (2015)
		Regression	Single model	Lughofer and Pratama (2018), Lughofer and Škrjanc (2023)
Batch	Stationary	Classification	Single model	Bordes et al. (2005), Qin et al. (2021), Fujii and Kashima (2016)
		Object detection	Single model	Beck et al. (2023)
	Drifting	Classification	Single model	Cheng et al. (2023), Martins et al. (2023), Ienco et al. (2013), Zhang et al. (2023), Yan et al. (2023)
			Ensemble	Zhu et al. (2007), Woźniak et al. (2023), Halder et al. (2023)
	Evolving	Classification	Single model	Subramanian et al. (2014), Weigl et al. (2016), Cernuda et al. (2014)

general solution to the problem. Second, we believe that there is potential for research into single-pass online sampling strategies for dynamic data streams. Ensemble models and batch-based approaches have been the dominant methods in online classification, but some of their assumptions or requirements may not hold in many real-world applications. For instance, in some applications, data may arrive in a continuous stream, and it may not be possible to divide it into batches due to time or memory constraints. In such cases, single-pass online sampling strategies that do not require the use or update of multiple models would be more practical. Moreover, it could be beneficial to develop online active learning strategies that are able to tackle all the types of distribution shifts introduced in Sect. 3.2. Finally, the combination of reinforcement learning and active learning in pool-based scenarios is an area of ongoing research. We believe that the study of online reinforcement learning to optimize sampling strategies could provide valuable insights into how to best perform active learning in dynamic environments.

7 Conclusion

The field of online active learning with data streams is a rapidly evolving and highly relevant area of research in machine learning. The ability to effectively learn from data streams in real-time is becoming increasingly important, as the amount of data generated by modern applications continues to grow at an exponential rate. However, obtaining annotated data to train complex prediction and decision-making models presents a major roadblock. This hinders the proper integration of artificial intelligence models with real-world applications such as healthcare, autonomous driving and industrial production. Our survey provides a comprehensive overview of the current state of the art in this field and highlights the challenges and opportunities that researchers face when developing methods for online active learning. We reviewed a wide range of strategies for selecting the most informative data points in online active learning, including methods based on uncertainty sampling, diversity sampling, query by committee, and reinforcement learning, among others. Our analysis has shown that these strategies have been applied in a variety of contexts, including online classification, online regression, and online semi-supervised learning. We hope that this survey will inspire further research in the field of online active learning with data streams and encourage the development of new and advanced methods for handling this type of data. In particular, we believe that there is significant potential for the development of model-agnostic and single-pass online active learning strategies that can be applied in practical settings.

Acknowledgements The authors gratefully acknowledge the support of the DTU Strategic Alliances Fund, which made this research possible. We would also like to extend our sincere thanks to John Sølve Tyssedal for his invaluable help and support throughout the project.

Author Contributions DC conceived and designed the analysis and wrote the paper. MK provided guidance and supervision throughout the work, reviewed the paper, and provided critical feedback to improve the manuscript.

Funding Open access funding provided by Technical University of Denmark The research reported in this paper was supported by the DTU Strategic Alliances Fund.

Data availability Not applicable.

Code Availability Not applicable.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

Ethical approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

References

Agarwal, S., Arora, H., Anand, S., et al. (2020). Contextual diversity for active learning. In *European conference on computer vision 2020*. https://doi.org/10.1007/978-3-030-58517-4_9. arXiv:2008.05723.

Aggarwal, C. C., Kong, X., Gu, Q., et al. (2014). *Data classification (Chapter: "Active learning: A survey")*. Taylor & Francis. http://charuaggarwal.net/active-survey.pdf.

Aguiar, G., Krawczyk, B., & Cano, A. (2023). A survey on learning from imbalanced data streams: Taxonomy, challenges, empirical study, and reproducible experimental framework. *Machine Learning*, 1–79.

Alabdulrahman, R., Viktor, H., & Paquet, E. (2016). An active learning approach for ensemble-based data stream mining. In *International conference on knowledge discovery and information retrieval, SCITEPRESS* (pp. 275–282).

Ash, J. T., Zhang, C., Krishnamurthy, A., et al. (2019). Deep batch active learning by diverse, uncertain gradient lower bounds. In *2020 international conference on learning representations*. arXiv:1906.03671.

Asprey, S., & Macchietto, S. (2002). Designing robust optimal dynamic experiments. *Journal of Process Control, 12*, 545–556. https://doi.org/10.1016/S0959-1524(01)00020-8

Audibert, J. Y., & Munos, R. (2010). Best arm identification in multi-armed bandits. In *COLT—23th conference on learning theory*. http://certis.enpc.fr/audibert/Mes%20articles/COLT10.pdf.

Avadhanula, V., Colini Baldeschi, R., Leonardi, S., et al. (2021). Stochastic bandits for multi-platform budget optimization in online advertising. *Proceedings of the Web Conference, 2021*, 2805–2817.

Azizi, M. J., Kveton, B., & Ghavamzadeh, M. (2022). Fixed-budget best-arm identification in structured bandits. In *Proceedings of the thirty-first international joint conference on artificial intelligence (IJCAI-22)*. https://www.ijcai.org/proceedings/2022/0388.pdf.

Baier, L., Schlör, T., Schöffer, J., et al. (2021). Detecting concept drift with neural network model uncertainty. In *Hawaii international conference on system sciences (HICSS) 2023*. arXiv:2107.01873.

Balcan, M. F., Broder, A., & Zhang, T. (2007). Margin based active learning. In *COLT—23th conference on learning theory 4739*. https://doi.org/10.1007/978-3-540-72927-3_5.

Bassily, R., Smith, A., & Thakurta, A. (2014). Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th annual symposium on foundations of computer science* (pp. 464–473). https://doi.org/10.1109/FOCS.2014.56.

Baum, E., & Lang, K. (1992). Query learning can work poorly when a human oracle is used. In *Proceedings of the IEEE international joint conference on neural networks*.

Baykal, C., Trinh, K., Iliopoulos, F., et al. (2022). *Robust active distillation*. arXiv:2210.01213.

Beck, N., Kothawade, S., Shenoy, P., et al. (2023). *Streamline: Streaming active learning for realistic multi-distributional settings*. arXiv preprint arXiv:2305.10643.

Bifet, A., & Gavaldà, R. (2007). Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 SIAM international conference on data mining* (pp. 443–448). https://doi.org/10.1137/1.9781611972771.42.

Bifet, A., & Gavaldà, R. (2009). Adaptive learning from evolving data streams. In *IDA 2009: Advances in intelligent data analysis VIII* (pp. 249–260). https://doi.org/10.1007/978-3-642-03915-7_22.

Bifet, A., Holmes, G., Pfahringer, B., et al. (2010). Moa: massive online analysis, a framework for stream classification and clustering. In *Proceedings of the first workshop on applications of pattern analysis, PMLR* (pp. 44–50)

Bisgaard, S., & Kulahci, M. (2011). *Time series analysis and forecasting by example*. New York: Wiley.

Bordes, A., Ertekin, S., Weston, J., et al. (2005). Fast kernel classifiers with online and active learning. *The Journal of Machine Learning Research, 6*. https://jmlr.csail.mit.edu/papers/v6/bordes05a.html.

Bouchachia, A., & Vanaret, C. (2014). Gt2fc: An online growing interval type-2 self-learning fuzzy classifier. *IEEE Transactions on Fuzzy Systems, 22*, 999–1018. https://doi.org/10.1109/TFUZZ.2013.2279554

Brzezinski, D., & Stefanowski, J. (2015). Prequential auc for classifier evaluation and drift detection in evolving data streams. In *3rd International workshop on new frontiers in mining complex patterns, (NFMCP 2014)* (pp. 87–101). https://doi.org/10.1007/978-3-319-17876-9_6.

Brzezinski, D., & Stefanowski, J. (2017). Prequential auc: Properties of the area under the roc curve for data streams with concept drift. *Knowledge and Information Systems, 52*, 531–562. https://doi.org/10.1007/s10115-017-1022-8

Burbidge, R., Rowland, J. J., & King, R.D. (2007). Active learning for regression based on query by committee. In *8th International conference on intelligent data engineering and automated learning, IDEAL 2007*. https://doi.org/10.1007/978-3-540-77226-2_22.

Cacciarelli, D., & Boresta, M. (2021). What drives a donor? A machine learning-based approach for predicting responses of nonprofit direct marketing campaigns. *International Journal of Nonprofit and Voluntary Sector Marketing*. https://doi.org/10.1002/nvsm.1724

Cacciarelli, D., & Kulahci, M. (2022). A novel fault detection and diagnosis approach based on orthogonal autoencoders. *Computers & Chemical Engineering, 163*, 107853. https://doi.org/10.1016/j.compchemeng.2022.107853

Cacciarelli, D., & Kulahci., M. (2023). Hidden dimensions of the data: PCA vs autoencoders. *Quality Engineering*, *35*, 741–750. https://doi.org/10.1080/08982112.2023.2231064

Cacciarelli, D., Kulahci, M., & Tyssedal, J. (2022a). Online active learning for soft sensor development using semi-supervised autoencoders. In *ICML 2022 workshop on adaptive experimental design and active learning in the real world*. arXiv:2212.13067.

Cacciarelli, D., Kulahci, M., & Tyssedal, J. S. (2022b). Stream-based active learning with linear models. *Knowledge-Based Systems, 254*, 109664. https://doi.org/10.1016/j.knosys.2022.109664.

Cacciarelli, D., Kulahci, M., & Tyssedal, J. S. (2023). Robust online active learning. *Quality and Reliability Engineering International*. https://doi.org/10.1002/qre.3392

Cai, W., Zhang, Y., & Zhou, J. (2013). Maximizing expected model change for active learning in regression. In *Proceedings—IEEE international conference on data mining, ICDM* (pp. 51–60). https://doi.org/10.1109/ICDM.2013.104.

Camilleri, R., Xiong, Z., Fazel, M., et al. (2021). Selective sampling for online best-arm identification. In *35th conference on neural information processing systems (NeurIPS 2021)*. arXiv:2110.14864.

Carcillo, F., Le Borgne, Y. A., Caelen, O., et al. (2017). An assessment of streaming active learning strategies for real-life credit card fraud detection. In *2017 IEEE international conference on data science and advanced analytics (DSAA)* (pp. 631–639). IEEE.

Carcillo, F., Le Borgne, Y. A., Caelen, O., et al. (2018). Streaming active learning strategies for real-life credit card fraud detection: Assessment and visualization. *International Journal of Data Science and Analytics, 5*, 285–300.

Carnein, M., & Trautmann, H. (2019). Customer segmentation based on transactional data using stream clustering. In *Advances in knowledge discovery and data mining: 23rd Pacific-Asia Conference, PAKDD 2019, Macau, China, April 14–17, 2019, proceedings, part I* (Vol. 23, pp. 280–292). Springer.

Carpentier, A., Lazaric, A., Ghavamzadeh, M., et al. (2015). Upper-confidence-bound algorithms for active learning in multi-armed bandits.

Castellani, A., Schmitt, S., & Hammer, B. (2022). Stream-based active learning with verification latency in non-stationary environments. https://doi.org/10.1007/978-3-031-15937-4_22. arXiv:2204.06822.

Cernuda, C., Lughofer, E., Mayr, G., et al. (2014). Incremental and decremental active learning for optimized self-adaptive calibration in viscose production. *Chemometrics and Intelligent Laboratory Systems, 138*, 14–29. https://doi.org/10.1016/j.chemolab.2014.07.008

Cerqueira, V., Torgo, L., & Mozetič, I. (2020). Evaluating time series forecasting models: An empirical study on performance estimation methods. *Machine Learning, 109*, 1997–2028. https://doi.org/10.1007/s10994-020-05910-7

Cesa-Bianchi, N., & Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge: Cambridge University Press. https://doi.org/10.1017/CBO9780511546921

Cesa-Bianchi, N., Gentile, C., Zaniboni, L. (2004). Worst-case analysis of selective sampling for linear-threshold algorithms. In *Advances in neural information processing systems*. https://proceedings.neurips.cc/paper_files/paper/2004/hash/92426b262d11b0ade77387cf8416e153-Abstract.html.

Cesa-Bianchi, N., Gentile, C., Zaniboni, L. (2006). Worst-case analysis of selective sampling for linear classification. *The Journal of Machine Learning Research, 7*. https://www.jmlr.org/papers/volume7/cesa-bianchi06b/cesa-bianchi06b.pdf.

Chae, J., & Hong, S. (2021). Stream-based active learning with multiple kernels. In *2021 international conference on information networking (ICOIN)* (pp. 718–722). https://doi.org/10.1109/ICOIN50884.2021.9333940.

Chan, L. L. T., Wu, Q. Y., & Chen, J. (2018). Dynamic soft sensors with active forward-update learning for selection of useful data from historical big database. *Chemometrics and Intelligent Laboratory Systems, 175*, 87–103. https://doi.org/10.1016/j.chemolab.2018.01.015

Cheng, J., Zheng, Z., Guo, Y., et al. (2023). Active broad learning with multi-objective evolution for data stream classification. *Complex & Intelligent Systems, 12*, 1–18.

Chu, W., Zinkevich, M., Li, L., et al. (2011). Unbiased online active learning in data streams. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining—KDD '11* (p. 195). https://doi.org/10.1145/2020408.2020444.

Citovsky, G., DeSalvo, G., Gentile, C., et al. (2021). Batch active learning at scale. In *35th Conference on neural information processing systems, NeurIPS 2021*. arXiv:2107.14263.

Cohn, D. A., Ghahramani, Z., & Jordan, M. I. (1996). Active learning with statistical models. *Journal of Artificial Intelligence Research, 4*, 129–145. https://doi.org/10.1613/jair.295

Crammer, K., Dekel, O., Keshet, J., et al. (2006). Online passive-aggressive algorithms. *The Journal of Machine Learning Research*. https://jmlr.csail.mit.edu/papers/volume7/crammer06a/crammer06a.pdf.

Dasgupta, S., Kalai, A. T., & Monteleoni, C. (2005). Analysis of perceptron-based active learning. In *COLT '05—international conference on computational learning theory* (pp. 249–263). https://doi.org/10.1007/11503415_17.

Desalvo, G., Gentile, C., & Thune, T. S. (2021). Online active learning with surrogate loss functions. In *Advances in neural information processing systems 34 (NeurIPS 2021)*. https://proceedings.neurips.cc/paper/2021/hash/c1619d2ad66f7629c12c87fe21d32a58-Abstract.html.

Donmez, P., Carbonell, J., & Bennet, P. (2007). Dual strategy active learning. In *18th European conference on machine learning, ECML 2007*, 4701. https://doi.org/10.1007/978-3-540-74958-5_14.

Duchi, J. C., Jordan, M. I., & Wainwright, M. J. (2013). Local privacy and statistical minimax rates. In *2013 IEEE 54th annual symposium on foundations of computer science* (pp. 429–438). https://doi.org/10.1109/FOCS.2013.53.

Ebbinghaus, H. (2013). Memory: A contribution to experimental psychology. *Annals of Neurosciences*. https://doi.org/10.5214/ans.0972.7531.200408

Fang, M., Li, Y., & Cohn, T. (2017). Learning how to active learn: A deep reinforcement learning approach. arXiv:1708.02383.

Ferdowsi, Z., Ghani, R., & Settimi, R. (2013). Online active learning with imbalanced classes. In *2013 IEEE 13th international conference on data mining* (pp. 1043–1048). https://doi.org/10.1109/ICDM.2013.12.

Feuz, K. D., & Cook, D. J. (2013). Real-time annotation tool (rat). In *Workshops at the twenty-seventh AAAI conference on artificial intelligence*.

Fiez, T., Jain, L., Jamieson, K., et al. (2019). Sequential experimental design for transductive linear bandits. In *33rd conference on neural information processing systems (NeurIPS 2019)*. https://proceedings.neurips.cc/paper_files/paper/2019/file/8ba6c657b03fc7c8dd4dff8e45defcd2-Paper.pdf.

Filippi, S., Cappe, O., Garivier, A., et al. (2010). Parametric bandits: The generalized linear case. In *Advances in neural information processing systems 23 (NIPS 2010)*. https://papers.nips.cc/paper_files/paper/2010/hash/c2626d850c80ea07e7511bbae4c76f4b-Abstract.html.

Fontaine, X., Perrault, P., Valko, M., et al. (2021). Online a-optimal design and active linear regression. http://proceedings.mlr.press/v139/fontaine21a/fontaine21a.pdf.

Fortuna, L., Graziani, S., Rizzo, A., et al. (2007). *Soft sensors for monitoring and control of industrial processes* (Vol. 22). Berlin: Springer. https://doi.org/10.1007/978-1-84628-480-9

Fowler, K., Kokilepersaud, K., Prabhushankar, M., et al. (2023). Clinical trial active learning. In *The 14th ACM conference on bioinformatics, computational biology and health informatics* (ACM-BCB).

Freeman, P. R. (1983). The secretary problem and its extensions: A review. *International Statistical Review, 51*, 189–206.

Freund, Y., Seung, H. S., Shamir, E., et al. (1997). Selective sampling using the query by committee algorithm. *Machine Learning, 28*, 133–168. https://doi.org/10.1023/a:1007330508534

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of *m* rankings. *The Annals of Mathematical Statistics, 11*, 86–92. https://doi.org/10.1214/aoms/1177731944

Frumosu, F. D., & Kulahci, M. (2018). Big data analytics using semi-supervised learning methods. *Quality and Reliability Engineering International, 34*, 1413–1423. https://doi.org/10.1002/qre.2338

Fu, Y., Zhu, X., & Li, B. (2013). A survey on instance selection for active learning. *Knowledge and Information Systems, 35*, 249–283. https://doi.org/10.1007/s10115-012-0507-8

Fujii, K., & Kashima, H. (2016). Budgeted stream-based active learning via adaptive submodular maximization. In *30th annual conference on neural information processing systems, NIPS 2016*. https://proceedings.neurips.cc/paper/2016/hash/07cdfd23373b17c6b337251c22b7ea57-Abstract.html.

Gajjar, S., Kulahci, M., & Palazoglu, A. (2018). Real-time fault detection and diagnosis using sparse principal component analysis. *Journal of Process Control, 67*, 112–128. https://doi.org/10.1016/j.jprocont.2017.03.005

Galvanin, F. (2010). Optimal model-based design of experiments in dynamic systems: Novel techniques and unconventional applications. Thesis. https://hdl.handle.net/11577/3427095.

Gama, J., Medas, P., Castillo, G., et al. (2004). Learning with drift detection. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 3171*, 286–295. https://doi.org/10.1007/978-3-540-28645-5_29

Gama, J., Sebastiao, R., & Rodrigues, P. P. (2009). Issues in evaluation of stream learning algorithms. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 329–338).

Gama, J., Sebastiao, R., & Rodrigues, P. P. (2013). On evaluating stream learning algorithms. *Machine Learning, 90*, 317–346.

Garivier, A., & Moulines, E. (2008). On upper-confidence bound policies for non-stationary bandit problems. arXiv:0805.3415.

Ge, D., & Zeng, X. J. (2020). Learning data streams online—An evolving fuzzy system approach with self-learning/adaptive thresholds. *Information Sciences, 507*, 172–184. https://doi.org/10.1016/j.ins.2019.08.036

Ge, Z. (2014). Active learning strategy for smart soft sensor development under a small number of labeled data samples. *Journal of Process Control, 24*, 1454–1461. https://doi.org/10.1016/j.jprocont.2014.06.015

Gemaque, R. N., Costa, A. F. J., Giusti, R., et al. (2020). An overview of unsupervised drift detection methods. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*. https://doi.org/10.1002/widm.1381

Ghassemi, M., Sarwate, A. D., & Wright, R. N. (2016). Differentially private online active learning with applications to anomaly detection. In *AISec 2016—Proceedings of the 2016 ACM workshop on artificial intelligence and security, co-located with CCS 2016* (pp. 117–128). https://doi.org/10.1145/2996758.2996766.

Ghiasi, S., Pazzi, G., Del Grosso, C., et al. (2023). Combining thermodynamics-based model of the centrifugal compressors and active machine learning for enhanced industrial design optimization. In *1st workshop on the synergy of scientific and machine learning modeling@ ICML2023*.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., et al. (2014). Generative adversarial networks. arXiv:1406.2661.

Gouk, H., Pfahringer, B., & Frank, E. (2019). Stochastic gradient trees. http://proceedings.mlr.press/v101/gouk19a/gouk19a.pdf.

Gu, X., Han, J., Shen, Q., et al. (2022). Autonomous learning for fuzzy systems: A review. *Artificial Intelligence Review*. https://doi.org/10.1007/s10462-022-10355-6

Gu, X., Han, J., Shen, Q., et al. (2023). Autonomous learning for fuzzy systems: A review. *Artificial Intelligence Review, 56*(8), 7549–7595.

Halder, B., Hasan, K. A., Amagasa, T., et al. (2023). Autonomic active learning strategy using cluster-based ensemble classifier for concept drifts in imbalanced data stream. *Expert Systems with Applications*, 120578.

Hanneke, S. (2014). Theory of disagreement-based active learning. *Foundations and Trends in Machine Learning, 7*, 131–309. https://doi.org/10.1561/2200000037

Hanneke, S., & Yang, L. (2021). Toward a general theory of online selective sampling: Trading off mistakes and queries. In *proceedings of the 24th international conference on artificial intelligence and statistics*. https://proceedings.mlr.press/v130/hanneke21a.html.

Hao, S., Hu, P., Zhao, P., et al. (2018). Online active learning with expert advice. *ACM Transactions on Knowledge Discovery from Data*. https://doi.org/10.1145/3201604

Hao, S., Lu, J., Zhao, P., et al. (2018). Second-order online active learning and its applications. *IEEE Transactions on Knowledge and Data Engineering, 30*, 1338–1351. https://doi.org/10.1109/TKDE.2017.2778097

Haussmann, E., Fenzi, M., Chitta, K., et al. (2020). Scalable active learning for object detection. In *Proceedings 31st IEEE intelligent vehicles symposium (IV)*. https://doi.org/10.1109/IV47402.2020.9304793.

He, K., Zhang, X., Ren, S., et al. (2015). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR.2016.90

Hoang, T. N., Hong, S., Xiao, C., et al. (2021). Aid: Active distillation machine to leverage pre-trained black-box models in private data settings. *Proceedings of the Web Conference, 2021*, 3569–3581. https://doi.org/10.1145/3442381.3449944

Hodges, J., & Lehmann, E. (1962). Rank methods for combination of independent experiments in analysis of variance. *The Annals of Mathematical Statistics, 33*, 482–497.

Hoffmann, H. (2007). Kernel PCA for novelty detection. *Pattern Recognition, 40*, 863–874. https://doi.org/10.1016/j.patcog.2006.07.009

Hoi, S. C., Sahoo, D., Lu, J., et al. (2021). Online learning: A comprehensive survey. *Neurocomputing, 459*, 249–289. https://doi.org/10.1016/j.neucom.2021.04.112

Hoi, S. C. H., Jin, R., Zhao, P., et al. (2013). Online multiple kernel classification. *Machine Learning, 90*, 289–316. https://doi.org/10.1007/s10994-012-5319-2

Houlsby, N., Hernandez-Lobato, J. M., & Ghahramani, Z. (2014). Cold-start active learning with robust ordinal matrix factorization. In *31st international conference on machine learning*. https://proceedings.mlr.press/v32/houlsby14.html.

Hua, J., Xiong, Z., Lowey, J., et al. (2005). Optimal number of features as a function of sample size for various classification rules. *Bioinformatics, 21*, 1509–1515. https://doi.org/10.1093/bioinformatics/bti171

Huang, B., Salgia, S., & Zhao, Q. (2022). Disagreement-based active learning in online settings. *IEEE Transactions on Signal Processing, 70*, 1947–1958. https://doi.org/10.1109/TSP.2022.3159388

Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing, 70*, 489–501. https://doi.org/10.1016/j.neucom.2005.12.126

Huang, S. J., Jin, R., & Zhou, Z. H. (2014). Active learning by querying informative and representative examples. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 36*, 1936–1949. https://doi.org/10.1109/TPAMI.2014.2307881

Hulten, G., Spencer, L., & Domingos, P. (2001). Mining time-changing data streams. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining—KDD '01* (pp. 97–106). https://doi.org/10.1145/502512.502529.

Ienco, D., Bifet, A., Zliobaite, et al. (2013). Clustering based active learning for evolving data streams. In *16th international conference on discovery science*. https://doi.org/10.1007/978-3-642-40897-7_6.

Ienco, D., Pfahringer, B., & Žliobaitė, I. (2014). High density-focused uncertainty sampling for active learning over evolving stream data. In *BIGMINE'14: Proceedings of the 3rd international conference on big data, streams and heterogeneous source mining: Algorithms, systems, programming models and applications*. https://proceedings.mlr.press/v36/ienco14.html.

Iman, R. L., & Davenport, J. M. (1980). Approximations of the critical region of the fbietkan statistic. *Communications in Statistics - Theory and Methods, 9*, 571–595. https://doi.org/10.1080/03610928008827904

Istrate, R., Malossi, A. C. I., Bekas, C., et al. (2018). Incremental training of deep convolutional neural networks. arXiv:1803.10232.

Jamieson, K. (2018). Online and adaptive machine learning. *Regression (Part 7)*. https://courses.cs.washington.edu/courses/cse599i/18wi/.

Jamieson, K., & Nowak, R. (2014). Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting. In *2014 48th annual conference on information sciences and systems (CISS)* (pp. 1–6). https://doi.org/10.1109/CISS.2014.6814096.

Jamil, S., & Khan, A. (2016). Churn comprehension analysis for telecommunication industry using alba. In *2016 international conference on emerging technologies (ICET)* (pp. 1–5). IEEE.

Jedra, Y., & Proutiere, A. (2020). Optimal best-arm identification in linear bandits. In *34th conference on neural information processing systems (NeurIPS 2020)*. https://proceedings.neurips.cc/paper/2020/file/7212a6567c8a6c513f33b858d868ff80-Paper.pdf.

Jin, Q., Yuan, M., Li, S., et al. (2022). Cold-start active learning for image classification. *Information Sciences, 616*, 16–36. https://doi.org/10.1016/j.ins.2022.10.066

Jin, R., Hoi, S., & Yang, T. (2010). Online multiple kernel learning: Algorithms and mistake bounds. In *Proceedings of the 21st international conference on algorithmic learning theory*. https://doi.org/10.1007/978-3-642-16108-7_31.

John, R. C. S., & Draper, N. R. (1975). D-optimality for regression designs: A review. *Technometrics, 17*, 15–23. https://doi.org/10.1080/00401706.1975.10489266

Joshi, A. J., Porikli, F., & Papanikolopoulos, N. (2009). Multi-class active learning for image classification. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 2372–2379). https://doi.org/10.1109/CVPR.2009.5206627.

Joyce, J. M. (2011). Kullback–Leibler divergence. https://doi.org/10.1007/978-3-642-04898-2_327.

Karlin, S., & Studden, W. J. (1966). Optimal experimental designs. *The Annals of Mathematical Statistics, 37*, 783–815.

Kiefer, J. (1959). Optimum experimental designs. *Journal of the Royal Statistical Society Series B (Methodological)*. https://www.jstor.org/stable/2983802.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational Bayes. In *2nd international conference on learning representations, ICLR*. arXiv:1312.6114.

Kottke, D., Krempl, G., & Spiliopoulou, M. (2015). *Probabilistic active learning in datastreams*. https://doi.org/10.1007/978-3-319-24465-5_13.

Kranjc, J., Smailović, J., Podpečan, V., et al. (2015). Active learning for sentiment analysis on data streams: Methodology and workflow implementation in the clowdflows platform. *Information Processing & Management, 51*(2), 187–203.

Kraskov, A., Stögbauer, H., & Grassberger, P. (2004). Estimating mutual information. *Physical Review E, 69*, 066138. https://doi.org/10.1103/PhysRevE.69.066138

Krawczyk, B., Minku, L. L., Gama, J., et al. (2017). Ensemble learning for data stream analysis: A survey. *Information Fusion, 37*, 132–156.

Krawczyk, B., Pfahringer, B., & Wozniak, M. (2018). Combining active learning with concept drift detection for data stream mining. In *2018 IEEE international conference on big data (big data)* (pp. 2239–2244). https://doi.org/10.1109/BigData.2018.8622549.

Škrjanc, I. (2009). Confidence interval of fuzzy models: An example using a waste-water treatment plant. *Chemometrics and Intelligent Laboratory Systems, 96*, 182–187. https://doi.org/10.1016/j.chemolab.2009.01.009

Kulkarni, R. V., Patil, S. H., & Subhashini, R. (2016). An overview of learning in data streams with label scarcity. In *Proceedings of the international conference on inventive computation technologies, ICICT, 2016* (Vol. 2). https://doi.org/10.1109/INVENTIVE.2016.7824874.

Kumar, P., & Gupta, A. (2020). Active learning query strategies for classification, regression, and clustering: A survey. *Journal of Computer Science and Technology, 35*, 913–945. https://doi.org/10.1007/s11390-020-9487-4

Kurlej, B., & Woźniak, M. (2011). *Learning curve in concept drift while using active learning paradigm*. https://doi.org/10.1007/978-3-642-23857-4_13.

Kwak, B., Kim, Y., & Kim, Y. J., et al. (2022). Trustal: Trustworthy active learning using knowledge distillation. In *The thirty-sixth AAAI conference on artificial intelligence (AAAI-22)*. arXiv:2201.11661.

Lakshminarayanan, B., Roy, D., & Teh, Y. W. (2014). Mondrian forests: Efficient online random forests. In *Advances in neural information processing systems (NIPS)*. https://proceedings.neurips.cc/paper_files/paper/2014/file/d1dc3a8270a6f9394f88847d7f0050cf-Paper.pdf.

Li, A., Boyd, A., Smyth, P., et al. (2021). Detecting and adapting to irregular distribution shifts in Bayesian online learning. In *35th conference on neural information processing systems (NeurIPS 2021)*. https://papers.nips.cc/paper/2021/file/362387494f6be6613daea643a7706a42-Paper.pdf.

Li, X., & Guo, Y. (2013). Adaptive active learning for image classification. In *2013 IEEE conference on computer vision and pattern recognition* (pp. 859–866). https://doi.org/10.1109/CVPR.2013.116.

Lieber, D., Konrad, B., Deuse, J., et al. (2012). Sustainable interlinked manufacturing processes through real-time quality prediction. In *Leveraging technology for a sustainable world: Proceedings of the 19th CIRP conference on life cycle engineering*, University of California at Berkeley, Berkeley, USA, May 23–25, 2012 (pp. 393–398). Springer.

Lima, M., Neto, M., Filho, T. S., et al. (2022). Learning under concept drift for regression-a systematic literature review. *IEEE Access, 10*, 45410–45429. https://doi.org/10.1109/ACCESS.2022.3169785

Liu, S., Xue, S., Wu, J., et al. (2021). Online active learning for drifting data streams. *IEEE Transactions on Neural Networks and Learning Systems*. https://doi.org/10.1109/TNNLS.2021.3091681

Long, J., Yin, J., Zhao, W., et al. (2008). Graph-based active learning based on label propagation. In *MDAI 2008: Modeling decisions for artificial intelligence* (pp. 179–190). https://doi.org/10.1007/978-3-540-88269-5_17.

Loy, C. C., Hospedales, T. M., Xiang, T., et al. (2012). Stream-based joint exploration-exploitation active learning. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition* (pp. 1560–1567). https://doi.org/10.1109/CVPR.2012.6247847.

Lu, J., Zhao, P., & Hoi, S. C. H. (2016). Online passive-aggressive active learning. *Machine Learning, 103*, 141–183. https://doi.org/10.1007/s10994-016-5555-y

Lu, J., Liu, A., Dong, F., et al. (2018). Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*. https://doi.org/10.1109/TKDE.2018.2876857

Lughofer, E. (2011). *Evolving fuzzy systems—Methodologies, advanced concepts and applications* (Vol. 266). Berlin: Springer. https://doi.org/10.1007/978-3-642-18087-3

Lughofer, E. (2012). Single-pass active learning with conflict and ignorance. *Evolving Systems, 3*, 251–271. https://doi.org/10.1007/s12530-012-9060-7

Lughofer, E. (2017). On-line active learning: A new paradigm to improve practical useability of data stream modeling methods. *Information Sciences, 415–416*, 356–376. https://doi.org/10.1016/j.ins.2017.06.038

Lughofer, E., & Pratama, M. (2018). Online active learning in data stream regression using uncertainty sampling based on evolving generalized fuzzy models. *IEEE Transactions on Fuzzy Systems, 26*, 292–309. https://doi.org/10.1109/TFUZZ.2017.2654504

Lughofer, E., & Škrjanc, I. (2023). Online active learning for evolving error feedback fuzzy models within a multi-innovation context. *IEEE Transactions on Fuzzy Systems*. https://doi.org/10.1109/TFUZZ.2023.3302403

Ma, L., Destercke, S., & Wang, Y. (2016). Online active learning of decision trees with evidential data. *Pattern Recognition, 52*, 33–45. https://doi.org/10.1016/j.patcog.2015.10.014

Mammen, E., & Tsybakov, A. B. (1999). Smooth discrimination analysis. *The Annals of Statistics*. https://doi.org/10.1214/aos/1017939240

Manjah, D., Cacciarelli, D., Standaert, B., et al. (2023). Stream-based active distillation for scalable model deployment. In *Proceedings of the IEEE/CVF computer vision and pattern recognition (CVPR) workshops*.

Manwani, N., Desai, K., Sasidharan, S., et al. (2013). Double ramp loss based reject option classifier. In *19th Pacific-Asia conference on advances in knowledge discovery and data mining (PAKDD)*. https://doi.org/10.1007/978-3-319-57454-7_53.

Martins, V. E., Cano, A., & Junior, S. B. (2023). Meta-learning for dynamic tuning of active learning on stream classification. *Pattern Recognition, 138*, 109359.

McSherry, F., & Talwar, K. (2007). Mechanism design via differential privacy. In *48th annual IEEE symposium on foundations of computer science (FOCS'07)* (pp. 94–103). https://doi.org/10.1109/FOCS.2007.41.

Menard, P., Domingues, O. D., Jonsson, A., et al. (2021). Fast active learning for pure exploration in reinforcement learning. In *Proceedings of the 38th international conference on machine learning*. http://proceedings.mlr.press/v139/menard21a/menard21a-supp.pdf.

Min, F., Zhang, S. M., Ciucci, D., et al. (2020). Three-way active learning through clustering selection. *International Journal of Machine Learning and Cybernetics, 11*, 1033–1046. https://doi.org/10.1007/s13042-020-01099-2

Minka, T. P. (2001). A family of algorithms for approximate Bayesian inference. Thesis. https://hd.media.mit.edu/tech-reports/TR-533.pdf.

Miu, T., Missier, P., & Plötz, T. (2015). Bootstrapping personalised human activity recognition models using online active learning. *2015 IEEE international conference on computer and information technology; ubiquitous computing and communications; dependable, autonomic and secure computing; pervasive intelligence and computing* (pp. 1138–1147). IEEE.

Mohamad, S., Bouchachia, A., & Sayed-Mouchaweh, M. (2018). A bi-criteria active learning algorithm for dynamic data streams. *IEEE Transactions on Neural Networks and Learning Systems, 29*, 74–86. https://doi.org/10.1109/TNNLS.2016.2614393

Mohamad, S., Sayed-Mouchaweh, M., & Bouchachia, A. (2020). Online active learning for human activity recognition from sensory data streams. *Neurocomputing, 390*, 341–358. https://doi.org/10.1016/j.neucom.2019.08.092

Mohamadi, S., & Amindavar, H. (2020). Deep Bayesian active learning, a brief survey on recent advances. arXiv:2012.08044.

Montgomery, D. C. (2012). *Design and analysis of experiments*. New York: Wiley. https://doi.org/10.1002/9781118147634

Myers, R. H., Montgomery, D., & Anderson-Cook, C. M. (2016). Response surface methodology: Process and product optimization using designed experiments. Wiley. https://www.wiley.com/en-au/Response+Surface+Methodology:+Process+and+Product+Optimization+Using+Designed+Experiments,+4th+Edition-p-9781118916018.

Naranjo, J. E., Sotelo, M. A., Gonzalez, C., et al. (2007). Using fuzzy logic in automated vehicle control. *IEEE Intelligent Systems, 22*(1), 36–45.

Narr, A., Triebel, R., & Cremers, D. (2016). Stream-based active learning for efficient and adaptive classification of 3d objects. In *Proceedings—IEEE international conference on robotics and automation 2016-June* (pp. 227–233). https://doi.org/10.1109/ICRA.2016.7487138.

Nguyen, H. T., & Smeulders, A. (2004). Active learning using pre-clustering. In *Proceedings of the twenty-first international conference on machine learning*. https://doi.org/10.1145/1015330.1015349.

Nixon, C., Sedky, M., & Hassan, M. (2021). Reviews in online data stream and active learning for cyber intrusion detection-a systematic literature review. In *2021 Sixth international conference on fog and mobile edge computing (FMEC)* (pp. 1–6). IEEE.

Pham, T., Kottke, D., Krempl, G., et al. (2022). Stream-based active learning for sliding windows under the influence of verification latency. *Machine Learning, 111*, 2011–2036. https://doi.org/10.1007/s10994-021-06099-z

Pitman, J., & Yor, M. (1997). The two-parameter Poisson–Dirichlet distribution derived from a stable subordinator. *The Annals of Probability, 25*. https://www.jstor.org/stable/20680193.

Polikar, R., Upda, L., Upda, S., et al. (2001). Learn++: An incremental learning algorithm for supervised neural networks. *IEEE Transactions on Systems, Man and Cybernetics. Part C (Applications and Reviews), 31*, 497–508. https://doi.org/10.1109/5326.983933

Prabhu, V., Chandrasekaran, A., Saenko, K., et al. (2020). Active domain adaptation via clustering uncertainty-weighted embeddings. https://github.com/virajprabhu/CLUE.

Pratama, M., Anavatti, S. G., & Lu, J. (2015). Recurrent classifier based on an incremental metacognitive-based scaffolding algorithm. *IEEE Transactions on Fuzzy Systems, 23*, 2048–2066. https://doi.org/10.1109/TFUZZ.2015.2402683

Qin, J., Wang, C., Zou, Q., et al. (2021). Active learning with extreme learning machine for online imbalanced multiclass classification. *Knowledge-Based Systems, 231*, 107385. https://doi.org/10.1016/j.knosys.2021.107385

Quade, D. (1979). Using weighted rankings in the analysis of complete blocks with additive block effects. *Journal of the American Statistical Association, 74*, 680. https://doi.org/10.2307/2286991

Réda, C., Kaufmann, E., & Delahaye-Duriez, A. (2020). Machine learning applications in drug development. *Computational and Structural Biotechnology Journal, 18*, 241–252.

Ren, P., Xiao, Y., Chang, X., et al. (2022). A survey of deep active learning. *ACM Computing Surveys, 54*, 1–40. https://doi.org/10.1145/3472291

Reyes, O., Altalhi, A. H., & Ventura, S. (2018). Statistical comparisons of active learning strategies over multiple datasets. *Knowledge-Based Systems, 145*, 274–288. https://doi.org/10.1016/j.knosys.2018.01.033

Riis, C., Antunes, F., Hüttel, F. B., et al. (2022). Bayesian active learning with fully Bayesian Gaussian processes. In *Proceedings of advances in neural information processing systems 35 (NeurIPS 2022)*. https://proceedings.neurips.cc/paper_files/paper/2022/file/4f1fba885f266d87653900fd3045e8af-Paper-Conference.pdf.

Riquelme, C. (2017). Online active learning with linear models. Thesis. http://purl.stanford.edu/rp382fv8012.

Riquelme, C., Ghavamzadeh, M., & Lazaric, A. (2017a). Active learning for accurate estimation of linear models. In *Proceedings of the 34th international conference on machine learning*. http://proceedings.mlr.press/v70/riquelme17a/riquelme17a.pdf.

Riquelme, C., Johari, R., & Zhang, B. (2017b). Online active linear regression via thresholding. In *Thirty-first AAAI conference on artificial intelligence*. www.aaai.org.

Rožanec, J. M., Trajkova, E., Dam, P., et al. (2022). Streaming machine learning and online active learning for automated visual inspection. *IFAC-PapersOnLine, 55*, 277–282. https://doi.org/10.1016/j.ifacol.2022.04.206

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review, 65*, 386–408. https://doi.org/10.1037/h0042519

Roth, D., & Small, K. (2006). Margin-based active learning for structured output spaces. *Machine Learning: ECML, 2006*, 413–424. https://doi.org/10.1007/11871842_40

Roy, N., & Mccallum, A. (2001). Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the eighteenth international conference on machine learning*. https://dl.acm.org/doi/10.5555/645530.655646.

Ruan, Y., Yang, J., & Zhou, Y. (2020). Linear bandits with limited adaptivity and learning distributional optimal design. In *STOC 2021: Proceedings of the 53rd annual ACM SIGACT symposium on theory of computing*. https://doi.org/10.1145/3406325.3451004.

Rudovic, O., Zhang, M., Schuller, B., et al. (2019). Multi-modal active learning from human data: A deep reinforcement learning approach. In *2019 international conference on multimodal interaction* (pp. 6–15). https://doi.org/10.1145/3340555.3353742.

Saran, A., Yousefi, S., Krishnamurthy, A., et al. (2023). Streaming active learning with deep neural networks. In Krause, A., Brunskill, E., Cho, K., et al. (Eds.), *Proceedings of the 40th international conference on machine learning, proceedings of machine learning research. PMLR* (Vol. 202, pp. 30005–30021). https://proceedings.mlr.press/v202/saran23a.html.

Schmidt, S., Rao, Q., Tatsch, J., et al. (2020). Advanced active learning strategies for object detection. In *2020 IEEE intelligent vehicles symposium (IV)* (pp. 871–876). https://doi.org/10.1109/IV47402.2020.9304565.

Schmitt, R., Jatzkowski, P., & Peterek, M. (2013). Traceable measurements using machine tools. In *Laser metrology and machine performance X: 10th international conference and exhibition on laser metrology, machine tool, CMM & robotic performance, Lamdamap* (pp. 20–21).

Sculley, D. (2007). Online active learning methods for fast label efficient spam filtering. In *Proceedings of the fourth conference on email and antispam*.

Sener, O., & Savarese, S. (2017). Active learning for convolutional neural networks: A core-set approach. In *ICLR*.

Settles, B. (2009). *Active learning literature survey*. Technical report 1648, University of Wisconsin-Madison Department of Computer Sciences. https://burrsettles.com/pub/settles.activelearning.pdf.

Seung, H. S., Opper, M., & Sompolinsky, H. (1992). Query by committee. In *Proceedings of the fifth annual workshop on computational learning theory—COLT '92* (pp. 287–294). https://doi.org/10.1145/130385.130417.

Shah, K., & Manwani, N. (2020). Online active learning of reject option classifiers. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 34, pp. 5652–5659). https://doi.org/10.1609/aaai.v34i04.6019.

Shan, J., Zhang, H., Liu, W., et al. (2019). Online active learning ensemble framework for drifted data streams. *IEEE Transactions on Neural Networks and Learning Systems, 30*, 486–498. https://doi.org/10.1109/TNNLS.2018.2844332

Shannon, E. (1948). A mathematical theory of communication. *The Bell System Technical Journal, 27*, 379–423.

Sheng, V. S., Provost, F., & Ipeirotis, P. G. (2008). Get another label? Improving data quality and data mining using multiple, noisy labelers. In *Proceeding of the 14th ACM SIGKDD international conference on knowledge discovery and data mining—KDD 08* (p. 614). https://doi.org/10.1145/1401890.1401965.

Shi, X., & Xiong, W. (2018). Approximate linear dependence criteria with active learning for smart soft sensor design. *Chemometrics and Intelligent Laboratory Systems, 180*, 88–95. https://doi.org/10.1016/j.chemolab.2018.07.009

Shilton, A., Palaniswami, M., Ralph, D., et al. (2005). Incremental training of support vector machines. *IEEE Transactions on Neural Networks, 16*, 114–131. https://doi.org/10.1109/TNN.2004.836201

Soare, M., Lazaric, A., & Munos, R. (2013). Active learning in linear stochastic bandits. Bayesian Optimization in Theory and Practice https://www.univ-orleans.fr/lifo/Members/soare/files/active_learning_linear_bandit.pdf.

Soare, M., Lazaric, A., & Munos, R. (2014). Best-arm identification in linear bandits. In *27th Conference on neural information processing systems (NeurIPS 2014)*.

Song, S., Chaudhuri, K., & Sarwate, A. D. (2013). Stochastic gradient descent with differentially private updates. In *2013 IEEE global conference on signal and information processing* (pp. 245–248). https://doi.org/10.1109/GlobalSIP.2013.6736861.

Souza, V., Pinho, T., & Batista, G. (2018). Evaluating stream classifiers with delayed labels information. In *2018 7th Brazilian conference on intelligent systems (BRACIS)* (pp. 408–413). https://doi.org/10.1109/BRACIS.2018.00077.

Steel, R. G. D. (1959). A multiple comparison sign test: Treatments versus control. *Journal of the American Statistical Association, 54*, 767. https://doi.org/10.2307/2282500

Steve, H., & Liu, Y. (2014). Minimax analysis of active learning. *Journal of Machine Learning Research*. https://www.jmlr.org/papers/volume16/hanneke15a/hanneke15a.pdf.

Subramanian, K., Das, A. K., Sundaram, S., et al. (2014). A meta-cognitive interval type-2 fuzzy inference system and its projection based learning algorithm. *Evolving Systems, 5*, 219–230. https://doi.org/10.1007/s12530-013-9102-9

Sudarsanam, N., & Ravindran, B. (2018). Using linear stochastic bandits to extend traditional offline designed experiments to online settings. *Computers & Industrial Engineering, 115*, 471–485.

Suresh, S., Sundararajan, N., & Saratchandran, P. (2008). Risk-sensitive loss functions for sparse multi-category classification problems. *Information Sciences, 178*, 2621–2638. https://doi.org/10.1016/j.ins.2008.02.009

Suárez-Cetrulo, A. L., Kumar, A., & Miralles-Pechuán, L. (2021). Modelling the covid-19 virus evolution with incremental machine learning. In *29th Irish conference on artificial intelligence and cognitive science, AICS 2021*. https://ceur-ws.org/Vol-3105/paper1.pdf.

Suárez-Cetrulo, A. L., Quintana, D., & Cervantes, A. (2023). A survey on machine learning for recurring concept drifting data streams. *Expert Systems with Applications, 213*, 118934. https://doi.org/10.1016/j.eswa.2022.118934

Suzuki, K., Sunagawa, T., Sasaki, T., et al. (2021). Annotation cost reduction of stream-based active learning by automated weak labeling using a robot arm. In *2021 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 9000–9007). https://doi.org/10.1109/IROS51168.2021.9636355.

Tang, Q., Li, D., & Xi, Y. (2018). A new active learning strategy for soft sensor modeling based on feature reconstruction and uncertainty evaluation. *Chemometrics and Intelligent Laboratory Systems, 172*, 43–51. https://doi.org/10.1016/j.chemolab.2017.11.001

Taylor, G., & Hinton, G. (2009). Factored conditional restricted Boltzmann machines for modeling motion style. In *Proceedings of the 26th international conference on machine learning, Montreal, Canada, 2009*. https://doi.org/10.1145/1553374.1553505.

Taylor, G., Hinton, G., & Roweis, S. (2006). Modeling human motion using binary latent variables. In *Advances in neural information processing systems 19 (NIPS 2006)*. https://papers.nips.cc/paper_files/paper/2006/hash/1091660f3dff84fd648efe31391c5524-Abstract.html.

Thompson, J., Walters, W. P., Feng, J. A., et al. (2022). Optimizing active learning for free energy calculations. *Artificial Intelligence in the Life Sciences, 2*, 100050. https://doi.org/10.1016/j.ailsci.2022.100050

Tieppo, E., dos Santos, R. R., Barddal, J. P., et al. (2022). Hierarchical classification of data streams: A systematic literature review. *Artificial Intelligence Review, 55*, 3243–3282. https://doi.org/10.1007/s10462-021-10087-z

Tong, S., & Koller, D. (2002). Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*. https://doi.org/10.1162/153244302760185243

Tran, T., Pham, T., Carneiro, G., et al. (2017). A Bayesian data augmentation approach for learning deep models. In *31st conference on neural information processing systems (NIPS 2017)*. https://proceedings.neurips.cc/paper_files/paper/2017/file/076023edc9187cf1ac1f1163470e479a-Paper.pdf.

Tran, T., Do, T. T., Reid, I., et al. (2019). Bayesian generative active deep learning. In *Proceedings of the 36th international conference on machine learning*. arXiv:1904.11643.

Tsybakov, A. B. (2004). Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*. https://doi.org/10.1214/aos/1079120131

Tsymbal, A., Pechenizkiy, M., Cunningham, P., et al. (2008). Dynamic integration of classifiers for handling concept drift. *Information Fusion, 9*, 56–68. https://doi.org/10.1016/j.inffus.2006.11.002

Vahdat, A., Belbahri, M., & Nia, V. P. (2019). Active learning for high-dimensional binary features. In *15th international conference on network and service management (CNSM)*. https://www.computer.org/csdl/proceedings-article/cnsm/2019/09012676/1hQr3hscsJG.

Vanhatalo, E., & Kulahci, M. (2016). Impact of autocorrelation on principal components and their use in statistical process control. *Quality and Reliability Engineering International, 32*, 1483–1500. https://doi.org/10.1002/qre.1858

Vanhatalo, E., Kulahci, M., & Bergquist, B. (2017). On the structure of dynamic principal component analysis used in statistical process monitoring. *Chemometrics and Intelligent Laboratory Systems, 167*, 1–11. https://doi.org/10.1016/j.chemolab.2017.05.016

Wang, L. (2011). Smoothness, disagreement coefficient, and the label complexity of agnostic active learning. *The Journal of Machine Learning Research*. https://www.jmlr.org/papers/volume12/wang11b/wang11b.pdf.

Wang, X., Fu, M., Ma, H., et al. (2015). Lateral control of autonomous vehicles based on fuzzy logic. *Control Engineering Practice, 34*, 1–17.

Wassermann, S., Cuvelier, T., & Casas, P. (2019). Ral-improving stream-based active learning by reinforcement learning. https://hal.archives-ouvertes.fr/hal-02265426.

Weigl, E., Heidl, W., Lughofer, E., et al. (2016). On improving performance of surface inspection systems by online active learning and flexible classifier updates. *Machine Vision and Applications, 27*, 103–127. https://doi.org/10.1007/s00138-015-0731-9

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin, 1*, 80. https://doi.org/10.2307/3001968

Woodward, M., & Finn, C. (2017). Active one-shot learning. In *NIPS 2016, deep reinforcement learning workshop*. arXiv:1702.06559.

Woźniak, M., Zyblewski, P., & Ksieniewicz, P. (2023). Active weighted aging ensemble for drifted data stream classification. *Information Sciences, 630*, 286–304.

Wu, J., Chen, J., & Huang, D. (2022). Entropy-based active learning for object detection with progressive diversity constraint. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*. https://doi.org/10.1109/CVPR52688.2022.00918.

Wu, R., Guo, C., Su, Y., et al. (2021). Online adaptation to label distribution shift. In *35th conference on neural information processing systems (NeurIPS 2021)*. https://www.kaggle.com/Cornell-University/arxiv.

Wu, Y., Chen, Y., Wang, L., et al. (2019). Large scale incremental learning. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*. https://doi.org/10.1109/CVPR.2019.00046.

Xu, W., Zhao, F., & Lu, Z. (2016). Active learning over evolving data streams using paired ensemble framework. In *2016 eighth international conference on advanced computational intelligence (ICACI)* (pp. 180–185). https://doi.org/10.1109/ICACI.2016.7449823.

Yan, X., Sarkar, M., Lartey, B., et al. (2023). An online learning framework for sensor fault diagnosis analysis in autonomous cars. *IEEE Transactions on Intelligent Transportation Systems*. https://doi.org/10.1109/TITS.2023.3305620

Yin, C., Chen, S., & Yin, Z. (2023). Clustering-based active learning classification towards data stream. *ACM Transactions on Intelligent Systems and Technology, 14*(2), 1–18.

Yu, H., Sun, C., Yang, W., et al. (2015). Al-elm: One uncertainty-based active learning algorithm using extreme learning machine. *Neurocomputing, 166*, 140–150. https://doi.org/10.1016/j.neucom.2015.04.019

Yu, K., Bi, J., & Tresp, V. (2006). Active learning via transductive experimental design. In *Proceedings of the 23rd international conference on machine learning*. https://doi.org/10.1145/1143844.1143980.

Yuan, M., Lin, H. T., & Boyd-Graber, J. (2020). Cold-start active learning through self-supervised language modeling. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*. https://doi.org/10.18653/v1/2020.emnlp-main.637.

Zhang, H., Liu, W., Shan, J., et al. (2018). Online active learning paired ensemble for concept drift and class imbalance. *IEEE Access, 6*, 73815–73828. https://doi.org/10.1109/ACCESS.2018.2882872

Zhang, H., Liu, W., Sun, L., et al. (2020a). Analyzing network traffic for protocol identification: An ensemble online active learning method. In *Proceedings—2020 6th international conference on big data and information analytics, BigDIA 2020* (pp. 167–172). https://doi.org/10.1109/BigDIA51454.2020.00035.

Zhang, H., Ravi, S. S., & Davidson, I. (2020b). A graph-based approach for active learning in regression. In *Proceedings of the 2020 SIAM international conference on data mining (SDM)*. https://doi.org/10.1137/1.9781611976236.32.

Zhang, H., Liu, W., & Liu, Q. (2022). Reinforcement online active learning ensemble for drifting imbalanced data streams. *IEEE Transactions on Knowledge and Data Engineering, 34*, 3971–3983. https://doi.org/10.1109/TKDE.2020.3026196

Zhang, K., Liu, S., & Chen, Y. (2023). Online active learning framework for data stream classification with density-peaks recognition. *IEEE Access, 11*, 27853–27864.

Zhang, T. (2004). Statistical behavior and consistency of classification methods based on convex risk minimization. *The Annals of Statistics*. https://doi.org/10.1214/aos/1079120130

Zheng, Z., & Padmanabhan, B. (2006). Selectively acquiring customer information: A new data acquisition problem and an active learning-based solution. *Management Science, 52*(5), 697–712.

Zhou, C., Ma, X., Michel, P., et al. (2021). Examining and combating spurious features under distribution shift. In *Proceedings of the 38th international conference on machine learning*. https://github.com/violet-zct/.

Zhu, J. J., & Bento, J. (2017). Generative adversarial active learning. arXiv:1702.07956.

Zhu, X., Zhang, P., Lin, X., et al. (2007). Active learning from data streams. In *Proceedings—IEEE international conference on data mining, ICDM* (pp. 757–762). https://doi.org/10.1109/ICDM.2007.101.

Zliobaite, I., Bifet, A., Pfahringer, B., et al. (2014). Active learning with drifting streaming data. *IEEE Transactions on Neural Networks and Learning Systems, 25*, 27–39. https://doi.org/10.1109/TNNLS.2012.2236570

Zwanka, R. J., & Buff, C. (2021). Covid-19 generation: A conceptual framework of the consumer behavioral shifts to be caused by the covid-19 pandemic. *Journal of International Consumer Marketing, 33*, 58–67. https://doi.org/10.1080/08961530.2020.1771646

Zyblewski, P., Ksieniewicz, P., & Woźniak, M. (2020). Combination of active and random labeling strategy in the non-stationary data stream classification. In *International conference on artificial intelligence and soft computing* (pp. 576–585). Springer.

APPENDIX C

PAPER 3 – Stream-based active learning with linear models

Knowledge-Based Systems 254 (2022) 109664

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Stream-based active learning with linear models

Davide Cacciarelli [a,b,*], Murat Kulahci [a,c], John Sølve Tyssedal [b]

[a] Department of Applied Mathematics and Computer Science, Technical University of Denmark, Kgs. Lyngby, Denmark
[b] Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway
[c] Department of Business Administration, Technology and Social Sciences, Luleå University of Technology, Luleå, Sweden

ARTICLE INFO

Article history:
Received 27 May 2022
Received in revised form 11 July 2022
Accepted 7 August 2022
Available online 13 August 2022

Keywords:
Active learning
Unlabeled data
Optimal experimental design
Linear regression
Soft sensor

ABSTRACT

The proliferation of automated data collection schemes and the advances in sensorics are increasing the amount of data we are able to monitor in real-time. However, given the high annotation costs and the time required by quality inspections, data is often available in an unlabeled form. This is fostering the use of active learning for the development of soft sensors and predictive models. In production, instead of performing random inspections to obtain product information, labels are collected by evaluating the information content of the unlabeled data. Several query strategy frameworks for regression have been proposed in the literature but most of the focus has been dedicated to the static pool-based scenario. In this work, we propose a new strategy for the stream-based scenario, where instances are sequentially offered to the learner, which must instantaneously decide whether to perform the quality check to obtain the label or discard the instance. The approach is inspired by the optimal experimental design theory and the iterative aspect of the decision-making process is tackled by setting a threshold on the informativeness of the unlabeled data points. The proposed approach is evaluated using numerical simulations and the Tennessee Eastman Process simulator. The results confirm that selecting the examples suggested by the proposed algorithm allows for a faster reduction in the prediction error.

1. Introduction

The term big data seems to be ubiquitous in many fields of application, and industrial production is no different. However, in production, this can be somewhat misleading as it often refers to process data that is obtained through automated data collection schemes with minimal manual interference. Product-related data is usually scarcer particularly in high-volume manufacturing due to costs of inspection. This creates an imbalance in the amount of available data that can at times be quite substantial. Yet in many cases, predictive modeling relating process variables to product characteristics is sought after. Therefore, it will be beneficial to guide the data collection schemes for product characteristics through a real-time sampling methodology. In current production environments, sampling of the product characteristics is often performed at regular time intervals or at random. However, this approach can be ineffective as the informativeness of the observations at the time of sampling is not taken into account. This problem is reinforcing the interest of researchers and practitioners in active learning. Active learning-based sampling schemes use an instance selection criterion to strategically select data points that allow a faster reduction of the generalization error [1].

Over the last decades, many active learning approaches have been proposed, but most of the focus has been dedicated to the pool-based scenario [2]. Pool-based active learning refers to a circumstance in which a large amount of unlabeled data is collected all at once and made available to the learner, which can then select offline the data points to be labeled with a greedy approach [3].

In real-time applications for high-volume production, where samples are processed at a fast pace, evaluating all the available instances before making a choice might not be realistic. In these cases, the learner might only have a short time frame to make the sampling decision. Indeed, if a sample is not selected for the quality check, it might get lost in the downstream process and no longer be traceable. This is particularly relevant in high-volume production, where tracing individual parts is a challenge. Also, in a chemical process, we might not be able to measure the level of the variable of interest once a component undergoes a specific treatment. In these contexts, a much more sensible scenario is represented by stream-based active learning, which is sometimes referred to as selective sampling [4]. Stream-based active learning investigates a scenario where instances are processed one at a time and the learner has to determine immediately whether to keep the instance and query its label or discard it. The task is very similar to the one described by a notorious statistical riddle, the secretary problem [5], where an observer sequentially interviews

[*] Corresponding author at: Department of Applied Mathematics and Computer Science, Technical University of Denmark, Kgs. Lyngby, Denmark.
E-mail address: dcac@dtu.dk (D. Cacciarelli).

https://doi.org/10.1016/j.knosys.2022.109664

a certain number of applicants and, after each interview, a decision must be made on whether the applicant is hired or not. An exhaustive survey about stream-based active learning has been proposed by Lughofer [6], who classified existing online active learning methods by taking into account the data processing functionality, the model class (regression or classification), and many other relevant properties. The survey reveals how stream-based active learning methods have been mostly developed in the classification field. Regression models, on the other hand, are extremely useful in the development of soft sensors for hard-to-measure process variables or in quality control problems where a product's characteristic is measured on a continuous scale. That is why active learning in conjunction with regression models is capturing the interest of many researchers [7–10].

In this paper, we focus on the use of linear regression models. These models are well suited for stream-based active learning as they can easily be trained on a small number of observations, being composed of a small number of parameters. This property is also very useful if we want to efficiently retrain the model each time the design is augmented by including an additional observation [11]. Moreover, despite recent advances in terms of interpretability for deep learning models, linear regression models are still amongst the most easily interpretable models. Indeed, their parameters offer a straightforward quantitative contribution of each specific feature, and their input features are directly derived from the empirical observations [12]. Besides the direct interpretation that comes from the signs and magnitudes of the coefficients, linear models can also be used to construct confidence intervals on the parameter estimates and variable selection can also be easily incorporated into such models [13]. Recently, additional variable selection methods for linear regression models have been suggested by Zhang et al. [14]. Being able to offer a robust feature importance analysis is particularly important in industrial problems, where practitioners and engineers might need to timely intervene in specific parts or components of the process to ensure safety and operational efficiency. The simplicity of these models and the low number of parameters that require tuning is also beneficial to foster their adoption and use in applications. Finally, linear regression models allow us to build on the optimal experimental design theory and leverage the criteria that are typically used to design highly efficient experiments. Despite the focus of this paper being dedicated to linear models, nonlinear models proved to be extremely useful in a wide variety of applications. In particular, deep learning models are very effective in dealing with complex high-dimensional data to perform tasks like image recognition, shape extraction, and pose recovery [15–19].

In this work, we propose a novel strategy to perform stream-based active learning with linear models. Given the impossibility to rank observations in real-time, we provide an algorithm that only uses unlabeled data to set a threshold on the informativeness of data points. Unlabeled data is also exploited in a semi-supervised manner to increase the predictive performance [20]. We show how the proposed approach outperforms random sampling and state-of-the-art methods.

The remainder of this paper is organized as follows. In Section 2, we define some basic concepts and discuss related works focusing on active learning for regression. Section 3 introduces the proposed sampling strategy. In Section 4 we test our approach using numerical simulations; the Tennessee Eastman Process data is also used to evaluate its performance on a typical industrial process. Finally, Section 5 provides some conclusions.

2. Preliminaries

The active learning problem is defined by an imbalance between the availability of process variables $\mathbf{x} \in \mathbb{R}^p$ and the corresponding labels $y \in \mathbb{R}$. In many circumstances, industrial processes are characterized by the presence of easy-to-measure process variables, which are collected through automated collection schemes, and hard-to-measure variables, whose values are difficult to track during routine operations. Large plants, measurement delays, and environments hostile to the survival of measuring devices are all situations where hard-to-measure process variables are commonly encountered [21]. Similar situations can be addressed by utilizing soft sensors based on predictive models to forecast the true values of hard-to-measure variables. For modeling purposes, we assume that the true underlying relationship between the process variables and the product information or hard-to-measure variable can be expressed with a linear model of the form

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{1}$$

where

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix},$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix}, \quad \text{and} \quad \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

$\mathbf{y}$ is a $n \times 1$ vector of response variables, $\mathbf{X}$ is a $n \times p$ model matrix, $\boldsymbol{\beta}$ is a $p \times 1$ vector of regression coefficients, and $\boldsymbol{\epsilon}$ is a $n \times 1$ vector representing the noise, with covariance matrix $\sigma^2\mathbf{I}$. Here n represents the total number of observations and p the number of process variables (as well as the number of parameters in a model with main effects only and no intercept). If the predictors and the response are not centered, an intercept term may be added to the model. In that case, the size of the model matrix becomes $n \times (p+1)$, and $\boldsymbol{\beta}$ a $(p+1) \times 1$ vector. When $k \geq p$ observations are available to the learner, we can obtain a least squares estimator for $\boldsymbol{\beta}$ using

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y} \tag{2}$$

such that the fitted linear regression model will be given by $\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$ and its residuals by $\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}}$. A key distinction between the experimental design approach and stream-based active learning concerns the assumption we make about the randomness of the process variables. In design of experiments, the $\mathbf{x}$ vectors are assumed to be fixed while in this case we assume that $\mathbf{X}$ is composed by random vectors, as the individual observations are sampled from a process subject to random variation and we are not able to set the precise location of the incoming data points. However, conditional on the observed $\mathbf{X}$ variables, $(\mathbf{x}_1, \ldots, \mathbf{x}_p)$, Eq. (2) still applies. It should be noted that the coefficients $\hat{\boldsymbol{\beta}}$ determined using Eq. (2) may not be stable if the data matrix $\mathbf{X}$ is affected by multicollinearity. To deal with this issue and achieve robust results, a solution might be to use a ridge estimate for the coefficients, $\hat{\boldsymbol{\beta}}_{\text{ridge}} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X} - \lambda\mathbf{I}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}$. An alternative approach to tackle multicollinearity is to perform a pre-whitening of $\mathbf{X}$ to remove the dependencies between the components.

We assume a small, labeled training set is initially available and can be used to fit the first regression model, as is common practice in active learning applications [22–24]. The number of observations provided to the learner usually corresponds to a modest fraction (e.g., 5%) of the total number of instances available [25,26]. After the first model has been built, the learner is granted a certain operational budget b to augment the design matrix by including additional observations. Some approaches focus on this problem in a pool-based context, in which the total number of observations n is represented by a closed and static set $\mathcal{U}$ and the label of a specific data point can always be queried. Among these approaches, query-by-committee (QBC) [22] suggests building an ensemble of regression models trained on bootstrap replica of the original training set. Once the ensemble, or committee, has been built, the variance of the predictions made by the committee members is computed for each unlabeled observation $\mathbf{x} \in \mathcal{U}$. This metric, also referred to as ambiguity, is used to rank the instances belonging to the unlabeled set $\mathcal{U}$ by prioritizing the data points with the highest variance. Expected model change maximization (EMCM) [26] is another noteworthy study that focuses on the observations that impact the most the current model's parameters. The model change is defined as the difference between the current model parameters and the parameters obtained after fitting the model on the augmented design, including the unlabeled observation $\mathbf{x} \in \mathcal{U}$ that is currently under evaluation. Because the learner does not have access to the true label for that data point, it estimates it using the mean prediction of a bootstrap ensemble, as the one employed by QBC. Another offline approach, inspired by statistical process control, combines the Hotelling T^2 statistic and the squared prediction error of a principal component regression (PCR) model to obtain a sampling evaluation index [23].

Besides the fact that all these methods focus on the pool-based scenario, it should be noted that the approaches that use ensembles may not be well suited for the online scenario, given the higher computational cost associated with training and updating the models.

Optimal experimental theory is another field of research that is intrinsically related to active learning [27,28]. Optimal designs aim to reduce the cost of experimentation by proposing design matrices that allow a robust parameter estimation with the minimum number of runs. The most commonly employed optimality criteria are D-optimality [29] and A-optimality. Important properties of a design can be derived from the moment matrix, or information matrix, which is defined as

$$\mathbf{M} = \frac{\mathbf{X}^{\mathrm{T}}\mathbf{X}}{N} \tag{3}$$

where N represents the total number of runs. The moment matrix specifies the distribution of points in space and can be used to describe the design geometry. In a 2^{k} factorial design, where variables are expressed in coded units $(-1, +1)$, the moment matrix is equal to the identity matrix $\mathbf{I}_{\mathrm{k}}$, as the columns of the design are orthogonal. In an orthogonal design, all the parameters can be estimated independently of one another [30]. D-optimal designs try to pursue such property by focusing on good model parameter estimation. Inverting the moment matrix we obtain the scaled dispersion matrix given by

$$\mathbf{M}^{-1} = N\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1} \tag{4}$$

This matrix contains the variances and covariances of the estimated coefficients of the regression model, scaled by N/σ^2 [28]. Indeed, if the k observations used to estimate $\hat{\boldsymbol{\beta}}$ are i.i.d. and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$, we have

$$\hat{\boldsymbol{\beta}}_k|\mathbf{X} \sim \mathcal{N}\left(\boldsymbol{\beta}, \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\Sigma^2\right) \tag{5}$$

It can be demonstrated how by increasing the determinant of $\mathbf{M}$, the variances and covariances of the model coefficients are reduced, leading to a better estimation of $\boldsymbol{\beta}$. A D-optimal design is attained by maximizing the determinant of the moment matrix. Formally, we are seeking the design $\mathcal{D}^*$ that satisfies

$$\max_{\mathcal{D}} |\mathbf{M}(\mathcal{D})| = |\mathbf{M}\left(\mathcal{D}^*\right)| \tag{6}$$

A-optimality is another important optimality criterion that tries to achieve good parameter estimation by minimizing the sum of the individual variances of the coefficients. This is achieved by the design $\mathcal{D}^*$ that satisfies

$$\min_{\mathcal{D}} \mathrm{tr}\,[\mathbf{M}(\mathcal{D})]^{-1} = \mathrm{tr}\left[\mathbf{M}\left(\mathcal{D}^*\right)\right]^{-1} \tag{7}$$

as the variances of the coefficients can be found on the diagonal of the scaled dispersion matrix multiplied by σ^2/N. It should be noted that A-optimality does not consider the covariances between coefficients.

Recently, the concept of A-optimality has been extended to stream-based active learning [31,32]. That is, the approach has been extended outside the design of experiments framework, assuming $\mathbf{X}$ is composed of random vectors and the observations are sequentially drawn. Riquelme et al. [32] show how to set a threshold to perform online active learning for linear regression models by minimizing the sum of the individual variances of $\hat{\boldsymbol{\beta}}$. They state that, in order to achieve A-optimality and minimize the trace of the inversed scaled dispersion matrix, the eigenvalues of the moment matrix should be as balanced as possible. This is because the eigenvalues of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ represent the trace of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$, which is also given by the sum of the norm of the observations. For this reason, they propose a norm-thresholding algorithm that pursues A-optimality by selecting observations with large, scaled norm. The scaling step can be ignored when whitening is used to remove dependencies. Finally, the design is augmented with the observations $\mathbf{x}$ whose norm exceeds a threshold Γ given by

$$\mathrm{P_D}\left(\|\mathbf{x}\| \geq \Gamma\right) = \alpha \tag{8}$$

where α is the ratio of observations we are willing to label out of the incoming data stream. This value is strongly dependent on the budget b and the sampling rate used to collect the data.

Another noteworthy approach focusing on stream-based active learning for regression tasks has been suggested by Lughofer and Pratama [33]. In this paper, the authors propose a single-pass selection criterion that takes into account ignorance about the input space, uncertainty in predictive model outputs, and uncertainty in model parameters. The main difference with our approach is that Lughofer and Pratama focus on the use of Takagi–Sugeno (TS) fuzzy models [34], combining adaptive error bars for the model output and A-optimality for the variances of the estimated parameters. Conversely, our method relies on statistical linear regression and tries to combine the exploration of lesser-known input space regions with accurate parameter estimates by employing the idea of D-optimality.

3. Proposed approach

In this work, we try to improve the approach proposed by Riquelme et al. [32] by moving from A-optimality to D-optimality. We believe that taking into account the covariance between the estimates of the model coefficients might be particularly advantageous with large datasets and models, where many factors might be active and influence the response. To adapt the D-optimality criterion to stream-based active learning, we start from the connection between D-optimality and prediction variance (PV) highlighted by Myers et al. [28]. The PV at a point $\mathbf{x}^{(m)}$ is the variance of the predictor $\hat{\mathbf{y}}(\mathbf{x}^{(m)})$, which corresponds to $\mathrm{Var}(\mathbf{x}^{(m)\mathrm{T}}\hat{\boldsymbol{\beta}})$, and is given by

$$\mathrm{PV}(\mathbf{x}) = \sigma^2\mathbf{x}^{(m)\mathrm{T}}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{x}^{(m)} \tag{9}$$

where $\mathbf{x}^{(m)}$ represents the data point where the variance is being estimated, expanded to the model form. We can also express the variance in a scale-free form using the scaled prediction variance (SPV), which is computed as

$$\mathrm{SPV}(\mathbf{x}) = N\mathbf{x}^{(m)\mathrm{T}}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{x}^{(m)} \tag{10}$$

It should be noted that the SPV is a quadratic form of the inverse moment matrix $\mathbf{M^{-1}}$, as it can also be written as $\mathbf{x}^{(m)\mathrm{T}}\mathbf{M}^{-1}$ $\mathbf{x}^{(m)}$. Since SPV considers the total number of runs N, it can be used to assess the quality of a design on a per observation basis. In the online scenario, we are not interested in comparing designs of different sizes but rather we investigate the individual contributions of incoming data points to the current design. In this circumstance, we can discard N and use the unscaled prediction variance (UPV), which is calculated as

$$\mathrm{UPV}(\mathbf{x}) = \mathbf{x}^{(m)\mathrm{T}}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{x}^{(m)} \tag{11}$$

As anticipated in Section 2, we are already given an initial random design that contains some labeled examples, which is being used to fit an initial model. Then, we are interested in augmenting our design by iteratively selecting observations from a continuous stream. Pursuing D-optimality, we aim at collecting observations that allow us to maximize the determinant of the moment matrix $\mathbf{M}$. If we consider that the current design is composed by k observations, we can decompose the numerator of the moment matrix (Eq. (3)) before the design is augmented by including the $(k+1)th$ observation as

$$\mathbf{X}_k^{\mathrm{T}}\mathbf{X}_k = \mathbf{X}_{k+1}^{\mathrm{T}}\mathbf{X}_{k+1} - \mathbf{x}_{k+1}\mathbf{x}_{k+1}^{\mathrm{T}} \tag{12}$$

we can then express the determinant of $\mathbf{X}_{\mathrm{k}}^{\mathrm{T}}\mathbf{X}_{\mathrm{k}}$ as the product of the determinant of the numerator of the augmented moment matrix and a second term as in

$$\begin{aligned}|\mathbf{X}_k^{\mathrm{T}}\mathbf{X}_k| &= |\mathbf{X}_{k+1}^{\mathrm{T}}\mathbf{X}_{k+1} - \mathbf{x}_{k+1}\mathbf{x}_{k+1}^{\mathrm{T}}|\\ &= |\mathbf{X}_{k+1}^{\mathrm{T}}\mathbf{X}_{k+1}||1 - \mathbf{x}_{k+1}^{\mathrm{T}}\left(\mathbf{X}_{k+1}^{\mathrm{T}}\mathbf{X}_{k+1}\right)^{-1}\mathbf{x}_{k+1}|\end{aligned} \tag{13}$$

It should be noted that the second term of the above equation is a scalar, irrespective of the number of variables p and the number of observations k. From there, we can observe that

$$\frac{|\mathbf{X}_{k+1}^{\mathrm{T}}\mathbf{X}_{k+1}|}{|\mathbf{X}_k^{\mathrm{T}}\mathbf{X}_k|} = \frac{1}{1 - \mathbf{x}_{k+1}^{\mathrm{T}}\left(\mathbf{X}_{k+1}^{\mathrm{T}}\mathbf{X}_{k+1}\right)^{-1}\mathbf{x}_{k+1}} \tag{14}$$

From the properties of the hat matrix, which is generally defined as $\mathbf{H} = \mathbf{X}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}$, we know that $0 \le h_{jj} \le 1$ is true for each element h_{jj} of $\mathbf{H}$ [35]. It follows that $\mathbf{x}_{k+1}^{\mathrm{T}}\left(\mathbf{X}_{k+1}^{\mathrm{T}}\mathbf{X}_{k+1}\right)^{-1}\mathbf{x}_{k+1} \le 1$. Hence, we can conclude that the determinant of the new, enlarged, training set is maximized by seeking observations $\mathbf{x}$ that maximize $\mathbf{x}_{k+1}^{\mathrm{T}}\left(\mathbf{X}_{k+1}^{\mathrm{T}}\mathbf{X}_{k+1}\right)^{-1}\mathbf{x}_{k+1}$. That is, we will only select points that maximize the UPV. This may be explained by the fact that a data point for which we have a large prediction variance represents a less known region of the input space, and the regression model will highly benefit from its inclusion in the design. From Myers et al. [28] we have that maximizing $\mathbf{x}_{k+1}^{\mathrm{T}}$ $\left(\mathbf{X}_{k+1}^{\mathrm{T}}\mathbf{X}_{k+1}\right)^{-1}\mathbf{x}_{k+1}$ is equivalent to maximizing $\mathbf{x}_{k+1}^{\mathrm{T}}\left(\mathbf{X}_k^{\mathrm{T}}\mathbf{X}_k\right)^{-1}$ $\mathbf{x}_{k+1}$, which is the UPV using the fitted model before the new point has been added to the training set.

Finally, following the norm-thresholding approach, we can set an upper control limit on new observations as

$$\mathrm{P_D}\left(\mathbf{x}_{k+1}^{\mathrm{T}}\left(\mathbf{X}_k^{\mathrm{T}}\mathbf{X}_k\right)^{-1}\mathbf{x}_{k+1} \ge \varGamma\right) = \alpha \tag{15}$$

In practice, as suggested by Riquelme et al. [32], when we start to observe the data points coming from the process, we allocate a first initial set of points to estimate the distribution of $\mathbf{x}_{k+1}^{\mathrm{T}}\left(\mathbf{X}_k^{\mathrm{T}}\mathbf{X}_k\right)^{-1}\mathbf{x}_{k+1}$. In this work, we used kernel density estimation (KDE) with a Gaussian kernel. The initial set is also used to estimate the sample covariance matrix $\mathbf{\Sigma}$. By performing an eigenvalue decomposition we can then express $\mathbf{\Sigma}$ as $\mathbf{U\Lambda U}^{\mathrm{T}}$, where $\mathbf{U}$ is an orthogonal matrix, whose ith column corresponds to the ith eigenvector of $\mathbf{\Sigma}$, and $\mathbf{\Lambda}$ is a diagonal matrix with the eigenvalues of $\mathbf{\Sigma}$ on the diagonal. The incoming observations $\mathbf{x}$ can then be whitened using

$$\mathbf{z} = \mathbf{\Lambda}^{-1/2}\mathbf{U}^{\mathrm{T}}\mathbf{x} \tag{16}$$

Before the whitening step, data can be centered and scaled using the sample mean and variances obtained from the initial set. In industrial contexts, when a lot of unlabeled process data is available in the form of a historical database, this step can also be performed offline. In this case, by fitting a principal component analysis (PCA) model to the large unlabeled dataset and using it to transform the incoming observations, we could improve the predictive performance using a semi-supervised PCR as suggested by Frumosu and Kulahci [20]. The use of semi-supervised classification models has also received some attention in active-learning problems [36–38]. Indeed, semi-supervised learning and active learning are both techniques that deal with scarcity of labels. However, they do so in two different ways. With semi-supervised learning, we try to get the most out of the currently available unlabeled data, whereas with active learning we try to acquire new data in the most effective way.

Algorithm 1 describes the complete stream-based active learning procedure with the proposed approach, which might also be referred to as conditional D-optimality (CDO).

An alternative representation of the CDO active learning routine is reported in the flowcharts in Figs. 1 and 2. The first flowchart depicts the warm-up phase, which is represented by the steps from 1 to 10 of Algorithm 1. The warm-up set is very important for the algorithm and serves two main purposes. First, it allows to estimate the covariance matrix of the data, which is later used for whitening the incoming observations. Secondly, it provides a set of unlabeled observations that can be leveraged to estimate the distribution of the UPV. The primary purpose of the whitening step is to address the multicollinearity issue in linear regression modeling, which can be aggravated when dealing with real-world data. The whitening step also ensures comparability with the norm-thresholding approach. Indeed, the norm-thresholding method without whitening would require computing a weighted norm to deal with dependencies between the components.

The second flowchart represents the instance selection phase, the core of the active learning strategy. At this stage, we compute the UPV for the new observation sampled from the stream and we compare it to a pre-defined threshold. If the UPV computed at this point exceeds $\varGamma$, we query its label and include the labeled example in the training set. After the inclusion of the new point, a new threshold is estimated. The threshold is found by applying Eq. (15) to the whitened warm-up set $\mathbf{V}$. That is, $\mathbf{X}_k$ is substituted by $\mathbf{Z}$, the currently labeled training set after whitening, and $\mathbf{x}_{k+1}$ is given by each unlabeled data point belonging to $\mathbf{V}$. By doing so, we obtain a one-dimensional array that has the same cardinality as the number of observations in $\mathbf{V}$. These statistics are then used to approximate the distribution of the UPV using KDE and determine the α-upper percentile.

4. Experiments

In the experiments, we compare the proposed method to the norm-thresholding approach and random sampling. The methods are tested using numerical simulations and data from a chemical process simulator. All the approaches start from the same labeled

Algorithm 1 Stream-based active learning using CDO

Input: an initial random design $\mathbf{X}$; a data stream $\mathcal{S}$; a warm-up length w; a sampling rate α; a budget b
Output: an augmented design $\mathbf{Z}$

```
1:  Set $\mathbf{W} = \emptyset$                                  ▷ storing initial data to estimate $\mathbf{\Sigma}$ and $\Gamma$
2:  $i \leftarrow 1, c \leftarrow 0$                               ▷ $c$ represents the currently used budget
3:  while $i \leq w$ do
4:      Observe the $i^{th}$ data point $\mathbf{x}_i \in \mathcal{S}$
5:      Select $\mathbf{x}_i$: $\mathbf{W} = \mathbf{W} \cup \mathbf{x}_i$
6:      $i \leftarrow i + 1$
7:  end while
8:  Estimate the covariance matrix $\mathbf{\Sigma}$ of $\mathbf{W}$ and perform eigendecomposition $\mathbf{\Sigma} = \mathbf{U\Lambda U}^{\mathrm{T}}$
9:  Whiten the initial design by computing $\mathbf{Z} = \mathbf{\Lambda}^{-1/2}\ \mathbf{U}^{\mathrm{T}}\mathbf{X}$
10: Whiten the warm-up observations by computing $\mathbf{V} = \mathbf{\Lambda}^{-1/2}\ \mathbf{U}^{\mathrm{T}}\mathbf{W}$
11: Estimate $\Gamma$ using KDE on $\mathbf{V}$ with the desired sampling rate $\alpha$ using Equation 15 and datasets $\mathbf{Z}$ and $\mathbf{V}$
12: while $c \leq b$ and $i \leq |\mathcal{S}|$ do
13:     Observe the $i^{th}$ data point $\mathbf{x}_i \in \mathcal{S}$
14:     Whiten $\mathbf{x}_i$ by computing $\mathbf{z}_i = \mathbf{\Lambda}^{-1/2}\ \mathbf{U}^{\mathrm{T}}\mathbf{x}_i$
15:     if $\mathbf{z}_i^{\mathrm{T}}(\mathbf{Z}^{\mathrm{T}}\mathbf{Z})^{-1}\mathbf{z}_i \geq \Gamma$ then
16:         Ask for the label $y_i$ and augment the labeled dataset $\mathbf{Z} = \mathbf{Z} \cup \mathbf{z}_i$
17:         $c \leftarrow c + 1$
18:         Estimate new threshold $\Gamma$ to measure the prediction variance of the enlarged design (step 11)
19:     else
20:         Discard $\mathbf{x}_i$
21:     $i \leftarrow i + 1$
22:     end if
23: end while
24: return $\mathbf{Z}$
```

training set and then they iteratively augment the design until the budget constraint b is met. The performance of the models is expressed, in predictive terms, by the root mean squared error (RMSE) of the predictions on a separate test set of n observations

$$\mathrm{RMSE} = \sqrt{\sum_{i=1}^{n} \frac{\left(\hat{y}_i - y_i\right)^2}{n}} \tag{17}$$

4.1. Numerical simulations

To analyze the validity of the proposed method in the stream-based scenario, multiple datasets were created, each with a different dimensionality in terms of the number of process variables p. Within each dataset, incoming observations $\mathbf{x}$ are distributed according to a joint multivariate normal distribution $\mathcal{N}_p\left(\mathbf{0},\ \mathbf{\Sigma}_0\right)$, where $\mathbf{\Sigma}_0$ is given by $\sigma^2\mathbf{I}$, with $\sigma^2 = 1$. We ran 50 simulations for each number of p and, for each simulation run, the true coefficients are generated as $\boldsymbol{\beta} \sim U\left(-5, 5\right)$. It should be noted that $\boldsymbol{\beta}$ has the same dimensionality as $\mathbf{x}$. This means that, using a first order model, a coefficient for each process variable needs to be estimated. The noise is given by $\epsilon \sim \mathcal{N}(0, 1)$. For each scenario, an initial random design $\mathbf{X}$ is assumed available to the learner. We selected $p + 2$ number of observations for the initial design, as $k \geq p$ observations are needed to uniquely estimate $\hat{\boldsymbol{\beta}}$.

The learning curves reported in Figs. 3 and 4 show the difference between the RMSE obtained with the two active learning strategies, using random sampling as the baseline. For each learning step, the percentage RMSE difference reported in the plots is obtained by computing $(\mathrm{RMSE}_{\text{Active Learning}} - \mathrm{RMSE}_{\text{Random}})/\mathrm{RMSE}_{\text{Random}}{}^{*}100$. This allows us to display a scale-free performance metric while comparing the different scenarios. The plots reporting the learning curves with the absolute RMSE values are included in the appendix.

The methods are tested using $b = 50$ and with different levels for the α shown in Eq. (8), and 15. In the case of random sampling, α represents the probability of selecting an incoming observation. That is, each time a new sample arrives, a number $s \sim U(0, 1)$ is generated and the data point is only selected if $s \geq 1 - \alpha$. The warm-up length w was set to 500 observations and it is being used by all the methods to estimate the covariance matrix, which is used for whitening the observations in a semi-supervised fashion. Moreover, it ensures comparability between the three strategies by setting the same starting points for the data streams. The models have been fitted without the intercept term as both process variables and outcome are centered.

Fig. 3 shows the performance when using an α equal to 10%. The x-axis reports the learning steps, which correspond to the inclusion of an additional observation to the training set. Indeed, when the design is augmented, the model is updated and new predictions are obtained for the same separate test set. It should be noted that the RMSE obtained in the first learning step is the same for the three methods, as all the models start from the same random design. It can be seen how the performance of the two active learning methods converges to the one obtained through random sampling as the number of labeled examples in the training set increases. Instead, when the number of labeled examples is lower, active learning proves to be particularly convenient. However, the proposed approach dominates the other strategies in all the scenarios. Furthermore, it should be noted how the norm-thresholding algorithm seems to worsen when more and more parameters need to be estimated. Instead, CDO consistently provides enhanced predictive performances. We believe this may be due to the fact that, by imposing a threshold on the norm, A-optimality seeks only points that are far from the design's center, without ensuring a distance between the data points that have already been collected. CDO, on the other hand, emphasizes points that correspond to a poor prediction, which is more likely associated with a design area that the learner has not thoroughly explored. As a result, we are less prone to acquire the labels of data points in locations where we have already collected a significant number of observations.

It should be noted that in real-time applications the improvement offered by active learning is not as large as the one that can be obtained in offline scenarios, where we can deterministically maximize the desired optimality criterion over a closed set of observations. Moreover, by setting $\alpha = 10\%$ we are not being too demanding in terms of selecting observations with large norms for the A-optimality or high prediction variances for CDO. In Fig. 4, we try to widen the gap with the random strategy by lowering α, in this case up to 0.01. By raising the threshold, we can be more

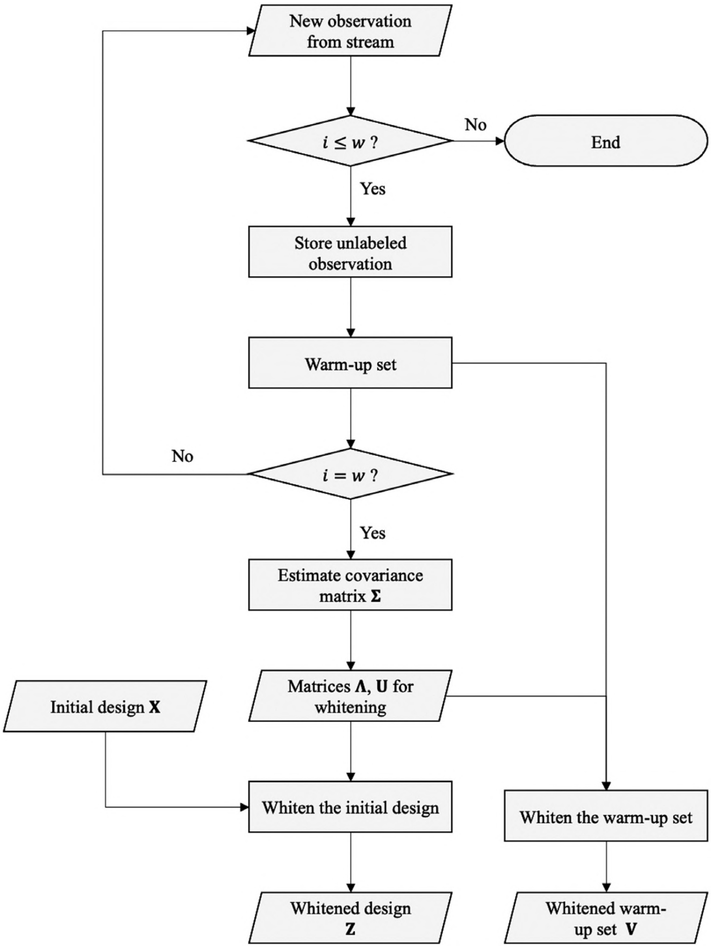

New observation from stream
$i \leq w$?
No
End
Yes
Store unlabeled observation
Warm-up set
No
$i = w$?
Yes
Estimate covariance matrix **Σ**
Initial design **X**
Matrices **Λ**, **U** for whitening
Whiten the initial design
Whiten the warm-up set
Whitened design **Z**
Whitened warm-up set **V**

Fig. 1. Flowchart of the warm-up phase of the stream-based active learning procedure.

demanding in terms of the desirability of the selected instances. The only drawback is that the algorithms will need to span more observations to achieve the desired size for the augmented design and meet the budget constraint. We believe this may not represent an issue since data is nowadays collected at very high sampling rates. However, in the final decision concerning the level of α, practitioners will need to make a trade-off between the desired prediction improvement and the time required to select the new labeled examples.

Fig. 4 reports the learning curves obtained using a smaller α. As expected, the enhancement obtained using the proposed strategy is increased with respect to the passive random sampling. However, it is worth noting that the improvement is more evident when the number of parameters is smaller, as the gain obtained in the high-dimensional cases was already significant with $\alpha = 10\%$.

Finally, we analyze the computational time required by the two active learning strategies. To this extent, we introduce a measure called average decision time, which quantifies the time required to decide whether to query the label of an unlabeled observation or discard it. The results obtained on the numerical simulations, for different number of process variables, are reported in Table 1. Both active learning strategies are highly efficient and do not require a high computational time. According to the CDO strategy, at each iteration we are simply computing the UPV for the new data point, as in step 15 of Algorithm 1, which requires less time than computing the norm of the new observation. It should be noted that the average decision time is lower because the inverse of the whitened moment matrix, $\left(\mathbf{Z}^{\mathrm{T}}\mathbf{Z}\right)^{-1}$, does not need to be computed at each iteration. However, it must be updated when the design is augmented by including an additional labeled observation. Updating and inverting the whitened moment matrix takes, on average, 0.31375 ms (ms).

Table 1
Average decision times (ms) for the two active learning methods (50 variables).

Strategy	10 variables	20 variables	50 variables	100 variables
CDO	0.00494	0.00527	0.00568	0.00690
Norm-thresholding	0.00635	0.00642	0.00673	0.00716

From an operational point of view, the average decision time is a highly relevant metric and it is closely related to the specific

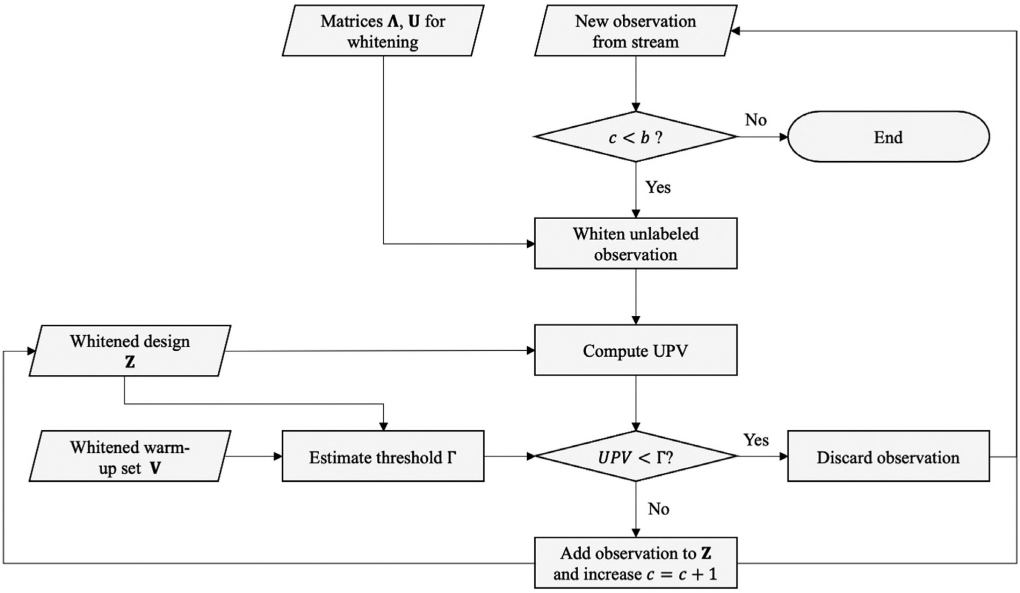

Matrices $\mathbf{\Lambda}$, $\mathbf{U}$ for whitening
New observation from stream
$c < b$?
No
End
Yes
Whiten unlabeled observation
Whitened design $\mathbf{Z}$
Compute UPV
Whitened warm-up set $\mathbf{V}$
Estimate threshold Γ
$UPV < \Gamma$?
Yes
Discard observation
No
Add observation to $\mathbf{Z}$ and increase $c = c + 1$

Fig. 2. Flowchart of the instance selection phase of the stream-based active learning procedure.

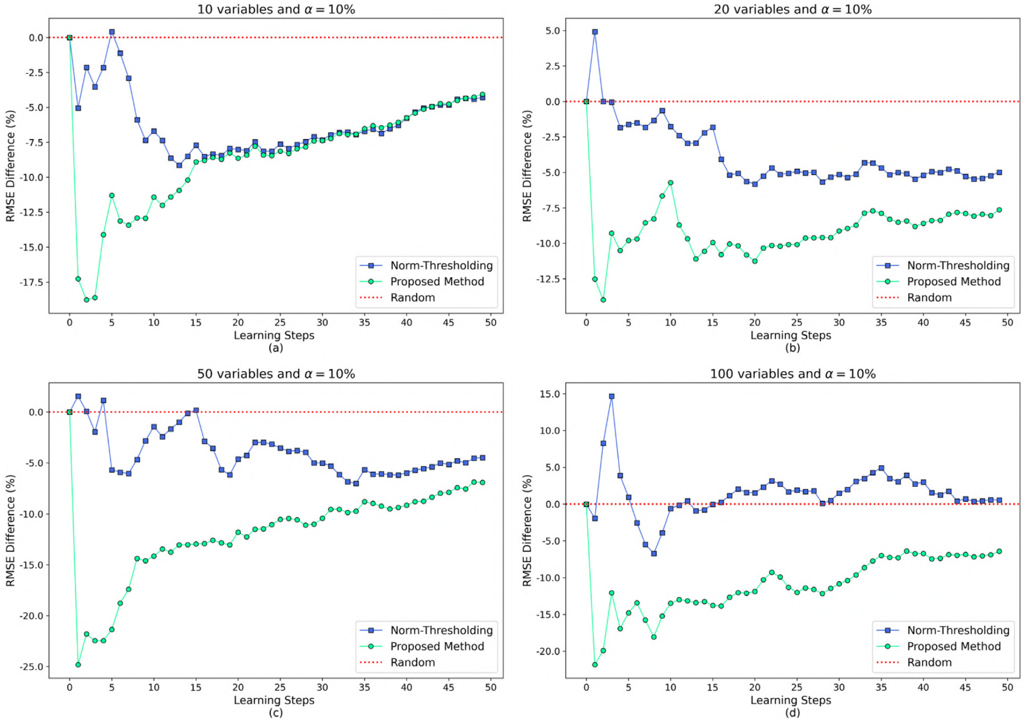


Fig. 3. Percentage difference in RMSE between random sampling and the active learning methods, using $\alpha = 10\%$ (50 simulations).

sampling frequency of the process. Indeed, to allow for a timely instance selection, the decision time should be strictly lower than the expiry date of the unlabeled data point, which is given by the time window where it is possible to query its label.

4.2. Tennessee Eastman Process

The Tennessee Eastman Process (TEP) is a commonly used benchmark in industrial and chemical engineering research and

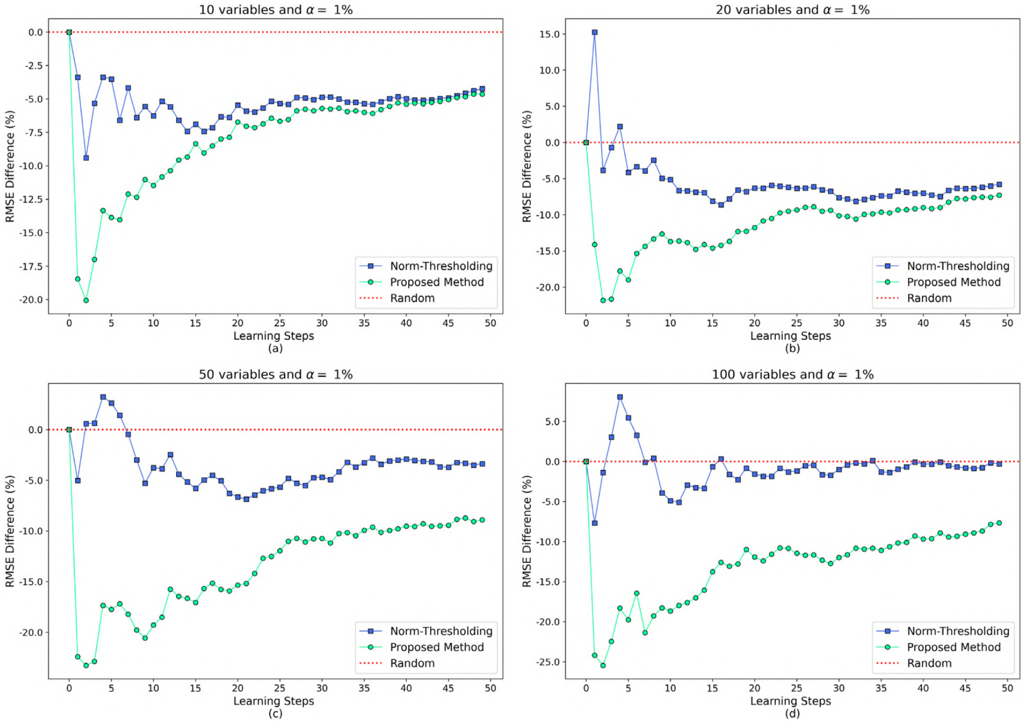


Fig. 4. Percentage difference in RMSE between random sampling and the active learning methods, using $\alpha = 1\%$ (50 simulations).

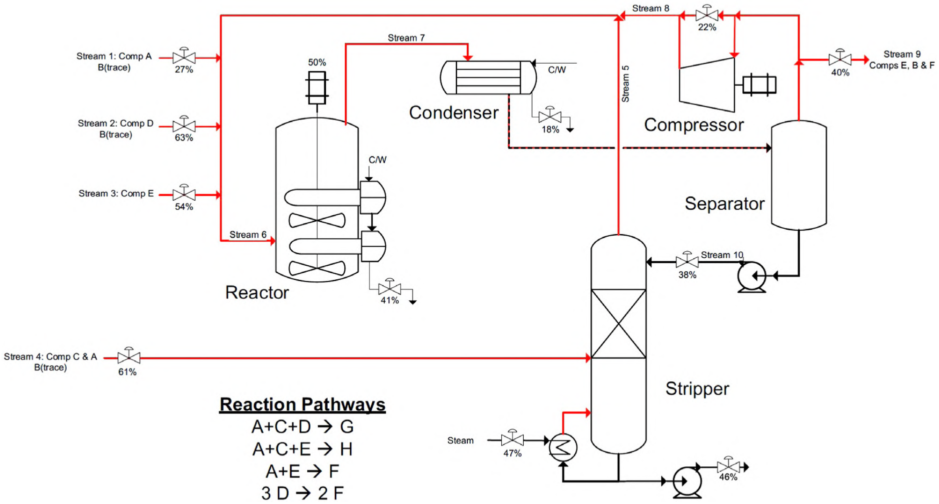


Fig. 5. The TEP piping and instrumentation diagram [39].

it has been thoroughly investigated in terms of process dynamics and control [40–44]. Recently, it has been also used to validate active learning or soft sensor modeling approaches [45–49]. It was initially published in 1993 [50] but since then it has been further developed and improved. For this study, we used a recently released MATLAB simulator to generate the data [39,51]. We generated 50 datasets with the process running in normal operating conditions, using a sampling rate of approximately 1 min. Fig. 5 depicts the TEP flowchart, which shows how the process is primarily composed of a reactor, a product condenser and separator, a stripper, and a compressor.

The TEP, like many other industrial processes, includes some easy-to-measure process variables whose real value can easily be monitored online, and some hard-to-measure variables,

Table 2
Variables of the TEP used as predictors in the regression models.

Number	Process variable	Code	Number	Process variable	Code
1	A feed	XMEAS1	9	Product separator temperature	XMEAS11
2	D feed	XMEAS2	10	Product separator pressure	XMEAS13
3	E feed	XMEAS3	11	Product separator underflow	XMEAS14
4	A and C feed	XMEAS4	12	Stripper pressure	XMEAS16
5	Recycle flow	XMEAS5	13	Stripper temperature	XMEAS18
6	Reactor feed rate	XMEAS6	14	Separator steam flow	XMEAS19
7	Reactor temperature	XMEAS9	15	Reactor cooling water outlet temperature	XMEAS21
8	Purge rate	XMEAS10	16	Separator cooling water outlet temperature	XMEAS22

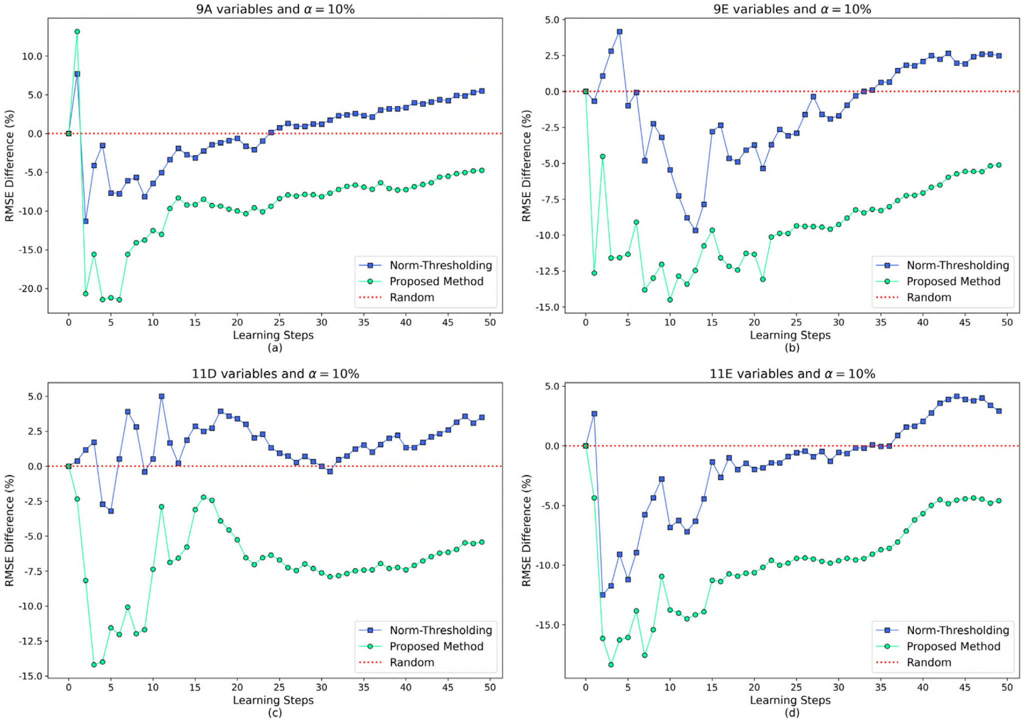


Fig. 6. Percentage difference in RMSE between random sampling and the active learning methods, using $\alpha = 10\%$ (50 simulations).

which are difficult to track during routine operations. Data-driven soft sensors are often developed to predict the latter in real-time. However, training regression models frequently necessitates a large number of labeled examples, and conducting quality inspections on chemical products may be costly and time-consuming. For this reason, optimizing the sampling strategy using active learning is highly desirable.

The 16 process variables shown in Table 2 are often used as predictors for the hard-to-measure process variables when testing active learning or soft sensor modeling approaches on the TEP. In most cases, the response variable is one of the composition measurements, such as the purge or product streams [45,47,48]. In this work, we selected two purge streams (Stream 9 A and Stream 9E) and two product streams (Stream 11D and Stream 11E) as the response to be predicted using the easy-to-measure variables.

As in the case of the numerical simulations, 50 datasets have been generated, and the average RMSE results are presented in the learning curve plots in Figs. 6 and 7. Most of the experimental parameters correspond to the ones used in the numerical study. The number of observations allocated to the first training set is equal to $p+2$, which in this case corresponds to 18. The warm-up length w is equal to 500 and the budget b is set to 50. The main difference from the models used in Section 4.1 is that, in this case, all the models include the intercept term.

We can see in Fig. 6 how the results obtained in Section 4.1 are still valid with data coming from a realistic industrial process simulator. Indeed, both the random and norm-thresholding approaches are outperformed by the proposed strategy. With regards to the level of α, the behavior observed in the numerical study does not seem to be altered and, as the threshold is raised, the performance gap between random sampling and active learning strategies widens. The plots of the learning curves with the absolute RMSE values are included in the appendix.

The plots in Fig. 8 show the residuals related to the first composition measurements analyzed, stream A of the purge. For illustrative purposes, the residuals refer to a smaller test set, composed of 100 observations. The first plot (a) shows the residuals obtained with the first random design, which is common to all the compared approaches. The remaining plots (b–d) illustrate the residuals obtained after five learning steps with each strategy. In general, we can see how the predictive performance improves

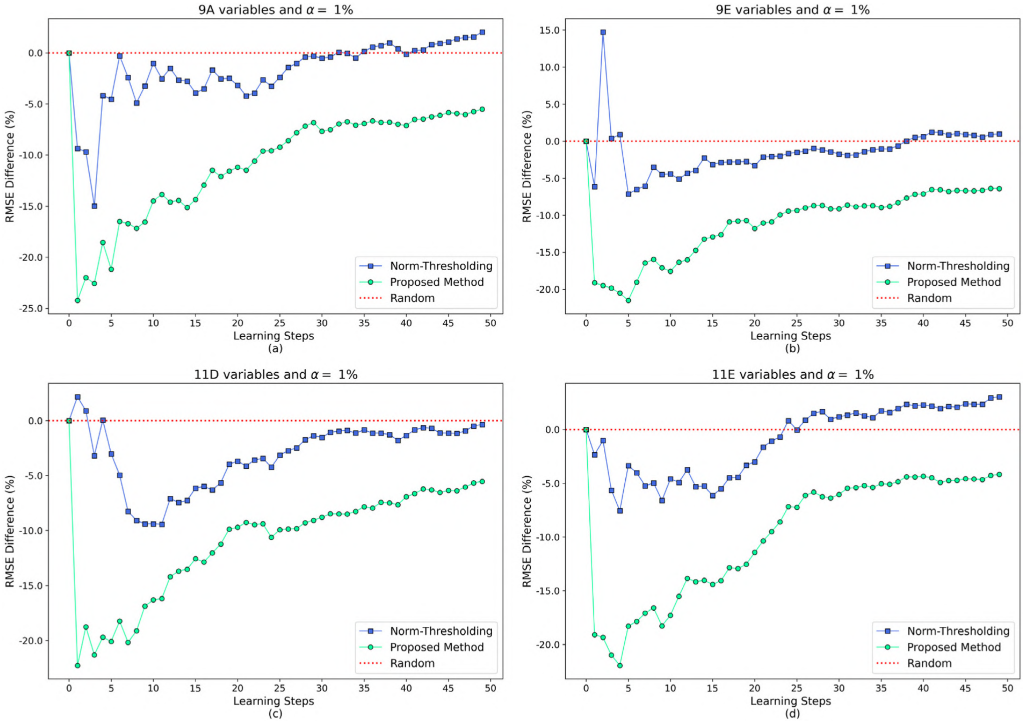


Fig. 7. Percentage difference in RMSE between random sampling and the active learning methods, using $\alpha = 1\%$ (50 simulations).

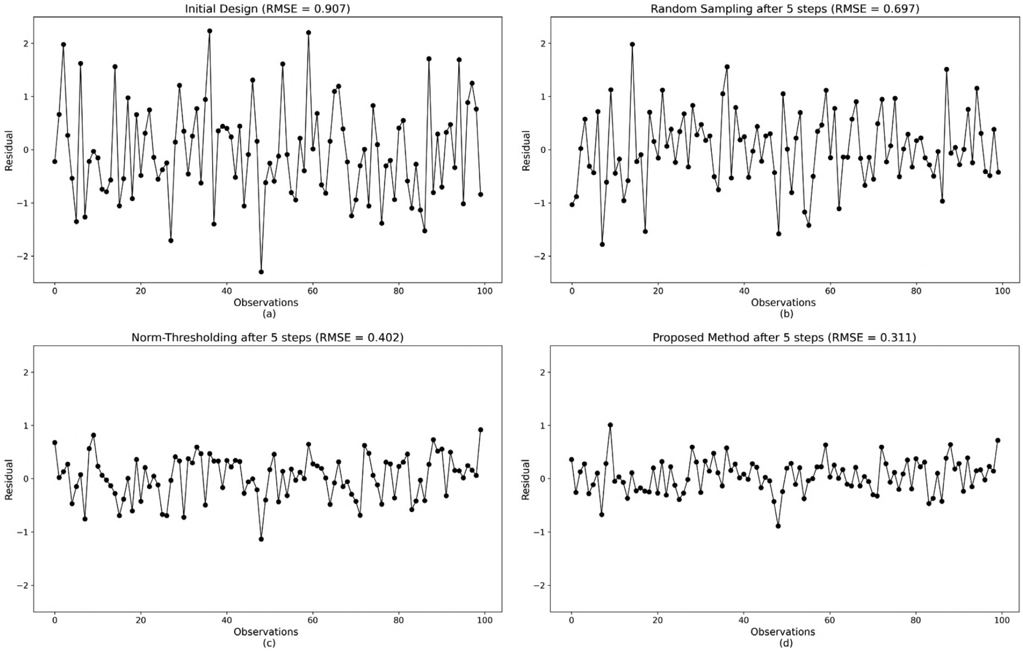


Fig. 8. Residuals of the Stream 9 A predictions: with the initial training set (a) and after augmenting the design with 5 additional labeled examples with the different methods (b–d) (one simulation with $\alpha = 1\%$).

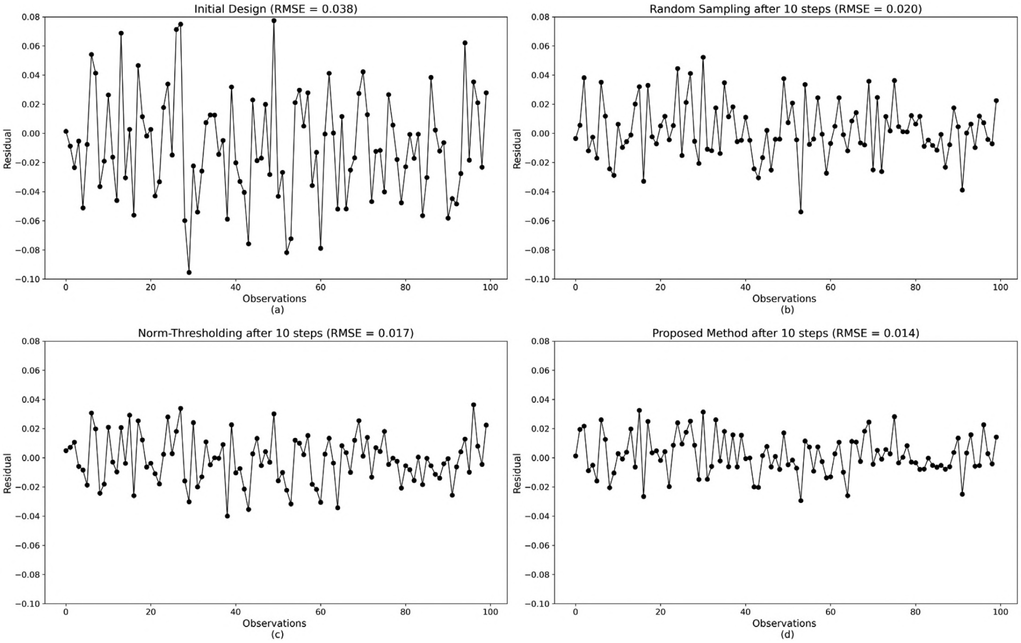


Fig. 9. Residuals of the Stream 11D predictions: with the initial training set (a) and after augmenting the design with 10 additional labeled examples with the different methods (b–d) (one simulation with $\alpha = 1\%$).

when more observations are included in the design. However, the predictions obtained with the proposed strategy are significantly better than the ones obtained with random sampling and norm-thresholding. Indeed, it should be noted how the RMSE obtained with the fifth model using CDO is 55 percent lower than the RMSE obtained with random sampling, and 23 percent lower than the RMSE obtained with the alternative active learning scheme. Finally, the improvement of CDO from the initial RMSE is higher than 65 percent.

It should be noted how a simple linear regression model fitted on a small training set can achieve compelling prediction results when the labeled examples are appropriately selected. This is true even when testing our approach on data from the TEP, which is characterized by highly nonlinear relationships.

Fig. 9 shows the predictions obtained for stream D of the product. In this case, to offer an additional view, we compared the models obtained after 10 learning steps. It can be seen how the behavior of the different schemes follows the same trend observed in Fig. 8. Indeed, after 10 iterations, the RMSE obtained with CDO is 18 percent lower than the one obtained by norm-thresholding and 30 percent lower than the one obtained with random sampling. From the initial design, the RMSE is reduced by more than 60 percent with CDO.

5. Conclusion

In many industrial processes and real-life applications, data is often abundant only in an unlabeled form. Moreover, the prohibitive cost required by quality inspections and the time required by manual annotation makes it unfeasible to label each data point with its quality characteristic. In these cases, active learning can significantly improve the predictive performance of regression models by smartly selecting the instances to include in the training set. In situations where many observations are sequentially processed, it is necessary to provide a real-time sampling strategy for selecting the most informative instances. In this paper, we propose an optimal strategy for performing stream-based active learning with linear regression models. Two case studies, one using numerical simulations and the other one using the TEP, show that the proposed approach offers improved predictive performance and reduces the prediction error faster.

CRediT authorship contribution statement

Davide Cacciarelli: Conceived and designed the analysis, Contributed data or analysis tools (implementation of the computer code), Performed the analysis, Wrote the paper. **Murat Kulahci:** Supervision, Performed the analysis, Wrote the paper. **John Sølve Tyssedal:** Conceived and designed the analysis, Provided guidance, Supervision, Performed the analysis, Wrote the paper.

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.

Appendix

The plots in Figs. 10–13 report the learning curves showing the RMSE values, without using random sampling as baseline. The plots begin from the third learning step to better show the differences between the curves. As all the models start from the same random design, the RMSE obtained in the first learning step is the same for the three methods as it is shown in Figs. 3, 4 and 6, 7.

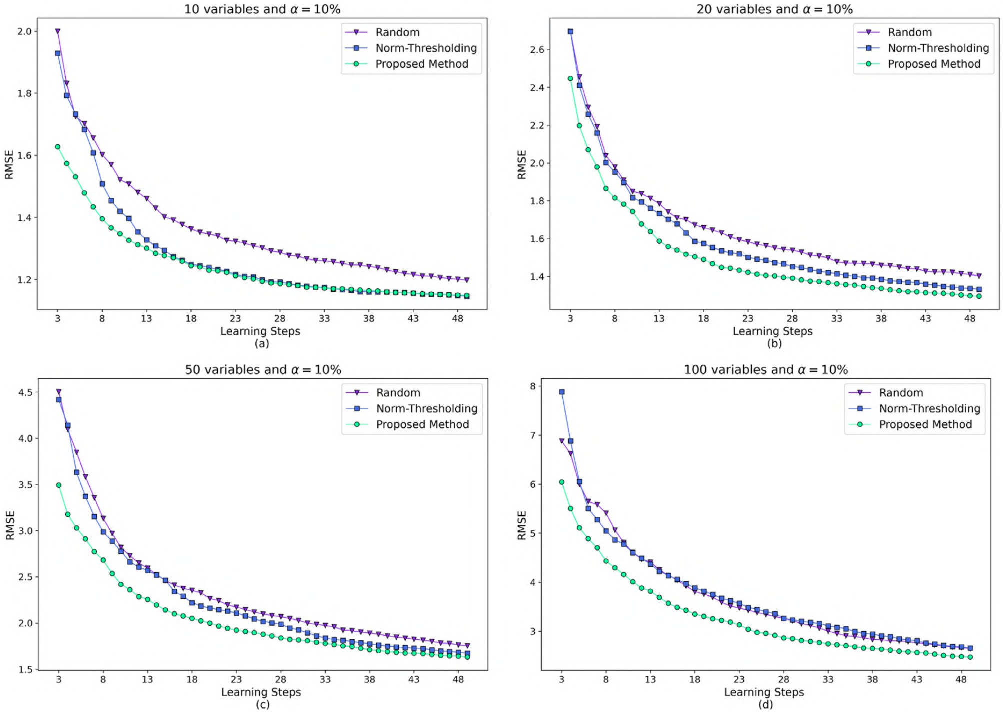


Fig. 10. Learning curves of different methods on numerical simulations with $\alpha = 10\%$ (50 simulations).

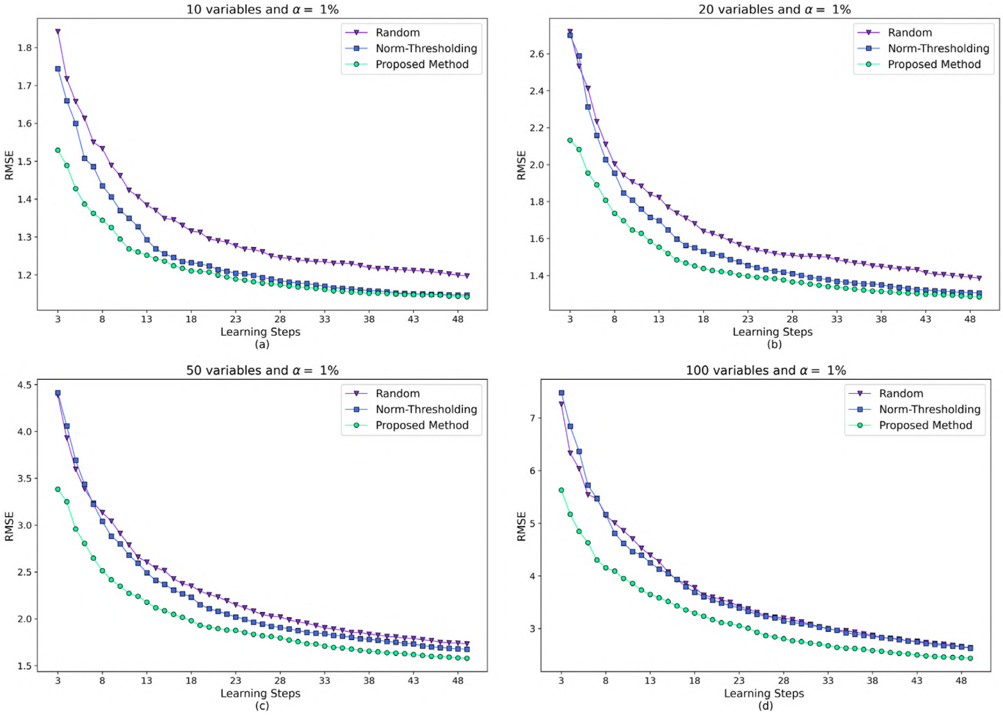


Fig. 11. Learning curves of different methods on numerical simulations with $\alpha = 1\%$ (50 simulations).

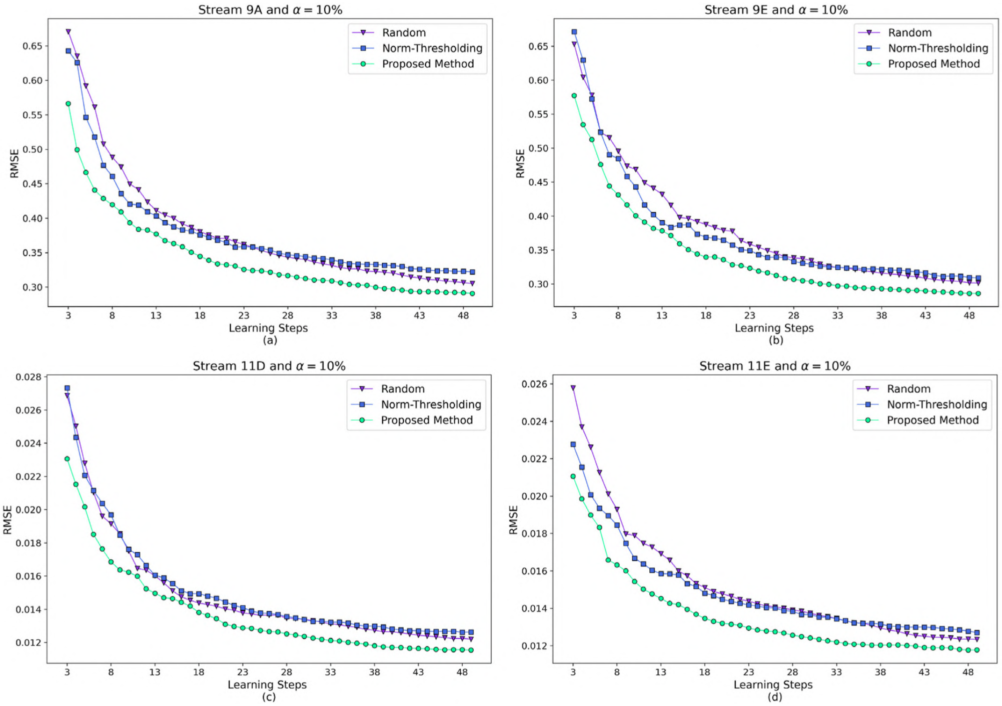


Fig. 12. Learning curves of different methods on TEP data with $\alpha = 10\%$ (50 simulations).

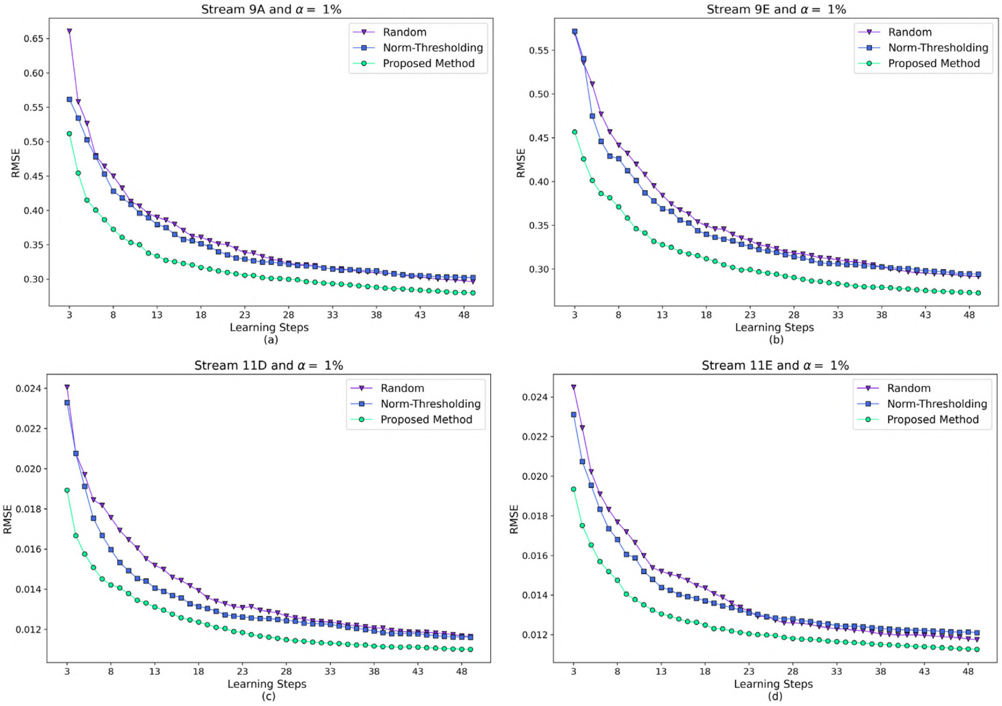


Fig. 13. Learning curves of different methods on TEP data with $\alpha = 1\%$ (50 simulations).

References

[1] P. Kumar, A. Gupta, Active learning query strategies for classification, regression, and clustering: A survey, J. Comput. Sci. Technol. 35 (2020) 913–945, http://dx.doi.org/10.1007/s11390-020-9487-4.

[2] B. Settles, Computer sciences department active learning literature survey, 2009.

[3] D.D. Lewis, W.A. Gale, A sequential algorithm for training text classifiers, 1994.

[4] D. Cohn, L. Atlas, R. Ladner, Improving generalization with active learning, Mach. Learn. 15 (1994) http://dx.doi.org/10.1007/BF00993277.

[5] P.R. Freeman, The secretary problem and its extensions: A review, 1983.

[6] E. Lughofer, On-line active learning: A new paradigm to improve practical useability of data stream modeling methods, Inform. Sci. 415–416 (2017) 356–376, http://dx.doi.org/10.1016/j.ins.2017.06.038.

[7] L.L.T. Chan, Q.Y. Wu, J. Chen, Dynamic soft sensors with active forward-update learning for selection of useful data from historical big database, Chemometr. Intell. Lab. Syst. 175 (2018) 87–103, http://dx.doi.org/10.1016/j.chemolab.2018.01.015.

[8] X. Shi, W. Xiong, Approximate linear dependence criteria with active learning for smart soft sensor design, Chemometr. Intell. Lab. Syst. 180 (2018) 88–95, http://dx.doi.org/10.1016/j.chemolab.2018.07.009.

[9] Z. Ge, Active learning strategy for smart soft sensor development under a small number of labeled data samples, J. Process Control. 24 (2014) 1454–1461, http://dx.doi.org/10.1016/j.jprocont.2014.06.015.

[10] Q. Tang, D. Li, Y. Xi, A new active learning strategy for soft sensor modeling based on feature reconstruction and uncertainty evaluation, Chemometr. Intell. Lab. Syst. 172 (2018) 43–51, http://dx.doi.org/10.1016/j.chemolab.2017.11.001.

[11] D. Macciò, Local linear regression for efficient data-driven control, Knowl.-Based Syst. 98 (2016) 55–67, http://dx.doi.org/10.1016/j.knosys.2015.12.012.

[12] D. Alvarez Melis, T. Jaakkola, Towards robust interpretability with self-explaining neural networks, in: S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc., 2018, https://proceedings.neurips.cc/paper/2018/file/3e9f0fc9b2f89e043bc6233994dfcf76-Paper.pdf.

[13] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, Ann. Stat. 32 (2004) http://dx.doi.org/10.1214/009053604000000067.

[14] C.-X. Zhang, J.-S. Zhang, Q.-Y. Yin, Early stopping aggregation in selective variable selection ensembles for high-dimensional linear regression models, Knowl.-Based Syst. 153 (2018) 1–11, http://dx.doi.org/10.1016/j.knosys.2018.04.016.

[15] C. Hong, J. Yu, J. Wan, D. Tao, M. Wang, Multimodal deep autoencoder for human pose recovery, IEEE Trans. Image Process. 24 (2015) 5659–5670, http://dx.doi.org/10.1109/TIP.2015.2487860.

[16] C. Hong, J. Yu, D. Tao, M. Wang, Image-based 3D human pose recovery by multi-view locality sensitive sparse retrieval, IEEE Trans. Ind. Electron. (2014) 1, http://dx.doi.org/10.1109/TIE.2014.2378735.

[17] J. Yu, M. Tan, H. Zhang, Y. Rui, D. Tao, Hierarchical deep click feature prediction for fine-grained image recognition, IEEE Trans. Pattern Anal. Mach. Intell. 44 (2022) 563–578, http://dx.doi.org/10.1109/TPAMI.2019.2932058.

[18] J. Yu, D. Tao, M. Wang, Y. Rui, Learning to rank using user clicks and visual features for image retrieval, IEEE Trans. Cybern. 45 (2015) 767–779, http://dx.doi.org/10.1109/TCYB.2014.2336697.

[19] C. Hong, J. Yu, J. Zhang, X. Jin, K.-H. Lee, Multimodal face-pose estimation with multitask manifold deep learning, IEEE Trans. Ind. Inform. 15 (2019) 3952–3961, http://dx.doi.org/10.1109/TII.2018.2884211.

[20] F.D. Frumosu, M. Kulahci, Big data analytics using semi-supervised learning methods, Qual. Reliab. Eng. Int. 34 (2018) 1413–1423, http://dx.doi.org/10.1002/qre.2338.

[21] L. Fortuna, S. Graziani, A. Rizzo, M.G. Xibilia, Soft Sensors for Monitoring and Control of Industrial Processes, Springer, 2007.

[22] R. Burbidge, J.J. Rowland, R.D. King, Active Learning for Regression based on Query by Committee.

[23] Z. Ge, Active learning strategy for smart soft sensor development under a small number of labeled data samples, J. Process Control. 24 (2014) 1454–1461, http://dx.doi.org/10.1016/j.jprocont.2014.06.015.

[24] Z. Ge, Active probabilistic sample selection for intelligent soft sensing of industrial processes, Chemometr. Intell. Lab. Syst. 151 (2016) 181–189, http://dx.doi.org/10.1016/j.chemolab.2016.01.003.

[25] O. Reyes, A.H. Altalhi, S. Ventura, Statistical comparisons of active learning strategies over multiple datasets, Knowl.-Based Syst. 145 (2018) 274–288, http://dx.doi.org/10.1016/j.knosys.2018.01.033.

[26] W. Cai, Y. Zhang, J. Zhou, Maximizing expected model change for active learning in regression, in: Proceedings - IEEE International Conference on Data Mining, ICDM, 2013, pp. 51–60, http://dx.doi.org/10.1109/ICDM.2013.104.

[27] S. Karlin, J. William, Studden, optimal experimental designs, Ann. Math. Stat. 37 (1966) 783–815.

[28] R.H. Myers, D. Montgomery, C.M. Anderson-Cook, Response surface methodology: process and product optimization using designed experiments, 2016.

[29] R.C. st. John, N.R. Draper, D-optimality for regression designs: A review, Technometrics 17 (1975) 15–23, http://dx.doi.org/10.1080/00401706.1975.10489266.

[30] D.C. Montgomery, Design and Analysis of Experiments, John Wiley & Sons Inc., Hoboken, NJ, USA, 2012, http://dx.doi.org/10.1002/9781118147634.

[31] X. Fontaine, P. Perrault, M. Valko, V. Perchet, Online a-Optimal Design and Active Linear Regression, 2021.

[32] C. Riquelme, R. Johari, B. Zhang, Online active linear regression via thresholding, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017, www.aaai.org.

[33] E. Lughofer, M. Pratama, Online active learning in data stream regression using uncertainty sampling based on evolving generalized fuzzy models, IEEE Trans. Fuzzy Syst. 26 (2018) 292–309, http://dx.doi.org/10.1109/TFUZZ.2017.2654504.

[34] E. Lughofer, Evolving Fuzzy Systems – Methodologies, Advanced Concepts and Applications, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, http://dx.doi.org/10.1007/978-3-642-18087-3.

[35] D.C. Hoaglin, R.E. Welsch, The Hat matrix in regression and ANOVA, Am. Stat. 32 (1978) 17, http://dx.doi.org/10.2307/2683469.

[36] G. He, Y. Li, W. Zhao, An uncertainty and density based active semi-supervised learning scheme for positive unlabeled multivariate time series classification, Knowl.-Based Syst. 124 (2017) 80–92, http://dx.doi.org/10.1016/j.knosys.2017.03.004.

[37] M.C. Fernandes, T.F. Covões, A.L.V. Pereira, Improving evolutionary constrained clustering using active learning, Knowl.-Based Syst. 209 (2020) 106452, http://dx.doi.org/10.1016/j.knosys.2020.106452.

[38] Y. Leng, X. Xu, G. Qi, Combining active learning and semi-supervised learning to construct SVM classifier, Knowl.-Based Syst. 44 (2013) 121–131, http://dx.doi.org/10.1016/j.knosys.2013.01.032.

[39] E.B. Andersen, I.A. Udugama, K. v. Gernaey, A.R. Khan, C. Bayer, M. Kulahci, An easy to use GUI for simulating big data using Tennessee Eastman process, Qual. Reliab. Eng. Int. 38 (2022) 264–282, http://dx.doi.org/10.1002/qre.2975.

[40] N.L. Ricker, Optimal steady-state operation of the Tennessee Eastman challenge process, Comput. Chem. Eng. 19 (1995) http://dx.doi.org/10.1016/0098-1354(94)00043-N.

[41] N. Lawrence Ricker, Decentralized control of the Tennessee Eastman challenge process, J. Process Control. 6 (1996) http://dx.doi.org/10.1016/0959-1524(96)00031-5.

[42] T.J. McAvoy, N. Ye, Base control for the Tennessee Eastman problem, Comput. Chem. Eng. 18 (1994) http://dx.doi.org/10.1016/0098-1354(94)88019-0.

[43] F. Capaci, E. Vanhatalo, M. Kulahci, B. Bergquist, The revised Tennessee Eastman process simulator as testbed for SPC and DoE methods, Qual. Eng. 31 (2019) http://dx.doi.org/10.1080/08982112.2018.1461905.

[44] P.R. Lyman, C. Georgakis, Plant-wide control of the Tennessee Eastman problem, Comput. Chem. Eng. 19 (1995) http://dx.doi.org/10.1016/0098-1354(94)00057-U.

[45] L. Bao, X. Yuan, Z. Ge, Co-training partial least squares model for semi-supervised soft sensor development, Chemometr. Intell. Lab. Syst. 147 (2015) 75–85, http://dx.doi.org/10.1016/j.chemolab.2015.08.002.

[46] X. Jia, W. Tian, C. Li, X. Yang, Z. Luo, H. Wang, A dynamic active safe semi-supervised learning framework for fault identification in labeled expensive chemical processes, Processes 8 (2020) http://dx.doi.org/10.3390/pr8010105.

[47] J. Zhu, Z. Ge, Z. Song, Robust semi-supervised mixture probabilistic principal component regression model development and application to soft sensors, J. Process Control. 32 (2015) 25–37, http://dx.doi.org/10.1016/j.jprocont.2015.04.015.

[48] R. Grbić, D. Slišković, P. Kadlec, Adaptive soft sensor for online prediction and process monitoring based on a mixture of Gaussian process models, Comput. Chem. Eng. 58 (2013) 84–97, http://dx.doi.org/10.1016/j.compchemeng.2013.06.014.

[49] L. Yin, H. Wang, W. Fan, Active learning based support vector data description method for robust novelty detection, Knowl.-Based Syst. 153 (2018) 40–52, http://dx.doi.org/10.1016/j.knosys.2018.04.020.

[50] J.J. Downs, E.F. Vogel, A plant-wide industrial process control problem, Comput. Chem. Eng. 17 (1993) http://dx.doi.org/10.1016/0098-1354(93)80018.

[51] C. Reinartz, M. Kulahci, O. Ravn, An extended Tennessee eastman simulation dataset for fault-detection and decision support systems, Comput. Chem. Eng. 149 (2021) 107281, http://dx.doi.org/10.1016/j.compchemeng.2021.107281.

APPENDIX D

PAPER 4 – Robust online active learning

Received: 9 January 2023 | Revised: 24 May 2023 | Accepted: 26 May 2023

DOI: 10.1002/qre.3392

WILEY

SPECIAL ISSUE ARTICLE

Robust online active learning

Davide Cacciarelli[1,2] | **Murat Kulahci**[1,3] | **John Sølve Tyssedal**[2]

[1] Department of Applied Mathematics and Computer Science, Technical University of Denmark, Kgs. Lyngby, Denmark

[2] Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway

[3] Department of Business Administration, Technology and Social Sciences, Luleå University of Technology, Luleå, Sweden

Correspondence
Davide Cacciarelli, Department of Applied Mathematics and Computer Science, Technical University of Denmark, Kgs, Lyngby, Denmark.
Email: dcac@dtu.dk

Funding information
DTU Strategic Alliances Fund

Abstract

In many industrial applications, obtaining labeled observations is not straightforward as it often requires the intervention of human experts or the use of expensive testing equipment. In these circumstances, active learning can be highly beneficial in suggesting the most informative data points to be used when fitting a model. Reducing the number of observations needed for model development alleviates both the computational burden required for training and the operational expenses related to labeling. Online active learning, in particular, is useful in high-volume production processes where the decision about the acquisition of the label for a data point needs to be taken within an extremely short time frame. However, despite the recent efforts to develop online active learning strategies, the behavior of these methods in the presence of outliers has not been thoroughly examined. In this work, we investigate the performance of online active linear regression in contaminated data streams. Our study shows that the currently available query strategies are prone to sample outliers, whose inclusion in the training set eventually degrades the predictive performance of the models. To address this issue, we propose a solution that bounds the search area of a conditional D-optimal algorithm and uses a robust estimator. Our approach strikes a balance between exploring unseen regions of the input space and protecting against outliers. Through numerical simulations, we show that the proposed method is effective in improving the performance of online active learning in the presence of outliers, thus expanding the potential applications of this powerful tool.

KEYWORDS

active learning, data stream, optimal experimental design, outliers, robust regression, unlabeled data

1 | INTRODUCTION

Predictive models often need to be trained on a large amount of labeled data before being deployed. However, in industrial applications data is often abundant only in an unlabeled form. Active learning strategies provide a solution to this problem

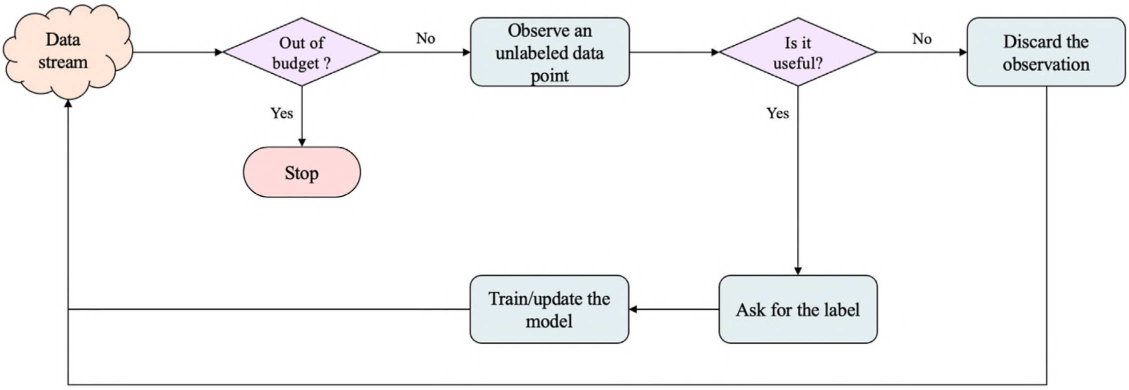


FIGURE 1 General online active learning flowchart.

by prioritizing the labeling of the most useful instances for building the model, thus accelerating the convergence of its learning curve.[1] Active learning problems can be classified into three macro-scenarios.[2] The first and most studied scenario is the pool-based scenario, where the learner can select the most useful instances to be labeled by maximizing an evaluation criterion over a closed set of observations. The second scenario is referred to as membership query synthesis, and it allows the learner to query the labels of synthetically generated instances rather than those sampled from the process distribution. Finally, the third scenario is online, or stream-based, active learning.[3] In this case, the unlabeled observations are drawn sequentially by the learner, which must immediately decide whether to keep the instance and query its label or discard it. While many researchers have been working on active learning in the recent years, the pool-based scenario has received the most attention.[4,5]

Although online active learning has become more popular in the last few years,[6–10] the majority of the methods have been developed for classification tasks.[11] An interesting approach to online active learning for fuzzy regression models has been proposed by Lughofer.[12] Other researchers tried to adapt the optimality criteria of the experimental design theory to the online active linear regression framework.[13–16] Linear regression models are still very useful in industrial applications as they can be efficiently trained on a small number of observations. They are able to offer a straightforward interpretation, along with the possibility of constructing confidence intervals on the parameter estimates.[17,18] They can also be easily coupled with variable selection and robust estimation methods. Furthermore, whereas many pool-based active learning approaches employ ensemble methods or complex models, linear models can support online active learning due to the decreased computational cost associated with model training and updating.

Figure 1 depicts a general online active learning flowchart. The main difference among the query strategies lies in how they assess the usefulness of an unlabeled instance when the learner samples it from the data stream. Another important aspect is the assumptions on the input distribution. Indeed, despite the increased interest in the online active linear regression framework, the performance of the sampling strategies in the presence of outliers has not been thoroughly explored. The few works we are aware of that analyze this issue, are related to the pool-based scenario. Deldossi et al.[19] highlighted how sampling methods based on D-optimality are affected by outliers and high leverage points. Zhao et al.[20] focused on robust active representations based on the $\ell_{2,p}$-norm constraints for selecting highly representative data. Finally, He et al.[21] emphasized the problem of being prone to sample outliers while proposing a semi-supervised active learning strategy for multivariate time series classification, using uncertainty and local density.

In this paper, we study the problem of learning from contaminated data streams with limited sampling resources. We first investigate the effects of outliers on the sampling decisions made by state-of-the-art online active learning approaches for linear regression, and successively propose a solution for this issue. It should be noted that the presence of outliers considered in this work cannot be tackled using traditional anomaly detection methods. Indeed, most unsupervised anomaly detection strategies rely on the assumption that a large training set free from outliers, usually referred to as phase I data in the statistical process control literature, is available beforehand.[22–25] However, this assumption is violated in many practical applications,[26] especially in label-scarce scenarios where few to no labels are available before the beginning of the active learning routine. The proposed strategy for online active learning utilizes a double-threshold approach to limit the search area of a conditional D-optimality (CDO) algorithm. By using two thresholds, the strategy aims to identify informative data points while excluding outliers. In cases of highly contaminated environments, robust estimators based on the Huber and Tukey bisquare loss are employed.

The remainder of this paper is organized as follows. In Section 2, we introduce the terminology and describe the sampling strategies that are used as the baseline in our analysis. Section 3 offers a review on the use of robust estimators and introduces ways of modifying the CDO algorithm. In Section 4, we test our approach using numerical simulations in four scenarios, using different contamination ratios. Section 5 offers a discussion on the results obtained. Finally, Section 6 provides some conclusions.

2 | BACKGROUND AND RELATED WORK

The labeled observations that are collected from the contaminated data stream are used to fit a linear model of the form

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{1}$$

where $\mathbf{y}$ is an $n \times 1$ vector of response variables, $\mathbf{X}$ is an $n \times p$ model matrix, $\boldsymbol{\beta}$ is a $p \times 1$ vector of regression coefficients, and $\boldsymbol{\varepsilon}$ is an $n \times 1$ vector representing the zero-mean Gaussian noise. Here, n represents the total number of observations and p the number of variables. Before starting the active learning routine and the collection of additional labels, it is commonly assumed to have at our disposal an initial set of labeled observations.[5,27,28] This set is used to obtain an initial estimate $\hat{\boldsymbol{\beta}}$ for the coefficients $\boldsymbol{\beta}$. Using an ordinary least squares (OLS) estimator, we have that $\hat{\boldsymbol{\beta}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\,\mathbf{X}^{\mathrm{T}}\mathbf{y}$. Then, the fitted linear regression model is $\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$, and the residuals are obtained as $\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}}$. When the variables are highly correlated, a pre-whitening might be performed to avoid an ill-conditioned problem when computing $(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}$. It should be noted that the matrix $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ is important to obtain information about the design geometry. In particular, for a design composed of n runs, the moment matrix, $\mathbf{M} = \mathbf{X}^{\mathrm{T}}\mathbf{X}/n$, plays a central role in the definition of optimal experimental designs. The two most commonly employed optimality criteria, which have been adapted for the online active learning scenario, are A-optimality and D-optimality. An A-optimal design is achieved by minimizing the trace of the inverse of the moment matrix $\mathbf{M}$. It can be shown how this corresponds to minimizing the individual variances of the estimated coefficients. This approach has been adapted for the online active linear regression framework by Riquelme et al.[14] They proposed a norm-thresholding algorithm that only selects observations $\mathbf{x}$ with large, scaled norm by estimating a threshold Γ as

$$\mathrm{P}_{\mathrm{D}}\left(\|\mathbf{x}\| \geq \Gamma\right) = \alpha \tag{2}$$

where α is the ratio of observations we are willing to label out of the incoming data stream. The probability distribution of the norms can be approximated using kernel density estimation (KDE) on a set of unlabeled observations $\mathbf{C}$, which can be regarded as a warm-up or calibration set and can either be retrieved from historical data or by observing the data stream for a while. Using this thresholding approach, we would be sampling, with high probability, observations that help achieve A-optimality. Given n statistics, $(s_1, \ldots, s_n)$, KDE can be used to estimate the shape of an unknown distribution f using

$$\hat{f}(s) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}K\left(\frac{s - s_i}{h}\right) \tag{3}$$

where the bandwidth h is a positive number that is used to control the amount of smoothing, and the kernel K is a smooth function such that $K(s) \geq 0$, $\int K(s)ds = 1$, $\int sK(s)ds = 0$ and $\sigma_K^2 \equiv \int s^2K(s)ds > 0$. In this paper, the Gaussian (Normal) kernel, $K(s) = (2\,\pi)^{-1/2}\,e^{-s^2/2}$ is used.

D-optimality is another fundamental criterion,[29] which takes both the variances and covariances of the model coefficients into account by maximizing the determinant of the moment matrix $\mathbf{M}$. As in the case of A-optimality, D-optimality has been adapted to the online active learning scenario with the proposal of a CDO algorithm.[16] CDO suggests setting a threshold Γ by using

$$\mathrm{P}_{\mathrm{D}}\left(\mathbf{x}_{l+1}^{\mathrm{T}}\left(\mathbf{X}_l^{\mathrm{T}}\mathbf{X}_l\right)^{-1}\mathbf{x}_{l+1} \geq \Gamma\right) = \alpha \tag{4}$$

where $\mathbf{X}_l$ is the model matrix with the l labeled observations currently available and $\mathbf{x}_{l+1}$ is the unlabeled data point that is under evaluation. It can be shown that by selecting observations that maximize $\mathbf{x}_{l+1}^{\mathrm{T}}(\mathbf{X}_l^{\mathrm{T}}\mathbf{X}_l)^{-1}\mathbf{x}_{l+1}$, we are at the same time seeking D-optimality and labeling observations with a large unscaled prediction variance (UPV),[30] which is generally defined as

$$\mathrm{UPV}\,(\mathbf{x}) = \mathbf{x}^{(m)\mathrm{T}} \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1} \mathbf{x}^{(m)} \tag{5}$$

where $\mathbf{x}^{(m)}$ represents the data point where the UPV is being estimated, expanded to the model form (e.g., if polynomial features are added to the model). To estimate the threshold Γ, we use KDE after computing the UPV of all the observations in $\mathbf{C}$. The CDO intuition is coherent with the idea that a point for which we have a large UPV value represents a less explored region of the input space and will help, with high probability, attaining D-optimality, conditional on the already collected observations. The equivalence between sampling data points with high UPV and D-optimality is demonstrated in our previous work.[16]

Given these preliminaries, we now propose methods that are robust to the presence of outliers in the data stream.

3 | METHODS

When training a linear regression model on a dataset corrupted by the presence of outliers, a simple yet effective solution is to resort to the use of robust estimators. An extensive overview of robust regression has been provided by Fox and Weisberg.[31] In general, robust estimation methods attempt to estimate the coefficients $\hat{\boldsymbol{\beta}}$ by minimizing a particular loss function given by

$$\mathcal{I} = \sum_{i=1}^{n} \rho\,(e_i) \;= \sum_{i=1}^{n} \rho\left(y_i - \mathbf{x}_i\hat{\boldsymbol{\beta}}\right) \tag{6}$$

where ρ is a function that regulates the contribution of each residual to the loss, and e_i is the residual for the ith observation $(\mathbf{x}_i,\ y_i)$. The function ρ is nonnegative, equal to zero when the argument is zero, symmetrical and monotone in $|e|$. In the case of an OLS estimator, the loss is given by

$$\rho_{LS} = e^2 \tag{7}$$

It can be seen how the objective function minimized by an OLS estimator is equally affected by all the observations for which we measure the residuals. Instead, robust estimators try to reduce the impact of observation with very large residuals on the estimation of $\hat{\boldsymbol{\beta}}$. One of the most popular robust loss functions is the Huber loss,[32] which is defined as

$$\rho_H = \begin{cases} e^2 & for\,|e| \le k \\ 2k\,|e| - k^2 & for\,|e| > k \end{cases} \tag{8}$$

where k is a tuning parameter, which is usually set to 1.345σ to achieve 95% efficiency when the errors are normally distributed, while keeping a good protection against outliers.[31] It can be seen how the contribution of each observation is reduced based on the magnitude of the corresponding residual. However, despite being much more robust than the OLS estimator, the Huber loss is still proportional to the magnitude of the residuals even when the absolute errors are larger than k. Conversely, the Tukey bisquare loss function[33] sets a threshold for the residuals, above which the value of the residuals does not influence the loss.

The Tukey loss function is given by

$$\rho_T = \begin{cases} \frac{k^2}{6}\left\{1 - \left[1 - \left(\frac{e}{k}\right)^2\right]^3\right\} & for\,|e| \le k \\ \frac{k^2}{6} & for\,|e| > k \end{cases} \tag{9}$$

ALGORITHM 1 Bounded CDO

```
Input: data stream $\mathbf{S}$; initial random design $\mathbf{X}$; warm-up length $m$; budget $B$
Output: an augmented design $\mathbf{Z}$
Set $\mathbf{C} = \emptyset$    // calibration set to estimate $\mathbf{\Sigma}, \Gamma_1, \Gamma_2$
$i \leftarrow 1, b \leftarrow 0$    // $b$ represents the currently used budget
while $i \leq m$ do
  Observe the $i$th data point $\mathbf{x}_i \in \mathbf{S}$
  Select $\mathbf{x}_i$ : $\mathbf{C} = \mathbf{C} \cup \mathbf{x}_i$
  $i \leftarrow i + 1$
end while
Estimate the covariance matrix $\mathbf{\Sigma}$ from $\mathbf{C}$ and perform eigendecomposition $\mathbf{\Sigma} = \mathbf{U\Lambda U}^{\mathrm{T}}$
Whiten the initial design by computing $\mathbf{Z} = \mathbf{\Lambda}^{-1/2} \mathbf{U}^{\mathrm{T}}\mathbf{X}$
Whiten the calibration set by computing $\mathbf{V} = \mathbf{\Lambda}^{-1/2} \mathbf{U}^{\mathrm{T}}\mathbf{C}$
Estimate $\Gamma_1, \Gamma_2$ by estimating the UPV of the model trained on $\mathbf{Z}$ on the points in $\mathbf{V}$
while $b \leq B$ and $i \leq |\mathbf{S}|$ do
  Observe the $i$th data point $\mathbf{x}_i \in \mathbf{S}$
  Whiten $\mathbf{x}_i$ by computing $\mathbf{z}_i = \mathbf{\Lambda}^{-1/2} \mathbf{U}^{\mathrm{T}}\mathbf{x}_i$
  if $\Gamma_1 \leq \mathbf{z}_i^{\mathrm{T}}(\mathbf{Z}^{\mathrm{T}}\mathbf{Z})^{-1}\mathbf{z}_i \leq \Gamma_2$ then
    Ask for the label $y_i$ and augment the labeled dataset $\mathbf{Z} = \mathbf{Z} \cup \mathbf{z}_i$
    $b \leftarrow b + 1$
    Update thresholds $\Gamma_1, \Gamma_2$ using the augmented design
  else
    Discard $\mathbf{x}_i$
  $i \leftarrow i + 1$
  end if
end while
return $\mathbf{Z}$
```

where the value of the tuning constant k is usually set up to 4.685σ.[31] Besides using a Huber or Tukey loss to obtain a robust estimator, we consider the possibility of filtering out outliers while selecting the most informative observations from the data stream. To this extent, we propose an adaptation of the CDO algorithm, where instead of estimating a threshold, we define a bounded area of interest for the unscaled prediction variance of an observation as

$$\mathrm{P_D}\left(\Gamma_1 \leq \mathbf{x}_{l+1}^{\mathrm{T}}\left(\mathbf{X}_l^{\mathrm{T}}\mathbf{X}_l\right)^{-1}\mathbf{x}_{l+1} \leq \Gamma_2\right) = \alpha \tag{10}$$

This approach is hereinafter referred to as bounded CDO. The idea is coherent with the method proposed by Hoaglin and Welsch.[34,35] of considering as potential outliers observations for which $\mathbf{x}_i^{\mathrm{T}}(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{x}_i \geq 2p/n$ is verified. The filtering approach suggested by Hoaglin and Welsch is also used by Deldossi et al.,[19] in the offline scenario. Here, instead of opting for a fixed value for Γ_2, we use KDE with a Gaussian kernel to estimate Γ_1 and Γ_2. The upper limit Γ_2 is selected by determining a cut-off value c, which is related to the amount of protection against outliers that we would like to achieve. This value is a tuning constant similar to the k used by robust estimators and, when possible, should be selected by exploiting previous knowledge of the process. Given the cut-off value c and the sampling rate α, Γ_2 is given by the $100(1-c)\%$ percentile, and Γ_1 by the $100(1-c-\alpha)\%$ percentile. As anticipated in Section 2, the threshold estimation is based on a set of unlabeled data, which is also used to estimate the covariance matrix $\mathbf{\Sigma}$ and whitening the observations to remove dependencies and facilitate the estimation of $\hat{\boldsymbol{\beta}}$. At this stage, semi-supervised methods might also be considered to perform tasks like feature extraction and exploit all the information available in the unlabeled data.[21,36–39]

Algorithm 1 provides a detailed explanation of how to implement the bounded CDO strategy for online active learning in a fixed-budget setting. The strategy involves collecting new labels and incorporating them into the design until a specified budget constraint B is reached. In some cases, it might be beneficial to anticipate the stop of the active learning routine if the marginal improvement of the model is no longer significant.[40] Previous studies have proposed various stopping criteria to enhance the efficiency of data collection schemes based on active learning.[41–45] Appendix A explores how some of these approaches could be adapted to the regression framework. From a computational standpoint, the update of $\hat{\boldsymbol{\beta}}$ is done by means of a complete retraining each time a new labeled example is added to the design. However, if the data matrix becomes considerably large and the time required for model updates increases, one may opt to update the model and estimate new thresholds when a batch of new observations is collected, aligning with the principles of batch-mode active learning.[46] Additionally, incremental and recursive updating techniques can also be considered for improving computational efficiency.

The estimation of the UPV can be modified by taking into account the weight matrix obtained from the robust estimators. The weighted UPV (UPV_w) is estimated as follows

$$\text{UPV}_w\left(\mathbf{x}\right) = \mathbf{x}^{(m)\text{T}}\left(\mathbf{X}^{\text{T}}\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{x}^{(m)} \tag{11}$$

where $\mathbf{W}$ represents the weight matrix used to downweigh the influence of outliers in the estimation of the regression parameters.[31] Each element of the weight matrix $\mathbf{W}$ is a positive number that determines the weight given to each observation in the regression analysis. Larger weights correspond to observations with less outlier-like behavior, while smaller weights correspond to observations with more outlier-like behavior. The weight matrix $\mathbf{W}$ is a diagonal matrix, where each diagonal element corresponds to the weight assigned to a particular observation. In the case of an OLS estimator, we have $\mathbf{W} = \mathbf{I}_k$, as the weight given to each observation is not sensitive to the residual. In other words, $w_{LS}(e) = 1$, regardless of the specific residual observed. With a Huber estimator, $w_H(e) = 1$ if $|e| \leq k$ and $w_H(e) = k/|e|$ if $|e| > k$. Finally, with a Tukey model, $w_T\ (e) = 0$ if $|e| > k$ and to $w_T\ (e) = [1 - (e/k)^2]^2$ if $|e| \leq k$. Then, to select the most informative observations while seeking protection against outliers, instead of estimating a single threshold, we define a bounded area of interest for the unscaled prediction variance of an observation as follows

$$\text{P}_\text{D}\left(\Gamma_1 \leq \mathbf{x}_{l+1}^{\text{T}}\left(\mathbf{X}_l^{\text{T}}\mathbf{W}\mathbf{X}_l\right)^{-1}\mathbf{x}_{l+1} \leq \Gamma_2\right) = \alpha \tag{12}$$

4 | EXPERIMENTS

In the experiments, we evaluate the performance of the active learning strategies in four scenarios, according to the percentage of outliers affecting the data stream. We compare the bounded CDO strategy, coupled with OLS and robust estimators, to the norm-thresholding approach, standard CDO, and random sampling. When using random sampling, each time a new sample arrives, a number $r \sim U(0,\ 1)$ is generated and the data point is only selected if $r \geq 1 - \alpha$, where α represents the labeling or sampling rate. The sampling strategies based on the use of robust estimators select the most informative data points using the standard UPV, as in Equation (10). The results obtained with the weighted prediction variance, UPV_w, were very similar and are included in the Appendix B for completeness. All the approaches receive as input the same random design and then they iteratively collect labeled observations until the budget constraint B is met. The number of observations contained in the initial design is equal to $p + 2$, where p is the number of process variables. We analyzed both the case of the initial design being outliers-free and contaminated. The results assuming the presence of outliers also in the initial design are included in the Appendix C. For each simulated scenario, the ith observation for the process variables, here considered a row vector, is generated according to a joint multivariate normal distribution

$$\mathbf{x}_i \sim \mathcal{N}_p\left(\mathbf{0},\ \boldsymbol{\Sigma}_0\right) \tag{13}$$

where $\boldsymbol{\Sigma}_0$ is given by $\sigma_{\mathbf{x}}^2\mathbf{I}$. The corresponding response is obtained using

$$y_i = \mathbf{x}_i\ \boldsymbol{\beta} + \varepsilon_i,\ \text{where}\ \varepsilon_i \sim \mathcal{N}\left(0,\ \sigma_\varepsilon^2\right) \tag{14}$$

For normal data points, we used $\sigma_{\mathbf{x}} = \sigma_\varepsilon = 1$ for both input and output variables, and, for simulating outliers, we set $\sigma_{\mathbf{x}} = \sigma_\varepsilon = 3$. Moreover, for each of the true coefficients of the underlying model, we assumed $\beta \sim U(-5,\ 5)$ for normal data points and $\beta \sim U(10,\ 15)$ for outliers. As in Deldossi et al.,[19] the outliers are introduced in the data stream in the form of isolated covariate and concept shifts. That is, an anomalous data point is a point for which we have both a larger variation in the input space, and a different relationship with the corresponding response variable. In the simulated scenarios, outliers are randomly distributed in the data stream according to a pre-defined percentage describing the contamination

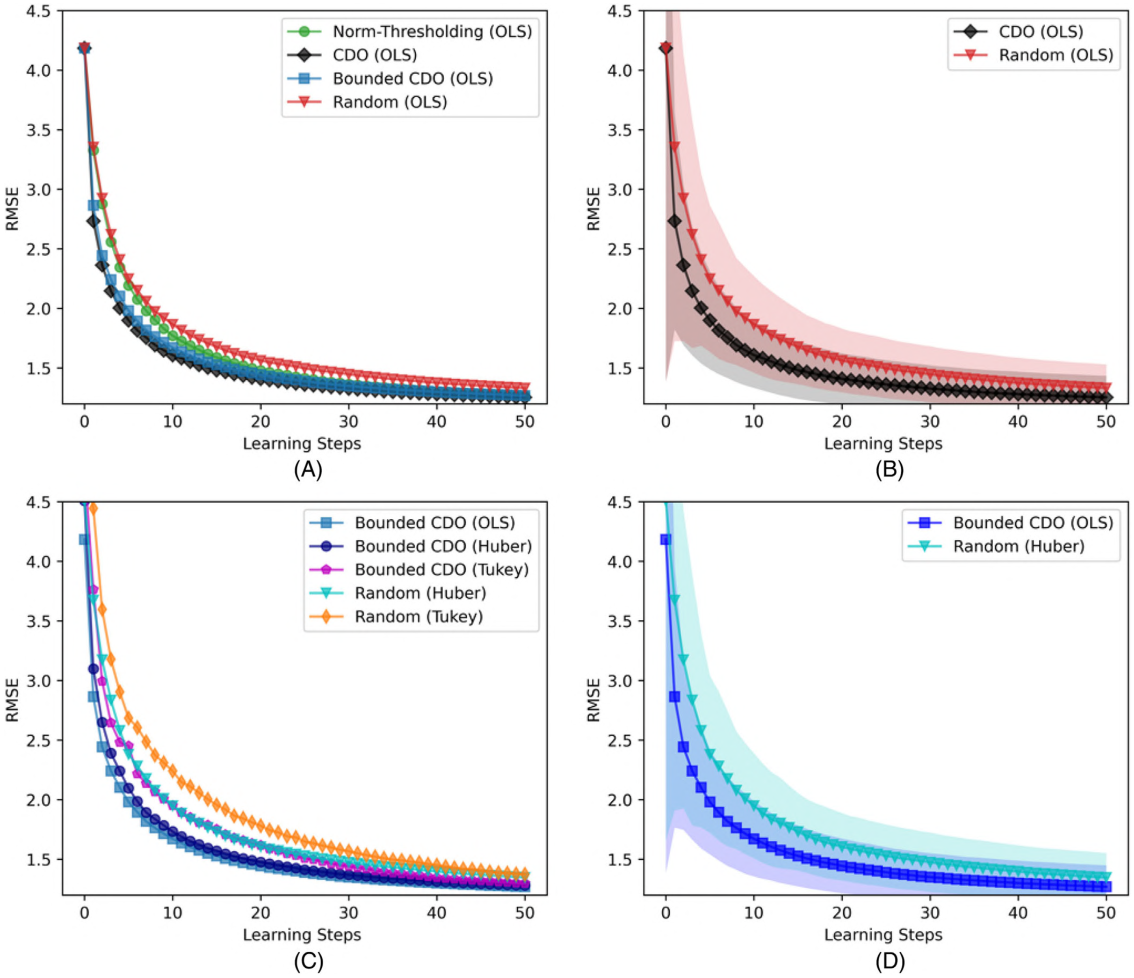


FIGURE 2 Comparing query strategies in the absence of outliers: results from 1000 simulations. Plots (B) and (D) offer a closer view of the two best strategies from plots (A) and (C), respectively, with shaded regions indicating the standard deviation across the simulations.

level of the environment. The performance of the models is expressed, in predictive terms, by the root mean squared error (RMSE) of the predictions on a separate test set, only composed of normal observations. This is coherent with the objective of trying to understand the true underlying relationship between predictors and response, and not the erroneous one that could be derived from the outliers.

The effectiveness of the proposed approach is evaluated by comparing the learning curves reporting the average RMSE values for each learning step, which are obtained using 1000 simulations for each scenario. A learning step indicates the acquisition of a new labeled observation and its inclusion in the training set. Hence, at each step, we are comparing models that are trained using the same number of labeled examples. We set the number of process variables equal to 20, the budget constraint B equal to 50, and the warm-up length m to 500. The warm-up length indicates the number of unlabeled observations that are used to estimate the covariance matrix $\mathbf{\Sigma}$ that is used for pre-whitening the observations. With regards to the sampling rate, we used $\alpha = 5\%$ for all the sampling strategies, and $c = 5\%$ for the protection cut-off value used by the bounded CDO algorithm. We selected 5% as it is a commonly employed value, especially when no previous specific knowledge is available.

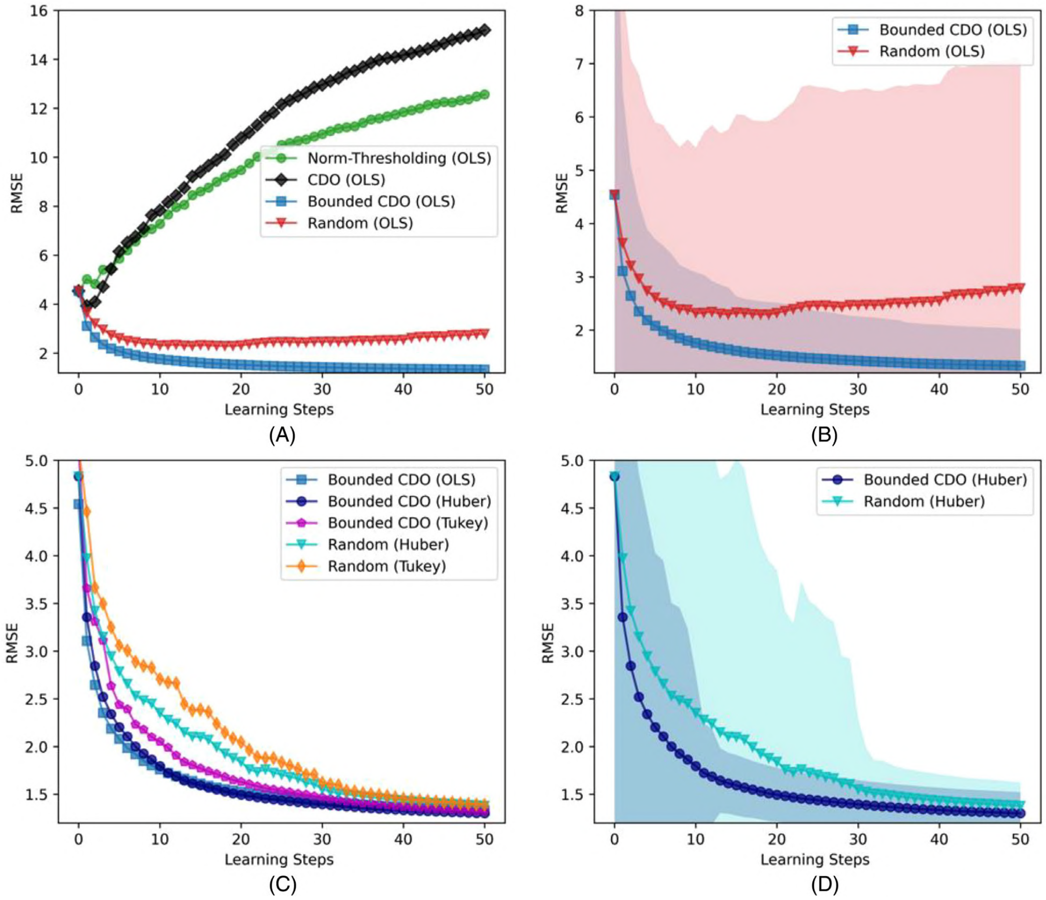


FIGURE 3 Comparing query strategies with 0.275% outliers (1000 simulations). Plots (B) and (D) offer a closer view of the two best strategies from plots (A) and (C), respectively, with shaded regions indicating the standard deviation across the simulations.

4.1 | No outliers

We first evaluated the query strategies to assess their performance in the absence of outliers. Consistently with the findings reported in our previous work,[16] our results in Figure 2 indicate that the standard CDO algorithm performs best when there are no outliers in the data stream. The use of robust estimators does not provide any added value in this scenario. Both the Huber and Tukey estimators are unable to outperform the bounded CDO strategy with the OLS model, which in turn is only marginally worse than the standard CDO. In Figure 2, plots (A) and (B) represent the strategies that rely on the OLS models, while plots (C) and (D) show the strategies that use robust models, with the bounded CDO based on OLS included for comparison.

4.2 | 0.275% outliers

The second scenario depicts a circumstance where only a modest fraction of the data stream is represented by outliers. We can see from the plot (A) of Figure 3 how the performance of the norm-thresholding and the CDO algorithm is dramatically worsened, as they are both prone to sample outliers. The random strategy seems to be a better option and the bounded CDO strategy offers the best results. In the plots (C) and (D) of the same figure, we can see the comparison with the

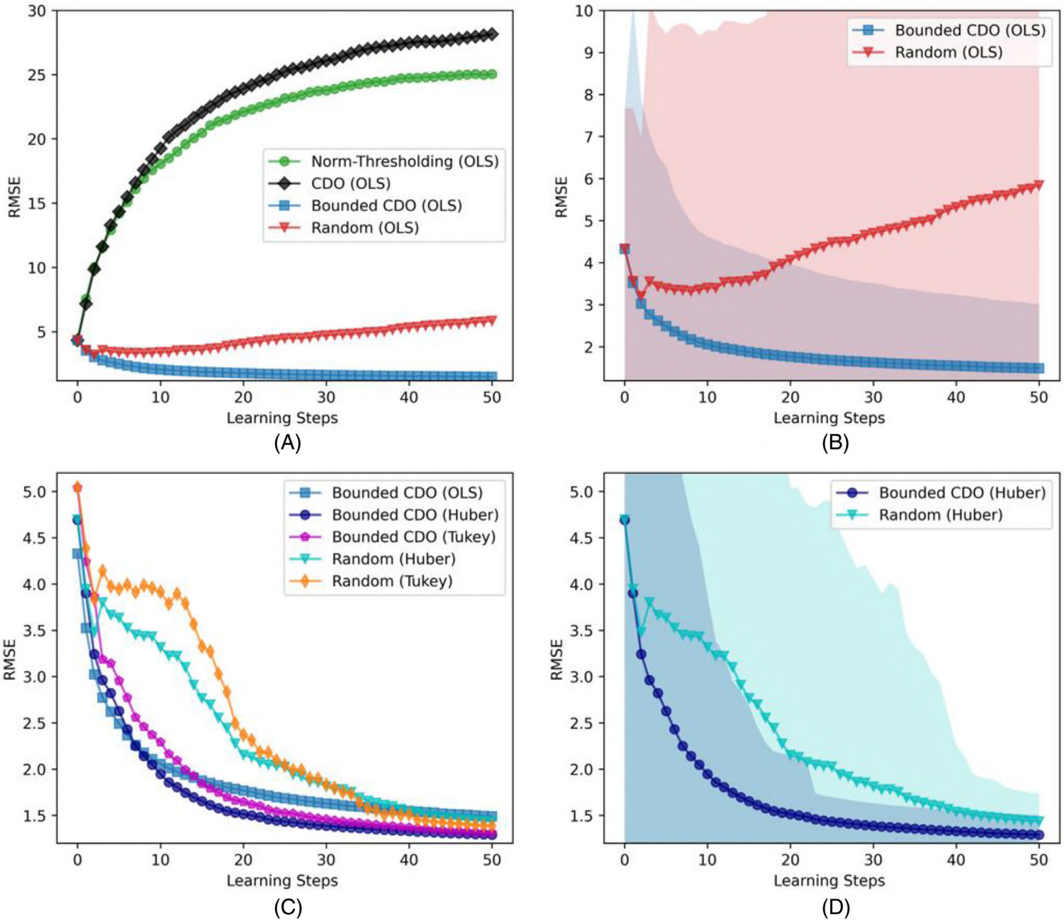


FIGURE 4 Comparing query strategies with 1% outliers (1000 simulations): results from 1000 simulations. Plots (B) and (D) offer a closer view of the two best strategies from plots (A) and (C), respectively, with shaded regions indicating the standard deviation across the simulations.

results obtained from the robust estimators. In this scenario, using a robust estimator does not seem to offer a significant improvement over the bounded CDO strategy based on OLS. Indeed, the learning curves obtained with the bounded strategy employing the OLS estimator and the Huber estimator are very similar.

4.3 | 1% outliers

The third scenario reports a worse situation, where the process is affected by a large number of outliers, that is, 1% of the total number of observations. The results in Figure 4 are similar to the ones from the previous scenario, with the exception that now the gap between bounded CDO and random sampling is much wider. This should be due to the fact that uniformly sampling observations with $\alpha = 5\%$ would most certainly lead to the inclusion of a greater number of outliers in the training set.

As per the robust estimators shown in the plots (C) and (D) of Figure 4, it is possible to see how the use of robust estimators now offers an evident value-added, also when compared to the OLS-based bounded CDO. While the learning curves are more or less overlapping in the first five learning steps, the models fitted using the Huber and Tukey losses are yielding a lower prediction error in the remaining steps.

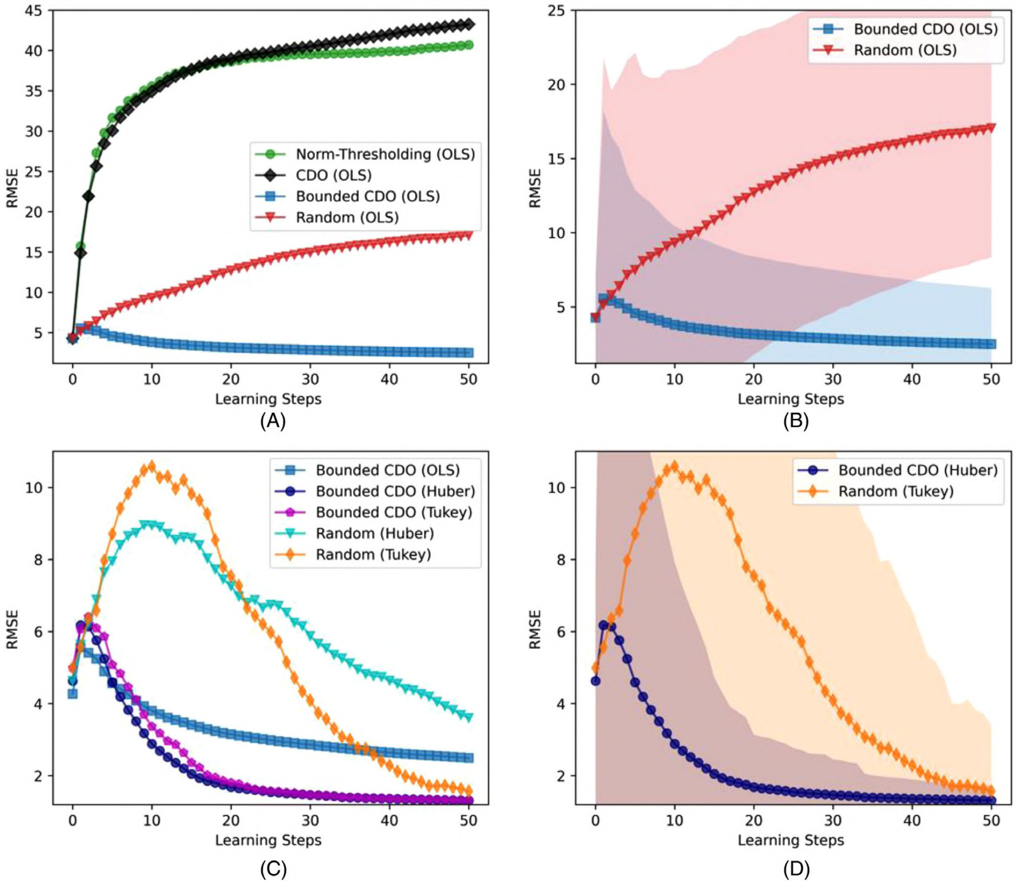


FIGURE 5 Comparing query strategies with 5% outliers (1000 simulations): results from 1000 simulations. Plots (B) and (D) offer a closer view on the two best strategies from plots (A) and (C), respectively, with shaded regions indicating the standard deviation across the simulations.

4.4 | 5% outliers

The final scenario simulates a pathological case, where 5% of the observations from the data stream are outliers. The results from the third scenario are exacerbated here. In the case of the OLS estimators, the bounded CDO is still the best strategy, being the only one with a descending learning curve (plots (A) and (B) of Figure 5). Instead, from the plots (C) and (D) of Figure 5 we can see how the robust estimators are able to improve the results obtained with the bounded CDO strategy. In this circumstance, there is not a clear distinction between the Huber and the Tukey models.

5 | DISCUSSION

The experiments presented in this study aimed to evaluate the performance of different active learning strategies in the presence of outliers in a data stream. The results showed that the standard CDO algorithm performed best in the absence of outliers, while the bounded CDO strategy coupled with OLS and robust estimators provided better results when outliers were present. In scenarios where an initial training set free from outliers is available and only a modest fraction of the data stream is represented by outliers, the bounded CDO strategy employing an OLS estimator seems to be the better option. Conversely, in the case of a larger contamination level, sampling strategies based on robust estimators yield the best results.

When using robust estimators, for our datasets we did not find solid evidence that using a weighted prediction variance is an advantage. Another interesting observation is that, in the presence of outliers, the standard OLS methods (random, norm-thresholding, and CDO) never converge to the results obtained with the robust query strategies. This is because they tend to accumulate outliers in the training set, which degrade the predictive performance as the model is not allowed to forget old or redundant data. The findings from this study have important consequences for practical applications of active learning strategies, especially in contexts where the data stream is contaminated by outliers. The results suggest that the choice of the active learning strategy should depend on the level of contamination of the data stream. When the data stream is free from outliers, the standard CDO is a good strategy. However, even when a modest fraction of the observations is corrupted, bounding the search area of the active learning algorithm or using robust estimators might be necessary. Overall, this study provides valuable insights into the performance of active learning strategies in the presence of outliers and can inform the development of more effective approaches for real-world applications. However, it is worth noting that the simulations were based on specific assumptions about the data generation process and may not fully capture the complexity of real-world data streams. Further research is needed to validate these findings on real-world datasets and to investigate the generalizability of the proposed approach.

6 | CONCLUSIONS

In many real-world problems, data is only available in an unlabeled form, and acquiring the labels is often an expensive and time-consuming task. In these circumstances, active learning is able to reduce the computational burden required to achieve compelling predictive performance by selecting the most informative data points to query. In this paper, we analyze the online active learning framework when the data stream is corrupted by the presence of outliers. In general, we show how the presence of outliers dramatically worsens the performance of the currently proposed methods for active linear regression. To tackle this issue, we propose a modification of the CDO algorithm that filters the outliers, while still focusing on the most promising observations based on the concepts of D-optimality and prediction variance. The analysis shows how this solution is sufficient to make the CDO strategy robust to a modest presence of outliers. When the percentage of outliers in the data stream is higher, the best results are obtained by coupling the bounded CDO strategy with a robust estimator. In general, the proposed approaches can effectively solve the problem of outliers contaminating the data stream, without adding computational complexity compared to the original CDO strategy.

ACKNOWLEDGMENTS

This work has been funded by the DTU Strategic Alliances Fund.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Davide Cacciarelli https://orcid.org/0000-0001-6664-9038
Murat Kulahci https://orcid.org/0000-0003-4222-9631
John Sølve Tyssedal https://orcid.org/0000-0003-1628-4725

REFERENCES

1. Kumar P, Gupta A. Active learning query strategies for classification, regression, and clustering: a survey. *J Comput Sci Technol*. 2020;35:913-945. doi:10.1007/s11390-020-9487-4
2. Settles B. Active learning literature survey. Technical Report 1648, University of Wisconsin-Madison Department of Computer Science. 2009.
3. Cacciarelli D, Kulahci M. A survey on online active learning. 2023. arXiv preprint 10.48550/arXiv.2302.08893
4. Chan LLT, Wu QY, Chen J. Dynamic soft sensors with active forward-update learning for selection of useful data from historical big database. *Chemom Intell Lab Syst*. 2018;175:87-103. doi:10.1016/j.chemolab.2018.01.015
5. Ge Z. Active learning strategy for smart soft sensor development under a small number of labeled data samples. *J Process Control*. 2014;24:1454-1461. doi:10.1016/j.jprocont.2014.06.015
6. Liu D, Zhang P, Zheng Q. An efficient online active learning algorithm for binary classification. *Pattern Recognit Lett*. 2015;68:22-26. doi:10.1016/j.patrec.2015.08.010

7. Bouguelia M-R, Belaïd Y, Belaïd A. An adaptive streaming active learning strategy based on instance weighting. *Pattern Recognit Lett*. 2016;70:38-44. doi:10.1016/j.patrec.2015.11.010
8. Lughofer E. Single-pass active learning with conflict and ignorance. *Evol Syst*. 2012;3:251-271. doi:10.1007/s12530-012-9060-7
9. Shan J, Zhang H, Liu W, Liu Q. Online active learning ensemble framework for drifted data streams. *IEEE Trans Neural Netw Learn Syst*. 2019;30:486-498. doi:10.1109/TNNLS.2018.2844332
10. Krawczyk B. Active and adaptive ensemble learning for online activity recognition from data streams. *Knowl Based Syst*. 2017;138:69-78. doi:10.1016/j.knosys.2017.09.032
11. Lughofer E. On-line active learning: a new paradigm to improve practical useability of data stream modeling methods. *Inf Sci (N Y)*. 2017;415-416:356-376. doi:10.1016/j.ins.2017.06.038
12. Lughofer E, Pratama M. Online active learning in data stream regression using uncertainty sampling based on evolving generalized fuzzy models. *IEEE Trans Fuzzy Syst*. 2018;26:292-309. doi:10.1109/TFUZZ.2017.2654504
13. Riquelme C, Ghavamzadeh M, Lazaric A. Active learning for accurate estimation of linear models. Proceedings of the 34th International Conference on Machine Learning; 2017.
14. Riquelme C, Johari R, Zhang B. Online active linear regression via thresholding. Thirty-First AAAI Conference on Artificial Intelligence; 2017. www.aaai.org
15. Fontaine X, Perrault P, Valko M, Perchet V. Online a-optimal design and active linear regression. Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021.
16. Cacciarelli D, Kulahci M, Tyssedal JS. Stream-based active learning with linear models. *Knowl Based Syst*. 2022;254:109664. doi:10.1016/j.knosys.2022.109664
17. Melis DA, Jaakkola T. Towards robust interpretability with self-explaining neural networks. In: Bengio S, Wallach H, Larochelle H, Grauman K, Cesa-Bianchi N, Garnett R, eds. *Adv Neural Inf Process Syst*. Curran Associates, Inc; 2018. https://proceedings.neurips.cc/paper/2018/file/3e9f0fc9b2f89e043bc6233994dfcf76-Paper.pdf
18. Efron B, Hastie T, Johnstone I, Tibshirani R. Least angle regression. *Ann Stat*. 2004;32. doi:10.1214/009053604000000067
19. Deldossi L, Pesce E, Tommasi C. A sub-sampling algorithm preventing outliers. 2022. arXiv preprint http://arxiv.org/abs/2208.06218
20. Zhao J, Yi S, Liang Y, Liu W, Cao X. Robust active representation via ℓ2,p-norm constraints[Formula presented]. *Knowl Based Syst*. 2022;235:107639. doi:10.1016/j.knosys.2021.107639
21. He G, Li Y, Zhao W. An uncertainty and density based active semi-supervised learning scheme for positive unlabeled multivariate time series classification. *Knowl Based Syst*. 2017;124:80-92. doi:10.1016/j.knosys.2017.03.004
22. Cacciarelli D, Kulahci M. A novel fault detection and diagnosis approach based on orthogonal autoencoders. *Comput Chem Eng*. 2022;163:107853. doi:10.1016/j.compchemeng.2022.107853
23. Nguyen QP, Lim KW, Divakaran DM, Low KH, Chan MC. GEE: A gradient-based explainable variational autoencoder for network anomaly detection. IEEE Conference on Communications and Network Security (CNS); 2019. doi:10.1109/CNS.2019.8802833
24. Zhou C, Paffenroth RC, Autoencoders ADRD. Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA; 2017. doi:10.1145/3097983.3098052
25. Ruff L, Kauffmann JR, Vandermeulen RA, et al. A unifying review of deep and shallow anomaly detection. *Proc IEEE*. 2021;109:756-795. doi:10.1109/JPROC.2021.3052449
26. Qiu C, Li A, Kloft M, Rudolph M, Mandt S. Latent outlier exposure for anomaly detection with contaminated data. Proceedings of the 39th International Conference on Machine Learning, PMLR 162; 2022. http://arxiv.org/abs/2202.08088
27. Burbidge R, Rowland JJ, King RD. Active learning for regression based on query by committee. Intelligent Data Engineering and Automated Learning; 2007:209-218.
28. Ge Z. Active probabilistic sample selection for intelligent soft sensing of industrial processes. *Chemom Intell Lab Syst*. 2016;151:181-189. doi:10.1016/j.chemolab.2016.01.003
29. st John RC, Draper NR. D-Optimality for regression designs: a review. *Technometrics*. 1975;17:15-23. doi:10.1080/00401706.1975.10489266
30. Myers RH, Montgomery D, Anderson-Cook CM. Response surface methodology: process and product optimization using designed experiments. Wiley Series in Probability and Statistics 2016. ISBN: 978-1-118-91601-8
31. Fox J, Weisberg S. Robust regression. An R and S-Plus Companion to Applied Regression. 2013.
32. Huber PJ. Robust estimation of a location parameter. *Ann Math Statist*. 1964;35:73-101.
33. Beaton AE, Tukey JW. The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data. *Technometrics*. 1974;16:147-185. doi:10.1080/00401706.1974.10489171
34. Hoaglin DC, Welsch RE. The hat matrix in regression and ANOVA. *Am Stat*. 1978;32:17. doi:10.2307/2683469
35. Chatterjee S, Hadi AS. Influential observations, high leverage points, and outliers in linear regression. *Stat Sci*. 1986;1(3):379-393.
36. Fernandes MC, Covões TF, Pereira ALV. Improving evolutionary constrained clustering using Active Learning. *Knowl Based Syst*. 2020;209:106452. doi:10.1016/j.knosys.2020.106452
37. Leng Y, Xu X, Qi G. Combining active learning and semi-supervised learning to construct SVM classifier. *Knowl Based Syst*. 2013;44:121-131. doi:10.1016/j.knosys.2013.01.032
38. Frumosu FD, Kulahci M. Big data analytics using semi-supervised learning methods. *Qual Reliab Eng Int*. 2018;34:1413-1423. doi:10.1002/qre.2338
39. Cacciarelli D, Kulahci M, Tyssedal J. Online active learning for soft sensor development using semi-supervised autoencoders. *ICML 2022 Workshop on Adaptive Experimental Design and Active Learning in the Real World*. 2022. https://arxiv.org/abs/2212.13067

40. Pullar-Strecker Z, Dost K, Frank E, Wicker J. Hitting the target: stopping active learning at the cost-based optimum. *Mach Learn*. 2022. doi:10.1007/s10994-022-06253-1
41. Zhang Y, Cai W, Wang W, Zhang Y. Stopping criterion for active learning with model stability. *ACM Trans Intell Syst Technol*. 2017;9:1-26. doi:10.1145/3125645
42. Ishibashi H, Hino H. Stopping criterion for active learning based on deterministic generalization bounds. Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS), PMLR 108; 2020.
43. Ghayoomi M. Using variance as a stopping criterion for active learning of frame assignment. Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing. 2010:1-9.
44. Laws F, Schütze H, Schütze S. Stopping criteria for active learning of named entity recognition. Proceedings of the 22nd International Conference on Computational Linguistics. 2008;108:465-472
45. Zhu J, Wang H, Hovy E. Multi-criteria-based strategy to stop active learning for data annotation. Proceedings of the 22nd International Conference on Computational Linguistics. 2008;108:1129-1136.
46. Ren P, Xiao Y, Chang X, et al. A survey of deep active learning. *ACM Comput Surv*. 2022;54:1-40. doi:10.1145/3472291
47. Bloodgood M, Vijay-Shanker K. A method for stopping active learning based on stabilizing predictions and the need for user-adjustable stopping. Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL), 2009:39-47.
48. Tomanek K, Hahn U. Approximating learning curves for active-learning-driven annotation, n.d. http://www.ncbi.nlm.nih.gov/
49. Farquhar S, Gal Y, Rainforth T. On statistical bias in active learning: how and when to fix it. International Conference on Learning Representations (ICLR); 2021.

How to cite this article: Cacciarelli D, Kulahci M, Tyssedal JS. Robust online active learning. *Qual Reliab Engng Int*. 2023;1-20. https://doi.org/10.1002/qre.3392

APPENDIX A: STOPPING CRITERION

In real-world applications of active learning, if we do not have an explicit operational budget on the number of experiments that can be run, it can be challenging to determine when to stop collecting new labels due to the unavailability of the true learning curves. To address this problem, it is beneficial to approximate the learning curve using proxy measures. In this study, we investigate the use of two proxy measures. Firstly, we propose monitoring the slope of the stabilization score, drawing inspiration from the stabilizing predictions[47] and validation set agreement[48] methods employed in classification. In the regression framework, we calculate the stabilization of predictions by averaging the sum of squares of the differences between the predictions of the w most recent pairs of models. Similarly to Bloodgood and Vijay-Shanker,[47] we utilize a window size of 3 ($w = 3$). The values being compared are the predicted values of the calibration set **C**, obtained through successive models. As the examples in **C** are not used in the annotation process, this curve is solely influenced by the impact of selected and labeled examples on training new models. Essentially, this curve monitors when the predictions from models trained with newly included observations start producing highly similar results. The stopping rule can then be determined through visual inspection of the curve, by setting a tolerance for the sum of squares not improving or

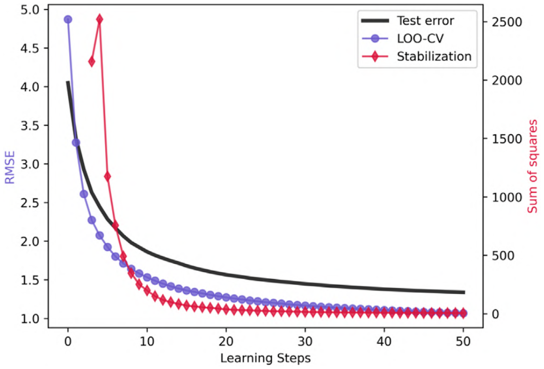


FIGURE 6 Approximating the learning curve: random sampling with no outliers (1000 simulations). The left axis reports the RMSE value for the curves related to the test error and the LOO-CV. The right axis shows the average sum of squares related to the stabilization score.

approaching zero, or by applying a hypothesis testing procedure. Another performance-based metric we consider is the leave-one-out cross-validation (LOO-CV) score obtained by the model on the currently available labeled observations. While this technique relies on ground-truth labels and may appear advantageous, it may not be the optimal choice if the collected training set is biased or does not accurately represent the real data distribution.[49] On the other hand, the stabilization score, despite not relying on real labels, could be more reliable if the calibration set **C** follows the population distribution. Figure 6 demonstrates the effectiveness of the two proposed methods in approximating the true test error curve, offering valuable insights for determining when to halt the active learning routine.

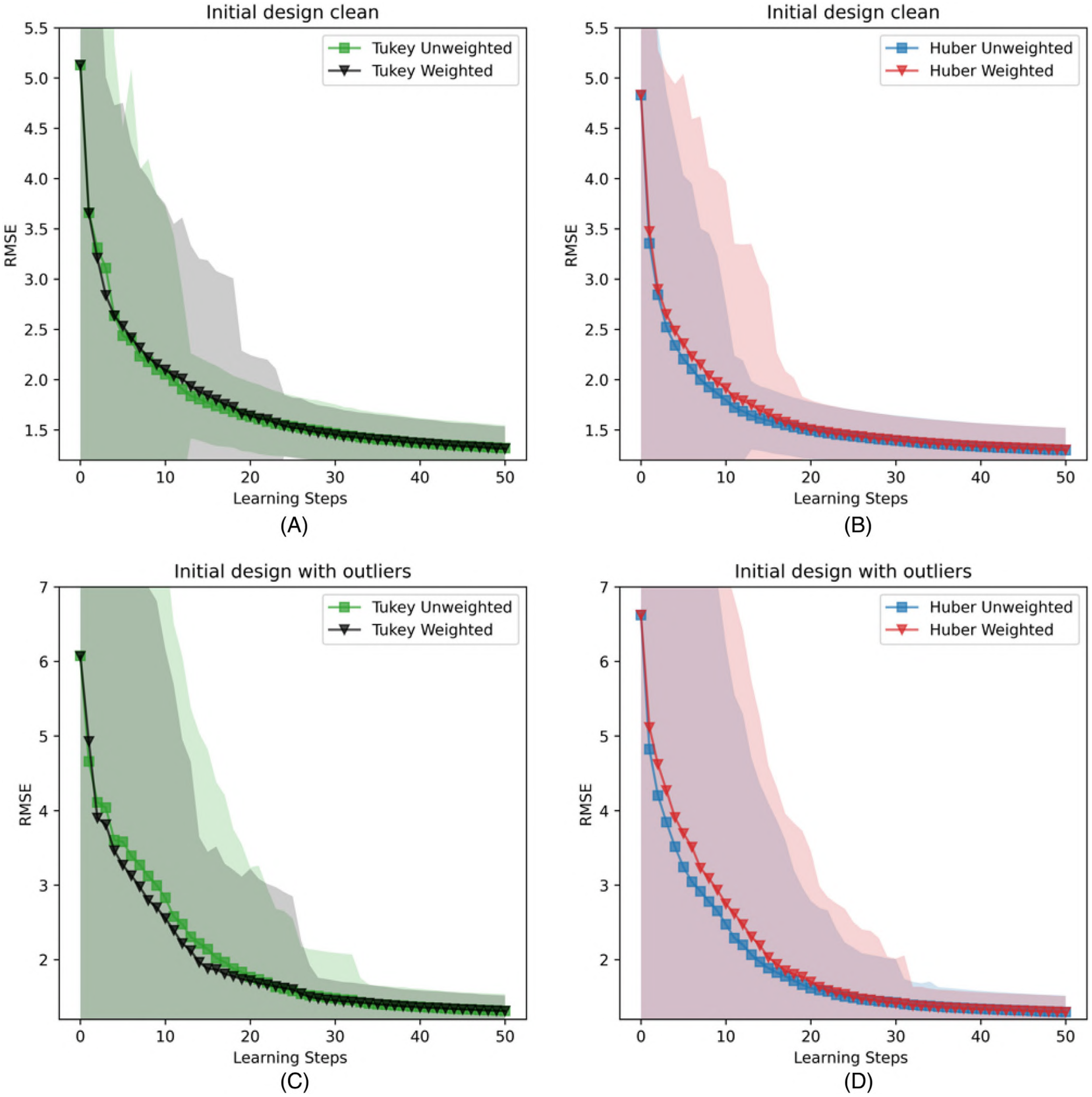


FIGURE 7 Comparing UPV and UPV$_w$ in the scenario with 0.275% outliers (1000 simulations).

APPENDIX B: WEIGHTED PREDICTION VARIANCE

In this section, we examine the impact of switching from the standard UPV to its weighted version on the learning curves of the robust bounded CDO strategies. While it may seem reasonable to use a weighted prediction variance from a theoretical standpoint, we found little compelling evidence that it improves performance even with the use of robust estimators (Figures 7–9). In fact, we observed that using the UPV$_w$ actually worsens results when the initial design is free from outliers. This could be because the robust models mistakenly identify some observations as outliers, resulting in $\mathbf{W} \neq \mathbf{I}_k$.

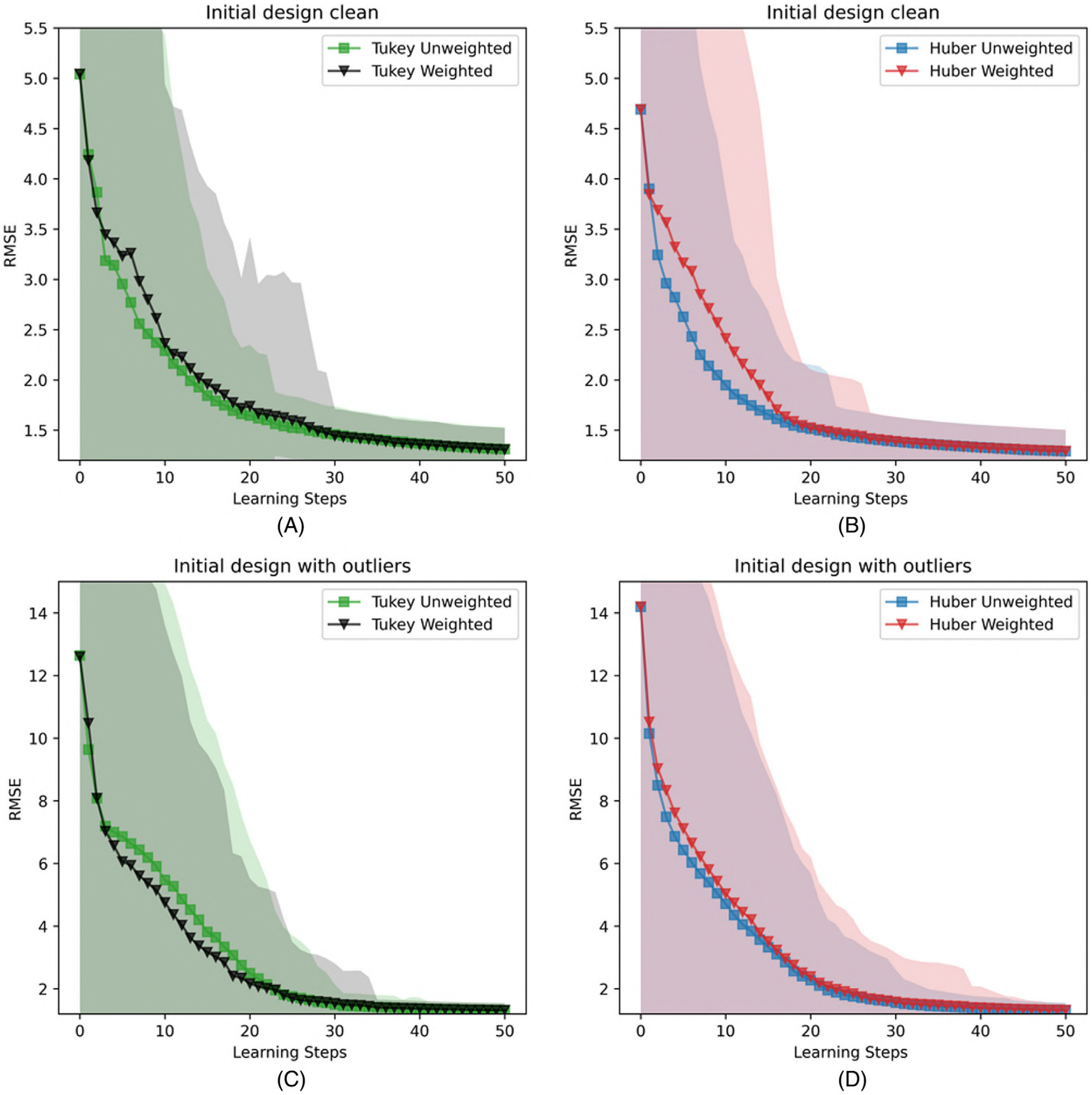


FIGURE 8 Comparing UPV and UPV$_w$ in the scenario with 1% outliers (1000 simulations).

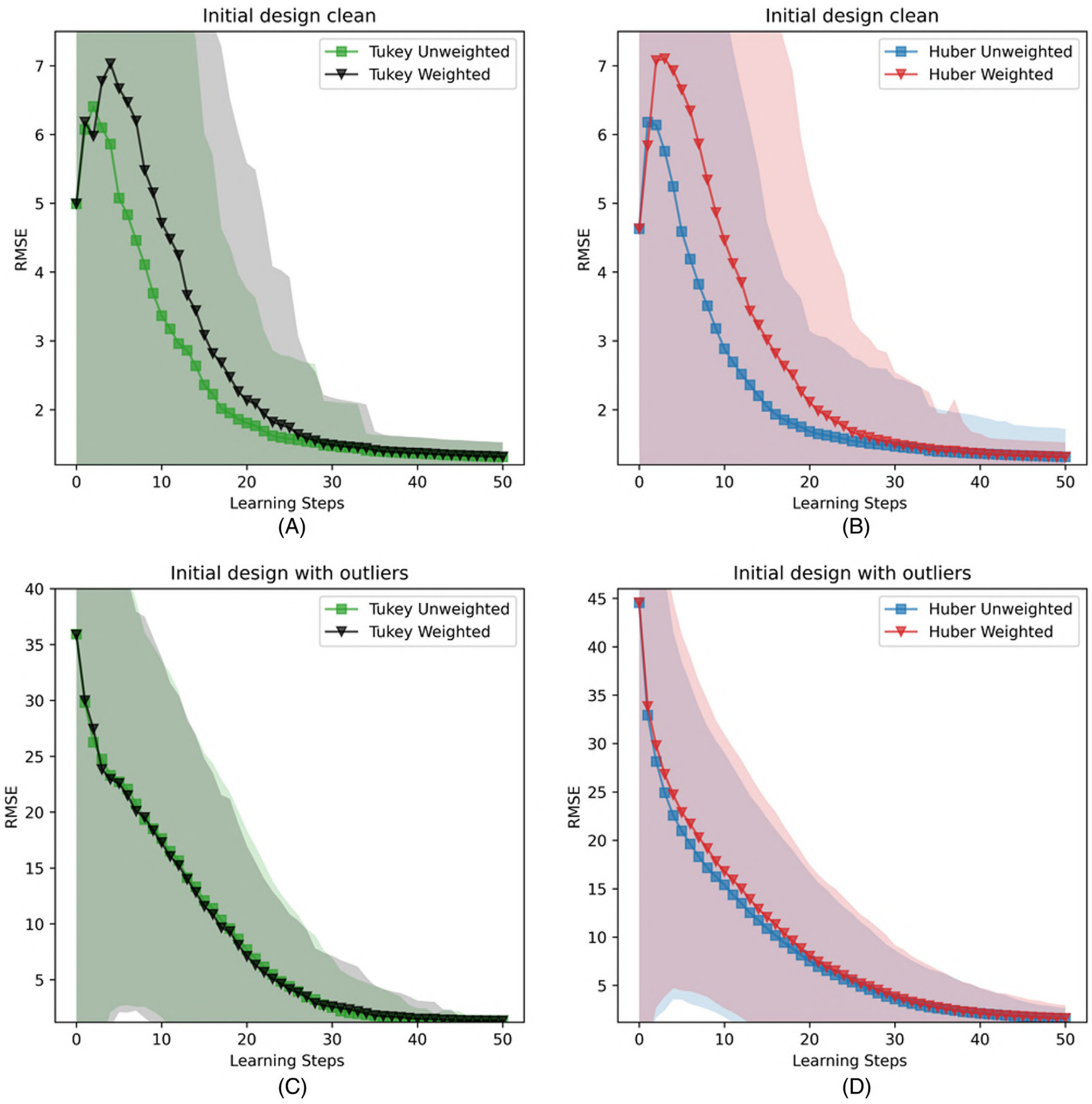


FIGURE 9 Comparing UPV and UPV$_w$ in the scenario with 5% outliers (1000 simulations).

APPENDIX C: PRESENCE OF OUTLIERS IN THE INITIAL DESIGN

In Figures 10–12, we investigate the impact of removing the assumption that the initial design is free from outliers on the sampling strategies. Despite the small size of the initial design when $p = 20$, we observed several notable behaviors. One of the most noticeable differences is that the learning curves start with higher errors, as there are outliers forcibly included in the data. However, over time, the learning curves of the robust strategies are able to converge to satisfactory predictive performance as they can minimize the impact of these observations on the model training. In contrast, the OLS-based bounded CDO performs significantly worse in this scenario. This is because estimating the cutoff value Γ_2 using a contaminated set does not provide adequate protection against the inclusion of outliers in the design.

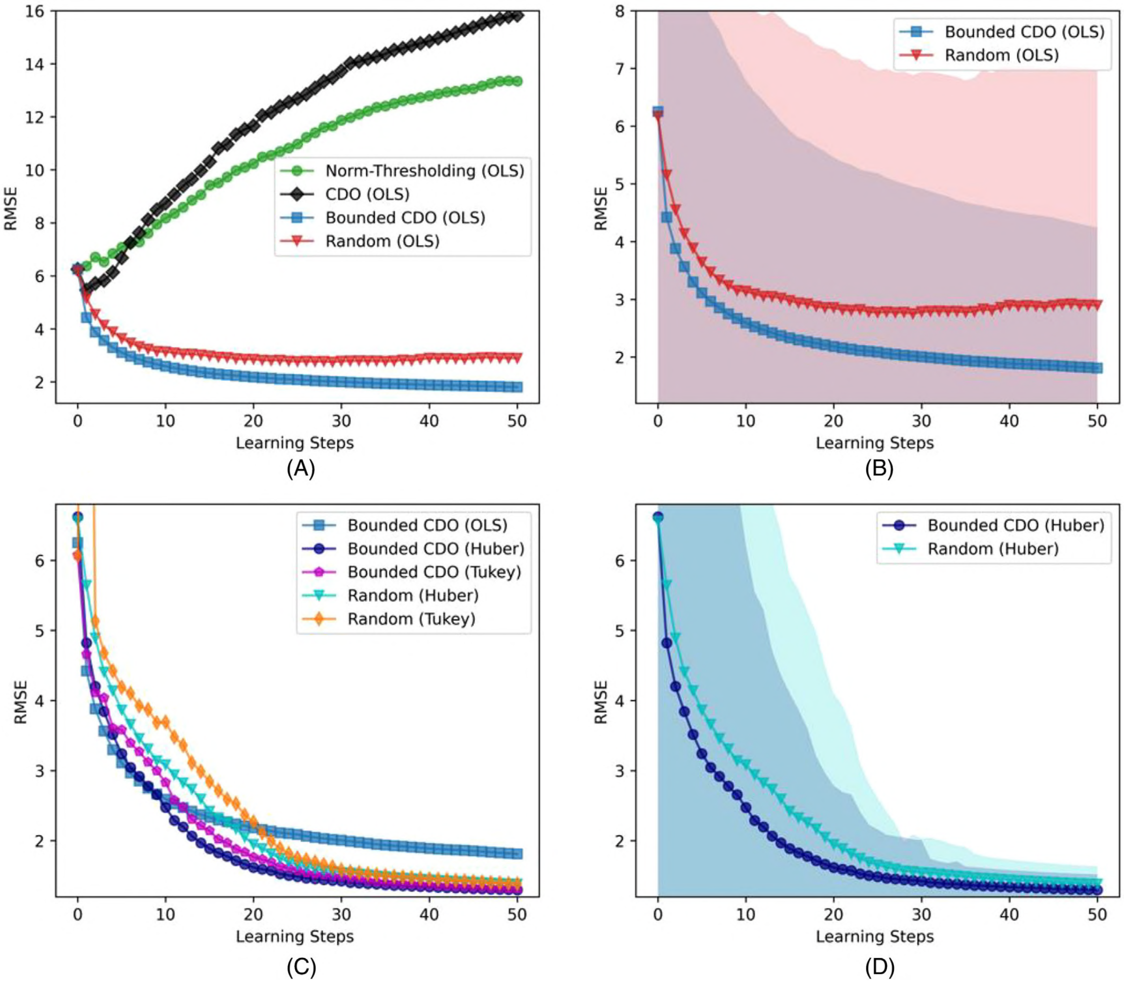


FIGURE 10 Comparing query strategies with 0.275% outliers (1000 simulations): results from 1000 simulations. Plots (B) and (D) offer a closer view of the two best strategies from plots (A) and (C), respectively, with shaded regions indicating the standard deviation across the simulations.

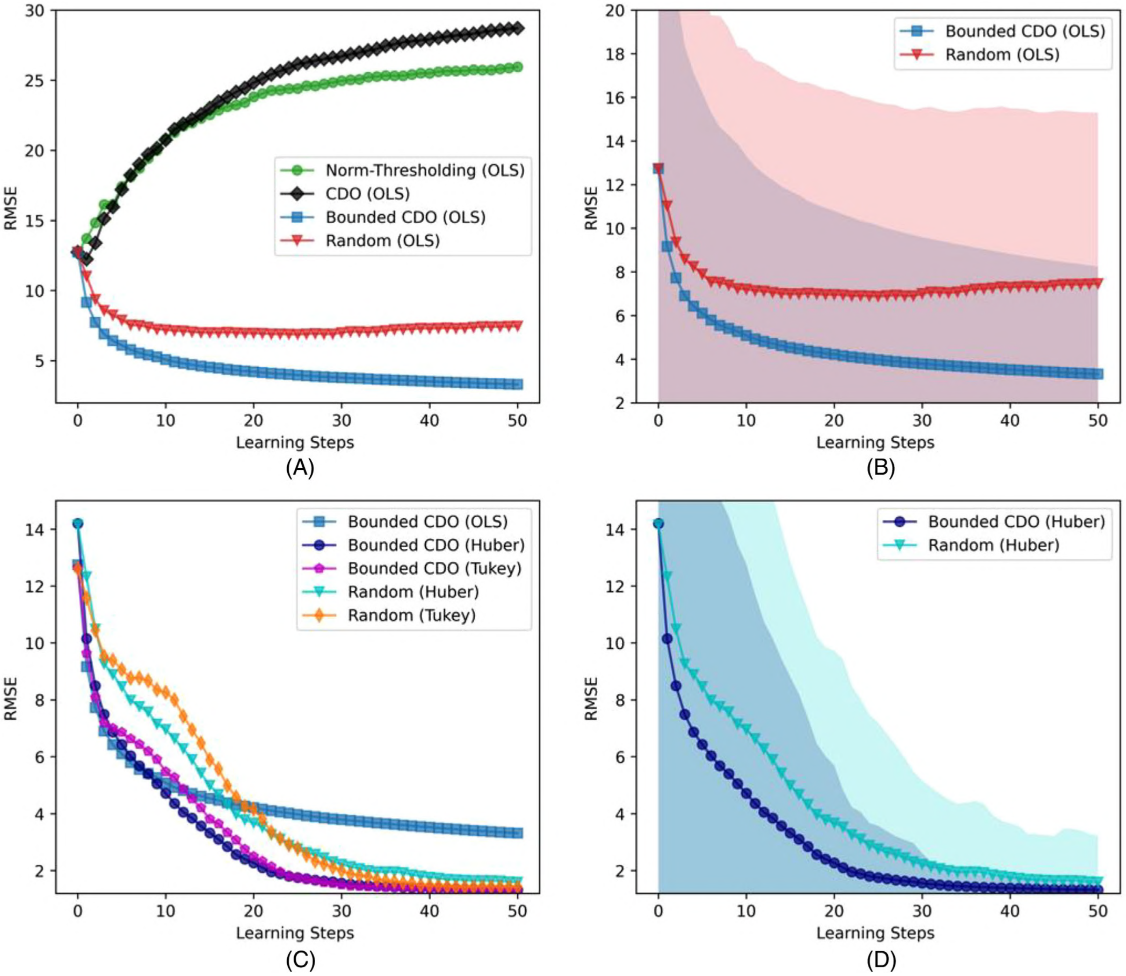


FIGURE 11 Comparing query strategies with 1% outliers (1000 simulations): results from 1000 simulations. Plots (B) and (D) offer a closer view of the two best strategies from plots (A) and (C), respectively, with shaded regions indicating the standard deviation across the simulations.

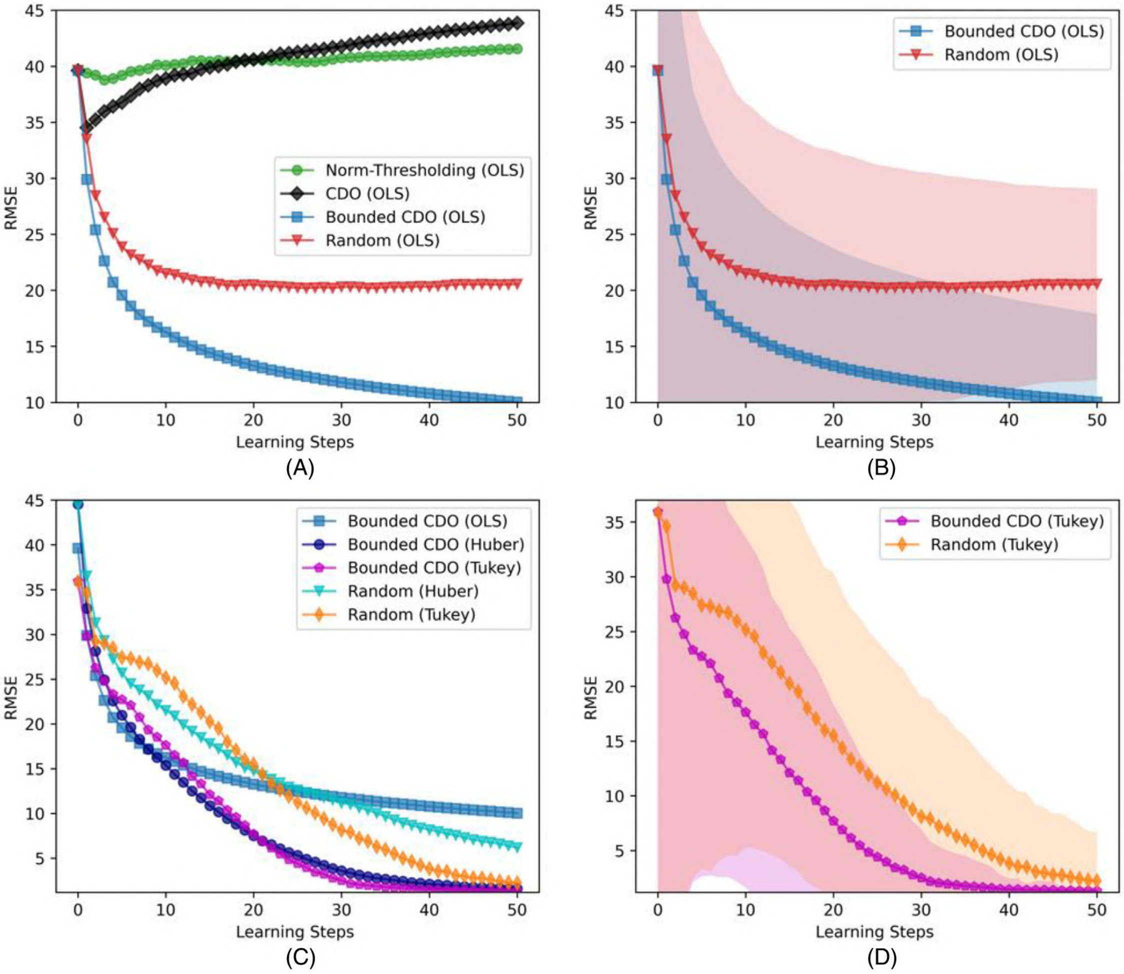


FIGURE 12 Comparing query strategies with 5% outliers (1000 simulations): results from 1000 simulations. Plots (B) and (D) offer a closer view of the two best strategies from plots (A) and (C), respectively, with shaded regions indicating the standard deviation across the simulations.

AUTHOR BIOGRAPHIES

Davide Cacciarelli is a PhD student at the Technical University of Denmark and the Norwegian University of Science and Technology. His research is related to active learning and statistical process monitoring.

Murat Kulahci is a Professor at the Technical University of Denmark and Luleå University of Technology. His research focuses on the design of physical and computer experiments, statistical process monitoring, time series analysis and forecasting, and financial engineering.

John Sølve Tyssedal is a Professor at the Norwegian University of Science and Technology. His research interests include design of experiments, statistical process control and time series analysis.

APPENDIX E

PAPER 5 – Stream-based active learning for regression with dynamic feature selection

Stream-Based Active Learning for Regression with Dynamic Feature Selection

Davide Cacciarelli*,†, John Sølve Tyssedal†, Murat Kulahci*,‡

*Technical University of Denmark, Department of Applied Mathematics and Computer Science
†Norwegian University of Science and Technology, Department of Mathematical Sciences
‡Luleå University of Technology, Department of Business Administration, Technology and Social Sciences

***Abstract*—In the era of big data, companies are increasingly driven to amass vast amounts of data, particularly in process industries where advanced sensor technologies are prevalent. However, obtaining accurate labels or product information through quality inspections can be prohibitively expensive. Active learning emerges as a promising approach to optimize data sampling by prioritizing the most informative data points. Nevertheless, active learning strategies heavily rely on predictive models that are iteratively updated. Aligning with the principles of data-centric AI, this study highlights the detrimental effects of passively incorporating all available process variables into a predictive model for guiding data collection. Specifically, in real-time sampling strategies based on online active learning, the inclusion of irrelevant features significantly hampers the efficiency of the learning process.**

***Index Terms*—Data-centric AI, active learning, unlabeled data, data streams, feature selection, design of experiments.**

I. Introduction

The ubiquity of big data extends to industrial production, where automated data collection schemes based on pervasive sensors often lead to a flood of process data. Many industries tend to collect vast amounts of data, frequently overlooking its relevance for modeling and predictive objectives. In contrast, product-related data, especially in high-volume manufacturing settings, is usually scarce due to the costs involved with quality inspections. This dichotomy can introduce complexities in creating predictive models that link process variables to product features. Hence, active learning, due to its ability to propose the most informative data points for labeling, is increasingly embraced as it promotes a data-efficient methodology for model training and deployment [1]. In high-volume, fast-paced production, it is not always feasible to evaluate all available instances prior to decision-making. Stream-based active learning, where data arrives in a stream and the learner must promptly decide whether to keep, label, or discard each instance, proves valuable in such scenarios [2]–[5]. Despite the significant research on active learning for classification tasks, regression models, which are fundamental for developing soft sensors or addressing quality control issues where a product characteristic is measured on a continuous scale, have received less attention [6]. This paper explores the application of the stream-based active learning framework for linear regression models, with an emphasis on scenarios involving the presence of irrelevant features in the data. Considering the relevance of data, from its initial collection to eventual use, enhances our understanding of the underlying process and contributes to the broader discourse on better big data practices in industrial environments.

II. Stream-Based Active Learning

By maintaining a representative subset of the data, active learning alleviates the computational demands that come with large datasets, employing a range of query strategies to accomplish this. These strategies either aim to enhance the model by selecting data points where the model shows uncertainty [7]–[10], or they work to secure a diverse and representative subset within the feature space [11]–[13]. It should be noted that in the data-centric AI paradigm, the concept of data representativity emerges as a critical aspect that transcends computational efficiency. It takes on an even more significant role, acting as a countermeasure against biases and promoting fairness during the development and deployment of AI systems [14].

In the stream-based active learning framework for linear regression, also referred to as online active linear regression [15], observations arrive in a sequential manner, often in real-time, necessitating instantaneous decisions about their relevance. The labeled data acquired from the stream are used to fit a linear model, described by $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, where $\mathbf{y}$ is the response variable vector, $\mathbf{X}$ is the model matrix, $\boldsymbol{\beta}$ is the vector of regression coefficients, and $\boldsymbol{\varepsilon}$ is the error vector. According to the norm-thresholding approach introduced by Riquelme et al. [16], when a new unlabeled data point $\mathbf{x}$ is observed, the learner asks for its label only if its norm exceeds a threshold Γ, defined by

$$P(\|\mathbf{x}\| \geq \Gamma) = \alpha \tag{1}$$

where α is the percentage of observations we are willing to label out of the incoming data stream. This value should be set taking into account the budget B and the sampling rate of the sensors placed along the process. The conditional D-optimality (CDO) approach [17], [18] further extends the norm-thresholding strategy by imposing a threshold Γ on the prediction variance as in

Corresponding author: Davide Cacciarelli (dcac@dtu.dk).

979-8-3503-5801-8/23/$31.00 ©2023 IEEE
DOI 10.1109/TransAI60598.2023.00030

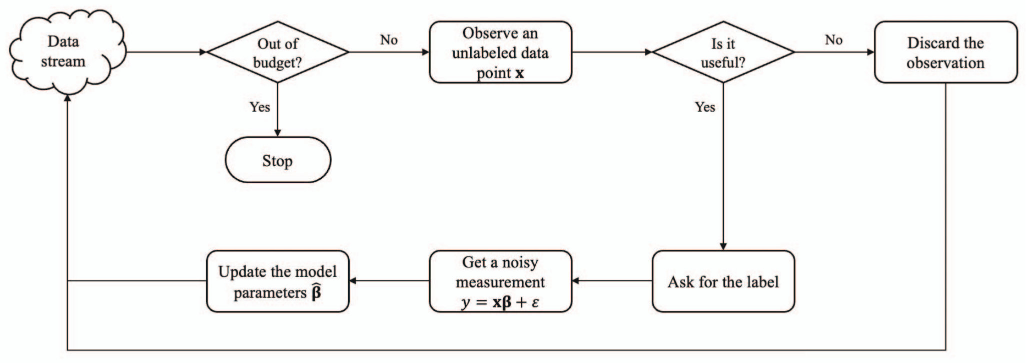

Data stream
Out of budget?
No
Yes
Stop
Observe an unlabeled data point $\mathbf{x}$
Is it useful?
No
Yes
Discard the observation
Ask for the label
Get a noisy measurement $y = \mathbf{x\beta} + \varepsilon$
Update the model parameters $\widehat{\boldsymbol{\beta}}$

Fig. 1. Stream-based active learning for regression in the fixed-budget setting.

$$P\left(\mathbf{x}_{k+1}^{\top}\left(\mathbf{X}_k^{\top}\mathbf{X}_k\right)^{-1}\mathbf{x}_{k+1} \geq \Gamma\right) = \alpha \tag{2}$$

where $\mathbf{X}_k$ is the model matrix including the k labeled observations collected to this point. This approach is inspired by the connection between D-optimality and prediction variance [19] and tries to combine accurate parameter estimation with the exploration of lesser known input space regions.

Fig. 1 illustrates the generic stream-based active learning flowchart for linear regression. In this setup, observations are evaluated on the fly to determine their inclusion in model updates. In the fixed-budget formulation of active learning, sampling continues until a predefined budget constraint on the number of data points that can be queried is met.

III. Feature Selection

In situations where data is being collected from high-dimensional sources in real-time, the relevance of features becomes particularly critical as the data may contain numerous irrelevant features. These features can pose serious challenges to the efficiency of stream-based active learning strategies, as they can not only negatively impact the performance of predictive models but also significantly inflate the data storage requirements. In light of these considerations, our approach integrates feature selection techniques into the stream-based active learning process. This allows for a more refined understanding of the data structure, enhancing the predictive power of the model and making the sampling process more efficient. Indeed, the adoption of appropriate feature selection techniques in the context of active learning can alleviate the curse of dimensionality and significantly improve the quality of data sampling, making the best out of a limited labeling budget [20]–[22]. Feature selection can broadly be divided into three main categories: information-theoretical methods, sparse methods, and statistical methods [23]. Within the realm of information-theoretical methods [24], mutual information (MI) is a commonly used technique. The MI score between two random variables measures the information gained about one variable through observing the other. For a pair of continuous variables (X, Y), defined over $\mathcal{X} \times \mathcal{Y}$, the MI is calculated as

$$I(X;Y) = \int_{\mathcal{Y}}\int_{\mathcal{X}} P_{X,Y}(x,y)\log\left(\frac{P_{X,Y}(x,y)}{P_X(x)P_Y(y)}\right)dxdy \tag{3}$$

where $P_{X,Y}$ is the joint probability density function of X and Y, and $P_X(x)$ and $P_Y(y)$ are the marginal probability density functions of X and Y, respectively. Among the sparse methods, the least absolute shrinkage and selection operator (Lasso) [25] has been widely adopted. Lasso achieves regularization and variable selection by limiting the sum of the absolute values of the model parameters. Considering the linear model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ introduced in Section II, the optimization problem set by Lasso can be formulated as

$$\min_{\boldsymbol{\beta}}\left\{\frac{1}{2n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1\right\} \tag{4}$$

where n is the number of observations and λ is a non-negative tuning parameter controlling the level of shrinkage. This method facilitates the selection of important features by pushing the coefficients of irrelevant ones toward zero. Sparse methods find extensive use in chemometrics, particularly when the number of predictors p greatly exceeds n, as enforcing sparsity promotes model interpretability and mitigates overfitting to the training data [26]

Finally, statistical methods can be used to assess the significance of features based on various statistical tests. F-tests are commonly employed in regression analysis for this purpose. One of the simplest approaches is to perform univariate tests to evaluate the effect of each regressor.

IV. Approach

The main objective of this work is to evaluate the performance of online active linear regression methods when many irrelevant measurements are taken from the process. To do so, we combine the CDO stream-based active learning strategy with three different feature selection methods. In this framework, the learner observes a full measurement vector of p process variables at each time step. After deciding whether to query this vector, the learner acquires the corresponding label, updating both the regression model and the feature scores. These feature scores are used to select the M most relevant features, thereby reducing the number of parameters to be estimated. Alg. 1 presents the modified version of the CDO algorithm that incorporates feature selection. The key modification in the proposed algorithm, compared to the original CDO algorithm, is the inclusion of feature scoring in addition to observing the data point and deciding whether to request its label. It should be noted that this algorithm provides the general framework to implement CDO with feature selection. The step where the M features are retained from the measurement vector and the data matrix varies according to the specific feature selection strategy chosen.

The threshold Γ is estimated using kernel density estimation on am unlabeled warm-up set that is collected by observing the process for a while without querying any label. The threshold is then iteratively refined in order to represent the most recent model. For more details related to the threshold estimation and the computational time required to update it, please refer to [17].

Algorithm 1 CDO with Feature Selection

```
Require: Initial training set $(\mathbf{X}, \mathbf{y})$ and model $\widehat{\boldsymbol{\beta}}$; data stream
    $\mathbf{S}$; sampling rate $\alpha$; budget $B$; threshold $\Gamma$; initialized
    feature score function $\mathcal{F}$ and maximum number of features
    $M$.
    $i \leftarrow 1$                                      ▷ Timestamp
    $c \leftarrow 0$                                      ▷ Labeling cost
    while $c \leq B$ and $i \leq |\mathbf{S}|$ do
        Observe the $i$th data point $\mathbf{x}_i \in \mathbf{S}$
        Retain top $M$ features from $\mathbf{x}_i$ and $\mathbf{X}$
        Predict response $\widehat{y} = \mathbf{x}\widehat{\boldsymbol{\beta}}$
        if $\mathbf{x}_i^\top \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{x}_i \geq \Gamma$ then
            Ask for the true label $y_i$
            Update training set $(\mathbf{X}, \mathbf{y}) = (\mathbf{X}, \mathbf{y}) \cup (\mathbf{x}_i, y_i)$
            Update model $\widehat{\boldsymbol{\beta}}$
            Update feature score function $\mathcal{F}$
            Update threshold $\Gamma$
            $c \leftarrow c + 1$                          ▷ Pay for the label
        else
            Discard $\mathbf{x}_i$
        end if
        $i \leftarrow i + 1$
    end while
```

In this work, model updates entail a complete retraining process, where a new estimation of the regression coefficients, $\widehat{\boldsymbol{\beta}}$, is obtained from scratch using the augmented training set. Alternatively, incremental training approaches that update the model gradually by incorporating new data points in small batches or one at a time could be considered [27].

V. Application

In our experiments, we explore the performance of the stream-based active learning strategies in three scenarios. In each scenario, the data is generated from a linear model where only 5 features impact the response, but 10, 20, and 30 irrelevant features are added to the data stream, respectively. This simulates real-world situations where numerous sensors are employed to collect process variables, but only a few of them have a real effect on the response. We assume that each measurement of the p process variables follows a multivariate Normal distribution

$$\mathbf{x}_i \sim \mathcal{N}_p\left(\mathbf{0}, \boldsymbol{\Sigma}\right) \tag{5}$$

where $\boldsymbol{\Sigma}$ is the covariance matrix, given by $\sigma_{\mathbf{x}}^2\mathbf{I}$. The corresponding response is obtained using

$$y_i = \mathbf{x}_i\boldsymbol{\beta} + \varepsilon_i, \quad \text{where } \varepsilon_i \sim \mathcal{N}\left(0, \sigma_\varepsilon^2\right) \tag{6}$$

We compare the performance of the CDO strategy with the norm-thresholding approach and random sampling. Our primary objective is to demonstrate the performance degradation experienced by traditional methods when irrelevant feature measurements are included in the data stream. Additionally, we investigate the effectiveness of three feature selection methods (information-theoretic, sparse, and statistical) in enhancing the identification of truly relevant features, within the CDO framework. For the Lasso estimator, the parameter λ is found with a five-fold cross-validation procedure. For the MI and F-test feature selection strategies, we retained the five top features, assuming to know the number of process variables truly affecting the y. Random sampling is implemented by drawing a random number $r \sim \mathcal{U}(0, 1)$ at each iteration, and each incoming data point is selected if $r \geq 1 - \alpha$. Moreover, we included the performance obtained with CDO using the optimal subset of features, assuming a perfect knowledge about the irrelevant features.

The learning curves for the three case studies are reported in Fig. 2. As expected, the gap between the performance obtained using the optimal subset and the sampling strategies gets larger as the number of irrelevant features increases. We can see how across the three case studies the best performance is obtained by combining CDO with Lasso. Using univariate F-tests does not offer significant advantages during the learning process. Finally, using the MI score in the feature selection process dramatically deteriorates the model. Figs. 3,4, and 5 can be used to get some insights on the behaviors of these feature selectors. As expected, the Lasso estimator is able to identify immediately the five relevant features by shrinking the coefficients of the irrelevant features to zero. Using F-tests we are able to locate the relevant features. However, there is a significant delay in identifying the proper features. Finally, while the MI heatmaps seem to highlight the relevant features, we can see a clear smearing effect that leads to including irrelevant features, at the expense of the relevant ones, especially at the beginning of the process. We believe

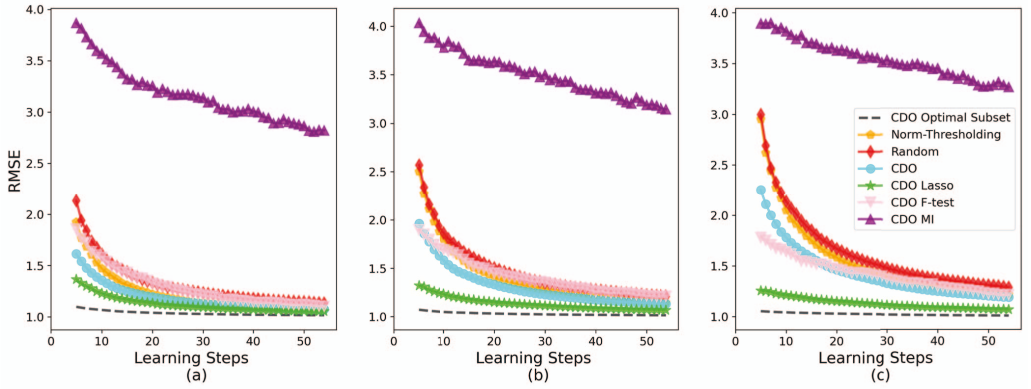


Fig. 2. Learning curves of the different Active Learning strategies (average from 500 simulation runs). Subplots (a), (b), and (c) show the cases with 10, 20, and 30 irrelevant features, respectively.

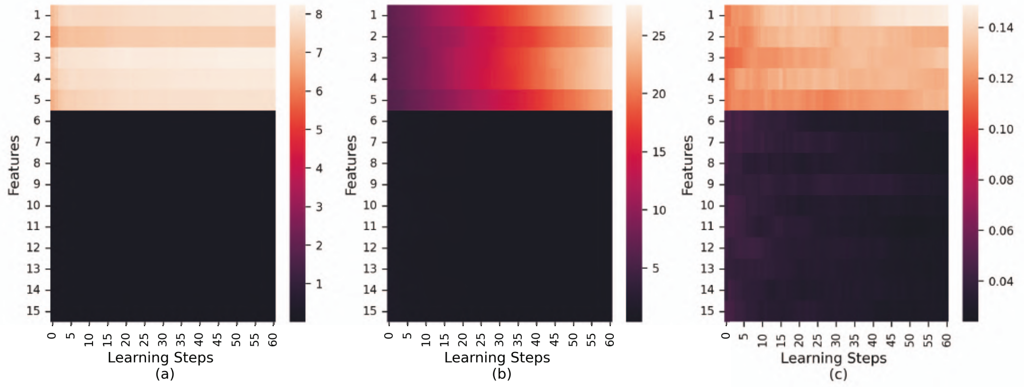


Fig. 3. Scores obtained with the different feature selection methods during the Active Learning routine for the case with 10 irrelevant features (average from 500 simulation runs). Subplot (a) shows the squared regression coefficients obtained with Lasso. Subplot (b) shows the feature importance scores obtained by performing univariate F-tests. Subplot (c) shows the Mutual Information scores.

this is due to the fact that in this case the MI scores are not reliable when only a few number of observations have been collected.

VI. Conclusion

This paper has highlighted the implications of including irrelevant variables in the learning process, and established the benefits of combining feature selection techniques with active learning. Existing stream-based active learning techniques for regression, including the norm-thresholding and the CDO methods, showed decreased efficiency and effectiveness when a large number of irrelevant features were included in the data stream. The learning curves, which illustrate the evolution of model error over time, clearly showcased an increased prediction error when the proportion of irrelevant features in the data was high. This result reinforces the idea that the mindless incorporation of all available variables can be detrimental to the learning process, supporting the data-centric AI philosophy that emphasizes the relevance and quality of data, rather than sheer quantity.

This study considered three feature selection methods: an information-theoretical approach based on MI, a sparse method employing the Lasso technique, and a statistical method using F-tests. In the experiments, we showed how the use of a Lasso estimator offered a more efficient and streamlined learning process. Particularly, this method helped to alleviate the issue of the curse of dimensionality while improving the quality of data sampling. In doing so, we underscored the power of feature selection in harnessing the potential of a limited labeling budget, contributing to the broader goal

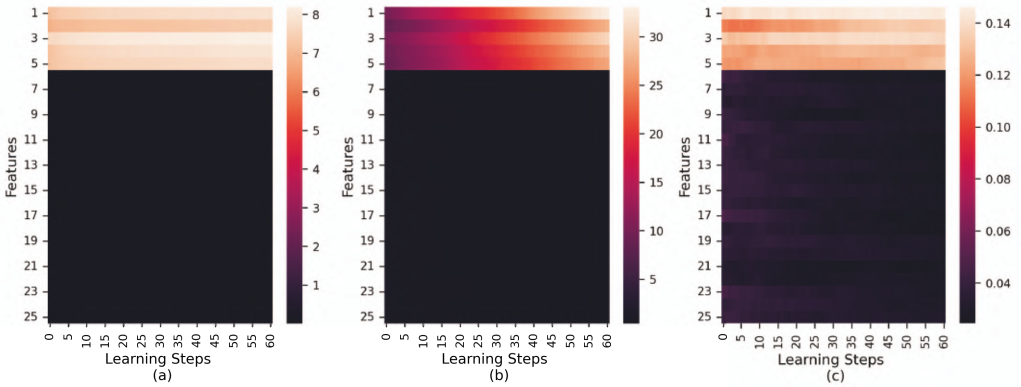


Fig. 4. Scores obtained with the different feature selection methods during the Active Learning routine for the case with 20 irrelevant features (average from 500 simulation runs). Subplot (a) shows the squared regression coefficients obtained with Lasso. Subplot (b) shows the feature importance scores obtained by performing univariate F-tests. Subplot (c) shows the Mutual Information scores.

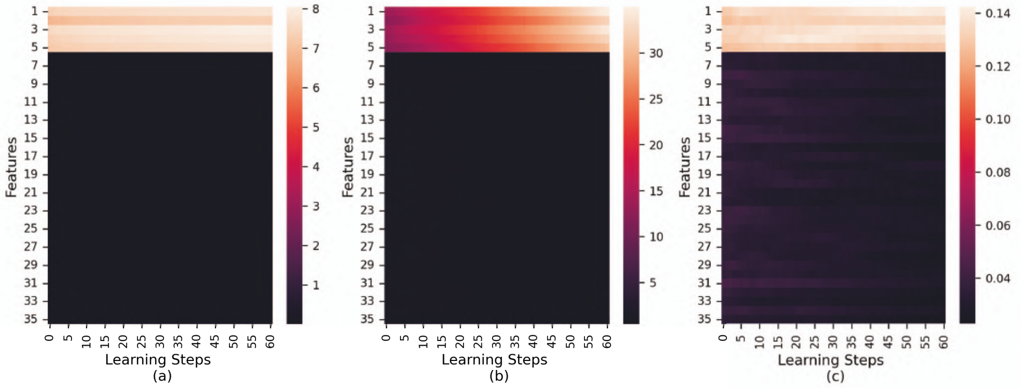


Fig. 5. Scores obtained with the different feature selection methods during the Active Learning routine for the case with 30 irrelevant features (average from 500 simulation runs). Subplot (a) shows the squared regression coefficients obtained with Lasso. Subplot (b) shows the feature importance scores obtained by performing univariate F-tests. Subplot (c) shows the Mutual Information scores.

of optimizing the use of big data in industrial contexts and increasing the knowledge about the underlying process. It is also important to note that the specific feature selection method used should ideally be tailored to the given context and the nature of the data. In this case, we limited the experimental setup to the case where the collected process variables are not correlated. While the Lasso method proves highly effective in this study, it might not be suitable for situations where features are highly correlated [28]. Specifically, if a block of process variables exhibits strong correlation, the application of Lasso might result in the random selection of a single variable within that block. Rotari and Kulahci [28] recently proposed a variable selection wrapper for random forests to reliably estimate feature importance scores in the presence of correlations. Random forests are easily interpretable models that can offer straightforward feature importance scores using metrics such as the Gini index [29]. Moreover, dimensionality reduction methods like principal component analysis or autoencoders could prove beneficial [30], [31] in extracting salient features from the process variables. However, in the case of p independent process variables, using dimensionality reduction techniques could be counterproductive, as it could hinder the isolation of the irrelevant features. This suggests that the choice of the feature selection method should carefully take into account the particular traits of the data and the specific needs of the task we are dealing with. Moreover, prior knowledge about the process or offline screening experiments should be leveraged to inform the feature selection procedure, whenever possible.

In summary, this study has shown that combining active learning and feature selection can result in enhanced efficiency and accuracy for modeling regression data streams. Future research could explore more complex or hybrid feature selection methods, as well as investigate the use of these methods with different types of learning models, beyond linear ones. Additionally, evaluating the effectiveness of this approach on real-world datasets from diverse industries would provide valuable insights. Ultimately, this work aimed to inspire further exploration and innovation in the field of data-centric AI, fostering smarter, more efficient, and responsible data utilization practices.

REFERENCES

[1] B. Settles, "Active learning literature survey," *Technical Report 1648, University of Wisconsin-Madison Department of Computer Sciences*, 2009.

[2] E. Lughofer, "On-line active learning: A new paradigm to improve practical useability of data stream modeling methods," *Information Sciences*, vol. 415-416, pp. 356–376, 11 2017.

[3] D. Cacciarelli and M. Kulahci, "A survey on online active learning," ArXiv Preprint, Tech. Rep., 2023.

[4] D. Manjah, D. Cacciarelli, B. Standaert, M. Benkedadra, G. Rotsart, S. Galland, B. Macq, and C. D. Vleeschouwer, "Stream-based active distillation for scalable model deployment," *Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition (CVPR) Workshops*, 2023.

[5] D. Cacciarelli, M. Kulahci, and J. Tyssedal, "Online active learning for soft sensor development using semi-supervised autoencoders," *ICML 2022 Workshop on Adaptive Experimental Design and Active Learning in the Real World*, 12 2022.

[6] W. Cai, Y. Zhang, and J. Zhou, "Maximizing expected model change for active learning in regression," *Proceedings - IEEE International Conference on Data Mining, ICDM*, pp. 51–60, 2013.

[7] J. Lu, P. Zhao, and S. C. H. Hoi, "Online passive-aggressive active learning," *Machine Learning*, vol. 103, pp. 141–183, 5 2016.

[8] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *The Journal of Machine Learning Research*, vol. 2, 2002.

[9] D. Roth and K. Small, "Margin-based active learning for structured output spaces," *Machine Learning: ECML 2006*, pp. 413–424, 2006.

[10] M.-F. Balcan, A. Broder, and T. Zhang, "Margin based active learning," *COLT - 23th Conference on Learning Theory*, vol. 4739, 2007.

[11] H. T. Nguyen and A. Smeulders, "Active learning using pre-clustering," *Proceedings of the twenty-first international conference on Machine learning*, 2004.

[12] F. Min, S. M. Zhang, D. Ciucci, and M. Wang, "Three-way active learning through clustering selection," *International Journal of Machine Learning and Cybernetics*, vol. 11, pp. 1033–1046, 5 2020.

[13] D. Ienco, A. Bifet, Zliobaite, and B. Pfahringer, "Clustering based active learning for evolving data streams," *16th International Conference on Discovery Science*, 2013.

[14] L. H. Clemmensen and R. D. Kjærsgaard, "Data representativity for machine learning and ai systems," *arXiv preprint arXiv:2203.04706*, 2022.

[15] X. Fontaine, P. Perrault, M. Valko, and V. Perchet, "Online a-optimal design and active linear regression," 2021.

[16] C. Riquelme, R. Johari, and B. Zhang, "Online active linear regression via thresholding," *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[17] D. Cacciarelli, M. Kulahci, and J. S. Tyssedal, "Stream-based active learning with linear models," *Knowledge-Based Systems*, vol. 254, p. 109664, 10 2022.

[18] ——, "Robust online active learning," *Quality and Reliability Engineering International*, 2023.

[19] R. H. Myers, D. Montgomery, and C. M. Anderson-Cook, *Response surface methodology: process and product optimization using designed experiments*. Wiley, 2016.

[20] C. Li, X. Wang, W. Dong, J. Yan, Q. Liu, and H. Zha, "Joint active learning with feature selection via cur matrix decomposition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 6, pp. 1382–1396, 2018.

[21] L. Zhang, "Data-driven building energy modeling with feature selection and active learning for data predictive control," *Energy and Buildings*, vol. 252, p. 111436, 2021.

[22] M. Okabe and S. Yamada, "Interactive spam filtering with active learning and feature selection," in *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, vol. 3. IEEE, 2008, pp. 165–168.

[23] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM Comput. Surv.*, vol. 50, no. 6, dec 2017.

[24] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, 2012.

[25] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[26] L. Clemmensen, T. Hastie, D. Witten, and B. Ersbøll, "Sparse discriminant analysis," *Technometrics*, vol. 53, no. 4, pp. 406–413, 2011.

[27] S. Vijayakumar, A. D'souza, and S. Schaal, "Incremental online learning in high dimensions," *Neural computation*, vol. 17, no. 12, pp. 2602–2634, 2005.

[28] M. Rotari and M. Kulahci, "Variable selection wrapper in presence of correlated input variables for random forest models," *Quality and Reliability Engineering International*, 2023.

[29] D. Cacciarelli and M. Boresta, "What drives a donor? a machine learning-based approach for predicting responses of nonprofit direct marketing campaigns," *International Journal of Nonprofit and Voluntary Sector Marketing*, 8 2021.

[30] D. Cacciarelli and M. Kulahci, "Hidden dimensions of the data: Pca vs autoencoders," *Quality Engineering*, pp. 1–10, 2023.

[31] ——, "A novel fault detection and diagnosis approach based on orthogonal autoencoders," *Computers & Chemical Engineering*, vol. 163, p. 107853, 7 2022.

APPENDIX F

Paper 6 – Stream-based active distillation for scalable model deployment

Stream-Based Active Distillation for Scalable Model Deployment

Dani MANJAH[1] Davide CACCIARELLI[2] Mohamed BENKEDADRA[3]
Baptiste STANDAERT[1] Gauthier ROTSART DE HERTAING[1] Benoît MACQ[1]
Stéphane GALLAND[4] and Christophe DE VLEESCHOUWER[1]
[1]Université catholique de Louvain [2] Technical University of Denmark
[3] Université de Mons [4] Université de technologie de Belfort Montbéliard
{dani.manjah, baptiste.standaert, gauthier.rotsart}@uclouvain.be
dcac@dtu.dk mohamed.benkedadra@umons.ac.be

Abstract

This paper proposes a scalable technique for developing lightweight yet powerful models for object detection in videos using self-training with knowledge distillation. This approach involves training a compact student model using pseudo-labels generated by a computationally complex but generic teacher model, which can help to reduce the need for massive amounts of data and computational power. However, model-based annotations in large-scale applications may propagate errors or biases. To address these issues, our paper introduces Stream-Based Active Distillation (SBAD) to endow pre-trained students with effective and efficient fine-tuning methods that are robust to teacher imperfections. The proposed pipeline: (i) adapts a pre-trained student model to a specific use case, based on a set of frames whose pseudo-labels are predicted by the teacher, and (ii) selects on-the-fly, along a streamed video, the images that should be considered to fine-tune the student model. Various selection strategies are compared, demonstrating: 1) the effectiveness of implementing distillation with pseudo-labels, and 2) the importance of selecting images for which the pre-trained student detects with a high confidence.

1. Introduction

Deep Neural Networks (DNNs) are effective for object detection in images, but their predictive power comes at a high cost. The training of highly performant DNNs is based on high-performance cloud servers with a large-scale data set. This requires (i) a large workforce to prepare the data set or implementation of training (ii) as well as a significant investment in time and money. These data, time, and hardware costs create a barrier for most practitioners in terms of transition from theory to practice [5]. Furthermore, a single investment in resources to create large general-purpose models, regardless of their size, is no longer sufficient. Without retraining, these models cannot be robust with respect to the **stochastic** and **ever-evolving** environments. In the example of Closed-Circuit Television (CCTV) monitoring traffic on the city scale, there is no data set large enough to cover all aspects of every urban landscape [35]. Therefore, a *scalable, efficient, and recurrent* retraining is necessary to reduce costs and avoid **under-performing** systems.

Knowledge Distillation (KD) is a promising technique that enables the creation of lightweight but powerful models. The process assumes that for the same data set, large models (that is, *teachers*) have higher knowledge capacity than smaller models (that is, *students)*. The teacher, typically a pre-trained or very large generic model (e.g., YOLOv8x6[1]), can transfer its knowledge (i.e., pattern recognition mechanisms) to students without significant model degradation. However, recourse to other models for labeling could lead to confirmation bias, a phenomenon that refers to noise accumulation when the model is trained using incorrect predictions for semi-supervised or unsupervised learning [2]. Furthermore, an immediate rebound effect of the scheme is

[1] There is no official paper available for this deep learning model. For the latest information, please visit the official repository: https://github.com/ultralytics/ultralytics.

the multiplication, on scale, of the number of models to be trained. The inference costs could become significant. Additionally, if the teacher model runs on a cloud-based platform, there may be additional costs associated with its usage, such as hourly usage fees or data transfer costs. This could be mitigated by using Active Learning (AL), which aims to identify the most informative examples for labeling. The importance of sampling has been first formulated in [4] as the problem of developing KD methods that are query-efficient and robust to labeling inaccuracies due to teacher imperfection (i.e., *confirmation bias*). The method developed in [4] was designed for a pool-based setting, which represents an offline scenario where a pool of unlabeled data points is made available to the learner. We claim that, in many real-world applications, a large number of unlabeled samples arrive in a streaming manner, **making it impossible to maintain all of the data in a candidate pool**. To the best of our knowledge, there is no framework supporting the development of AL methods that are query-efficient and robust to labeling inaccuracies in stream-based settings. **The contributions of this paper are the following:**

1. Formulate Stream-Based Active Distillation (SBAD) as the problem of developing AL methods that are both query-efficient and robust to labeling inaccuracies in stream-based settings.

2. Demonstrate the benefits of the proposed scheme for large-scale video-based object detections on a public dataset [26].

3. Establish simple but effective baselines to train a YOLOv8n student from a YOLOv8x6 teacher.

4. A code to reproduce the experiences and the framework available at `https://github.com/manjahdani/SBAD/`.

2. Related Work

2.1. Knowledge Distillation

KD is a method that involves training a smaller model to imitate the performance of a larger model. The main objectives of this technique are to prevent a decrease in the model's performance when it operates on a data set that is distributed differently than the source domain, referred to as Unsupervised Domain Adaptation (UDA), and to produce lightweight models suitable for the storage and computational capacities of miniaturized devices, referred to as Model Compression (MC) applications. In this study, we use a technique called *Self-training with knowledge distillation*, which was introduced by [6]. This technique trains a student model using pseudo-labels generated by a teacher model, which is beneficial when the labeled data is limited but we have access to a large sample of unlabeled data. Furthermore, the aforementioned distillation scheme does not need a direct access to the teacher. Yet, it may also propagate errors or biases.

In addition, we will discuss two additional techniques of interest in the following paragraphs: online distillation and context-aware distillation.

Online Distillation. This approach involves training a smaller student model to mimic the output of a larger teacher model on a per-example basis. In [13], the authors designed an online knowledge distillation scheme to perform real-time human segmentation in sports videos. Experiments show the ability of the model to adapt to contextual variations. Online distillation is also employed in [24] to adapt a low-cost semantic segmentation model to a target video where the data distribution is not necessarily stationary.

Context-aware Distillation. The works in [19, 28] attempt to exploit the contextual characteristics of the scene to develop effective KD. They directly worked on the distillation scheme to develop more specialized students. For example, [19] added a temporal dimension such that the student learns the variations in the intermediate features of the teacher over time, taking into account the redundancies of the frames within a CCTV stream.

2.2. Active Learning

AL is a sampling approach that selects the most informative data points to minimize the number of labels required for model training [33]. AL can be divided into three macro scenarios: synthesis of membership queries, pooled AL, and streamed AL [7]. The majority of approaches in deep AL have focused on the pool-based scenario, where the learner selects the most useful data points from a closed set of unlabeled observations. The stream-based AL scenario for object detectors has not been investigated. Moreover, AL assumes the availability of a perfect oracle, where the true label of a data point is revealed when queried. However, this assumption does not hold in a KD framework, where the pseudo-labels provided by the teacher may be incorrect.

Active Learning for Image Classification. AL strategies for pool-based classification can be categorized into uncertainty-based or diversity-based approaches [36]. Uncertainty-based strategies estimate model uncertainty using techniques such as Monte Carlo dropout [18] or ensemble networks [23], while entropy and margin-based sampling strategies are also widely employed [29]. Task-agnostic methods, such as Learn loss [38], use a loss prediction module to estimate data points that are likely to be wrongly predicted. Among diversity-based strategies, Core-set [32] is one of the most popular, using a K-center Greedy algorithm to locate a set of representative data points. Cluster-Margin [14] combines uncertainty and diversity, while DRMRS [16] takes into account the margin and diversity. BADGE [3] balances uncertainty and diversity using a $k-$MEANS++ seeding algorithm on gradients obtained from the last layer of the network. CDAL [1] replaces the Euclidean distance with the pairwise contextual diversity in the greedy K-center algorithm used in the Core-set. CLUE [25] performs uncertainty-weighted clustering to identify target instances that are uncertain according to the model and diverse in feature space. VAAL [34] uses a Variational Autoencoder (VAE) to map instances into a latent space, which is then fed into a discriminator that learns to differentiate between labeled data and unlabeled samples.

Active Learning for Object Detection. AL approaches to object detection can be classified into black-box and white-box methods [30]. Black-box methods do not depend on the underlying network architecture and use informativeness scores, such as the confidence obtained from the softmax layer, while white-box methods are dependent on the architecture of the underlying network. The Minmax approach, which selects the least confident images among the unlabeled pool, is a popular black-box method [30]. Ensemble methods have also been used for object detection-oriented AL [17, 31]. Query strategies based on localization tightness and stability [21], mixture density networks [12], and a unified box regression and classification metric [39] have also been proposed. MIAL [40] is a multiinstance framework that filters out noisy instances to bridge the gap between instance-level and image-level uncertainty. PPAL [37] is a two-stage algorithm that includes difficulty-calibrated uncertainty sampling and category-conditioned matching similarity. [20] proposed to cluster the unlabeled observations into groups based on the frequency domain values and to use different sampling rates for each group.

2.3. Challenges of Stream-based Active Distillation

The importance of sampling has been first formulated in [4] as the problem of developing KD methods that are both query-efficient and robust to labeling inaccuracies due to the imperfection of the teacher (i.e., *confirmation bias*). Their methods provide a theoretical guarantee that the scheme leads to queries where the teacher provides the correct labels. However, this approach has been developed in a pool-based setting where the student has access to the entire information pool. In contrast, in stream-based scenarios, techniques such as diversity-based strategies, clustering, or pairwise distance matrices may not be feasible, especially in contexts where the spatio-temporal correlation among the data is significant. Another aspect is that, due to the complexity of the student model, uncertainty techniques relying on Monte Carlo dropout or Learn loss modules may not be viable options.

3. Problem Statement

Let $\theta^{general}_{student}$ define a compact general pre-trained model learning the distribution $\mathcal{D}$ of a data stream $\mathcal{X}$. We assume a spatio-temporal correlation among the data. The student is equipped with SELECT (I_t), a rule that determines whether an image I_t should be selected to fine-tune the student model, using the pseudo-label predicted by a universal but imperfect model $\theta^{general}_{teacher}$. The objective is to train a high-performing student by querying the minimum number of teacher pseudo-labels. In this work, the pseudo-labels consist of bounding boxes generated by $\theta^{general}_{teacher}$ for each selected image. We assume a large-scale setting (e.g., city-scale deployment of CCTV, monitoring of large construction sites) and affordable hardware. Therefore, the selected frames and their associated pseudo-labels, which constitute the training set $\mathcal{L}$, must not exceed a maximum training frame budget per student B, i.e., $|\mathcal{L}| \leq B$. Furthermore, efficient SELECT strategies are necessary to ensure the scalability of our stream-based active distillation (SBAD). Indeed, if a selection rule takes longer than the frame rate to make a decision, a temporary buffer will be required to store recently seen images until the decision is made. This would increase the system resource requirements for data storage and processing, which is not scalable.

Algorithm 1 SBAD Framework

```
Require: a pre-trained student model $\theta^{general}_{student}$, a
    general purpose teacher model $\theta^{general}_{teacher}$, a training
    frame budget $B$ and a SELECT strategy.
Ensure: $B \geq 1$
    $\mathcal{L} \leftarrow \emptyset$    ▷ Selected frames and their pseudo-labels
    $t \leftarrow 0$                          ▷ Timestamp
    while $|\mathcal{L}| \leq B$ do
        Observe current frame $I_t$
        if SELECT($I_t$) is TRUE then
            $\{b_i^{pl}\}_t \leftarrow \theta_{teacher}(I_t)$    ▷ Pseudo-labels
            $\mathcal{L} \leftarrow \mathcal{L} \cup (I_t, \{b_i^{pl}\}_t)$
        end if
        $t \leftarrow t + 1$
    end while
    return update($\theta^{general}_{student}$, $\mathcal{L}$)
```

Figure 1 provides a visual illustration of the SBAD framework. During the sampling phase, the SELECT rule is used to identify the most informative samples. The selected frames are then pseudo-labeled by the teacher model and used to fine-tune the student models. Once the fine-tuning is complete, specialized models could be optionally evaluate using a test-set with ground truth $\mathcal{T} := \{I^{test}, \mathbf{b^{gt}}\}$. Note that this step is not necessary for SBAD, but in real-life scenarios, it could be seen as a sanity check if you have access to a test-set.

4. Methodology

In the context of stream-based active learning, single-pass evaluation of data points is often addressed by applying a threshold to certain informativeness scores [8–11, 15, 27]. However, this approach has not been tested in online active distillation tasks for object detection. In this paper, we investigate the effectiveness of thresholding algorithms based on the confidence of the base student model $\theta^{general}_{student}$ for the SBAD framework. At round t, when the student model $\theta^{general}_{student}$ observes an image I_t, $n \geq 0$ objects are detected, which are defined by the bounding boxes b_{it} and confidence scores c_{it}. According to [30], a unique confidence score $C(I_t)$ can be obtained for I_t using:

$$C(I_t) := \max_i c_{it}$$

This means that the confidence of each image is approximated by the highest confidence score among the objects detected in that image. Using this confidence metric, we can then apply a threshold Δ to the confidence scores of the incoming frames. The general structure of the top confidence threshold sampling scheme is presented in Algorithm 1. To estimate the threshold Δ for selecting the most informative frames, we introduce a warm-up phase where the student model $\theta^{general}_{student}$ observes the incoming frames for a period of length w without querying any image and without storing anything other than a single scalar representing the confidence scores $C(I_t)$ at the image level, where $t = 1, ..., w$. At the end of the warm-up phase, the student model estimates an $(1 - \alpha)$-upper percentile on the distribution of confidence scores, where α represents the desired sampling rate. In other words, the threshold Δ is chosen so that:

$$\mathbb{P}(C(I_t) \geq \Delta) = \alpha,$$

and the frames to pseudo-label and fine-tune $\theta^{general}_{student}$ correspond to a ratio of α frames out of the total number of frames.

While in traditional AL, the focus is on querying images that the student model is least confident about, this approach may not be optimal for stream-based object-detection KD scenarios. The least confident images often correspond to very hard examples that may not be informative enough for the student model in the early rounds of AL when it has not been fine-tuned for the specific scene. Additionally, selecting images with high uncertainty for pseudo-labeling may lead to confirmation bias as the pseudo-labels may not align with the ground truth due to the imperfection of the teacher model $\theta^{general}_{teacher}$ as an oracle. This is why, in our work, we propose to let the student model $\theta^{general}_{student}$ query the most confident frames. Ideally, by doing so, the student will sample informative examples that the teacher model can accurately pseudo-label. These examples will contribute best to the student's fine-tuning while avoiding frames that are too uncertain to be used in the initial stages AL.

5. Experiments

5.1. Experimental Settings

Dataset. We evaluated the effectiveness of the SBAD approach using the Watch and Learn Time-lapse (WALT) data set [26], which comprises 12[2]

[2] We tested two out of twelve cameras and produced extra annotations to evaluate our techniques. Detailed information about this process and the dataset are available in our GitHub repository.

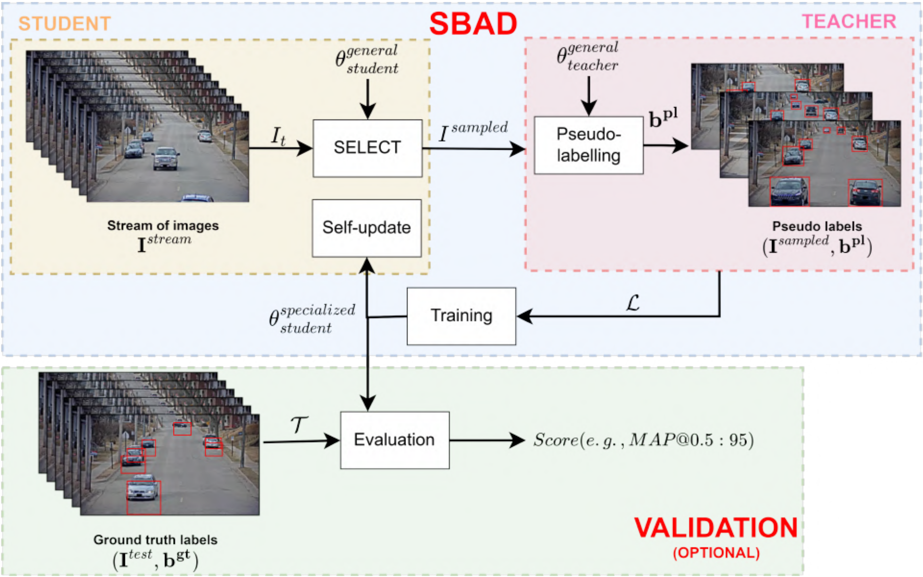


Figure 1. SBAD pipeline: sampling, fine-tuning and evaluation.

cameras that capture an urban environment. This data set offers a diverse range of spatial and temporal settings, with varying viewpoints and lighting conditions, including both day and night settings. By testing our approach on this realistic data set, we assess its performance in real-world scenarios.

Distillation implementation. In line with the principles of data distillation proposed by [6], we employ a large and complex teacher model, YOLOv8x6 (261.1 GFLOPs), to generate pseudo-labels. These labels are then used to train several smaller student models, YOLOv8n (8.7 GFLOPs), with less architectural complexity. Both networks are initially pretrained on the COCO dataset [22]. The student models are re-trained for 100 epochs with a batch size of 16 and a learning rate (LR) of 0.01. The learning rate is adjusted for each epoch with a change factor (LF) of 0.01 using Equation 1. The budget of the SBAD framework is determined by the number of pseudo-labels used for fine-tuning, which ranges from 25 to 250 in our experiments.

$$LR = \left(\frac{1 - LR}{epochs}\right) \times (1 - LF) + LF \quad (1)$$

Methods. Due to the lack of prior research on the SBAD problem in object detection, there are no baselines to compare with. To explore the effectiveness of the confidence-based thresholding algorithm, we used different baselines. First, a naive N-*First* approach has been implemented, where the student models are fine-tuned by simply taking the first N images observed from each camera. A second baseline is given by a *random* sampling approach, where a number $s \sim U(0, 1)$ is generated for each incoming frame, which is queried only if $s \geq 1 - \alpha$. A third baseline is given by a more active learning-oriented *least confidence* approach, where similarly to the case of the highest confidence, we impose a threshold on the confidence score at the image level. The main difference is that the threshold Δ is estimated by taking the α-lower percentile from the warm-up set $\mathcal{W}$.

In our experiments, both α-lower and α-higher methods used $\alpha = 10\%$. However, it is important to note that this choice was influenced by the frame rate and the length of the data stream recorded for each week. Although smaller values of α may yield better performance, they would need to span a longer data stream as we become more selective in terms of selecting only the most confident frames. There-

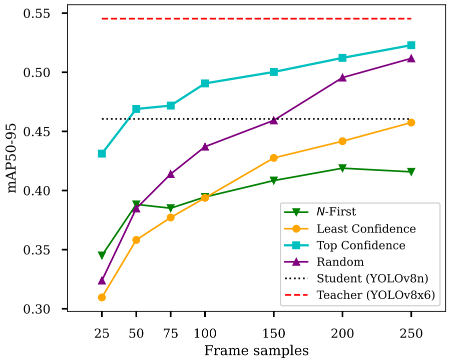


Figure 2. Learning curves obtained on the first two cameras of WALT. Results show that increasing the number of frames used for fine-tuning improves the student model's performance, approaching that of the teacher model with 250 frames. However, using only a small number of frames may lead to overfitting and poor performance on balanced evaluation sets. Top confidence thresholding is more effective than least confidence-based methods for stream-based active learning, highlighting the importance of avoiding highly uncertain images during fine-tuning.

fore, the choice of α should be based on a balance between performance and the length of the data stream required to select the desired number of frames.

5.2. Experimental Results

Figures 2. and 3. shows the learning curves obtained using stream-based active learning techniques on the WALT dataset. Our analysis can be approached from two perspectives. Firstly, from a knowledge distillation standpoint, we observe how the student model's performance improves as we use more frames for fine-tuning. In particular, we found that the mAP50-95 score approaches that of the teacher model when 250 pseudo-labeled frames are used. However, we also noticed that the student's performance deteriorates significantly when only a small number of frames are used for fine-tuning, which could be attributed to overfitting due to the limited number of images presented to the network. In addition, if the model is fine-tuned on images biased towards a specific time of day, such as only night or day, it may perform poorly on the balanced test set used for evaluation. Furthermore, as depicted in Figure 4, choosing highly uncertain images for pseudo-labeling may lead to incorrect labels due to the teacher's own bad prediction.

From an active learning perspective, the performance achieved with the *top confidence threshold* algorithm is significantly better than that obtained using the least confidence-based method. This highlights the importance of fine-tuning the model with highly certain images, especially when the model has not yet been specialized for the scene.

5.3. Limitations

The present work has three limitations. Firstly, the maximum budget is limited to 250 due to the frame rate and length of the data stream. Second, our approach was only evaluated on the WALT data set, and its generalizability to other data sets remains to be investigated. Third, the reduced number of heuristics may limit the effectiveness of the approach, and further exploration of different methods or combinations of methods could be a fruitful research direction. Additionally, exploring other deep neural network architectures, such as Transformers or Mask-RCNN, could also enhance the approach.

6. Conclusion

This paper proposes SBAD to bridge the gap between large-scale and affordable deep learning models while adapting to changing environments. This framework enables the scalable deployment of deep learning models under tight budget constraints.

The framework evaluates the informativeness of each frame, accounting for teacher imperfections in a KD scheme. Experiments demonstrate that traditional AL strategies may not be optimal for KD. Future research could explore alternative sampling strategies and distillation mechanisms to improve performance.

Acknowledgments

This work was partially funded by Win2WAL (#1910045) and Trusted AI Labs. We also thank *Openhub* for providing the equipment.

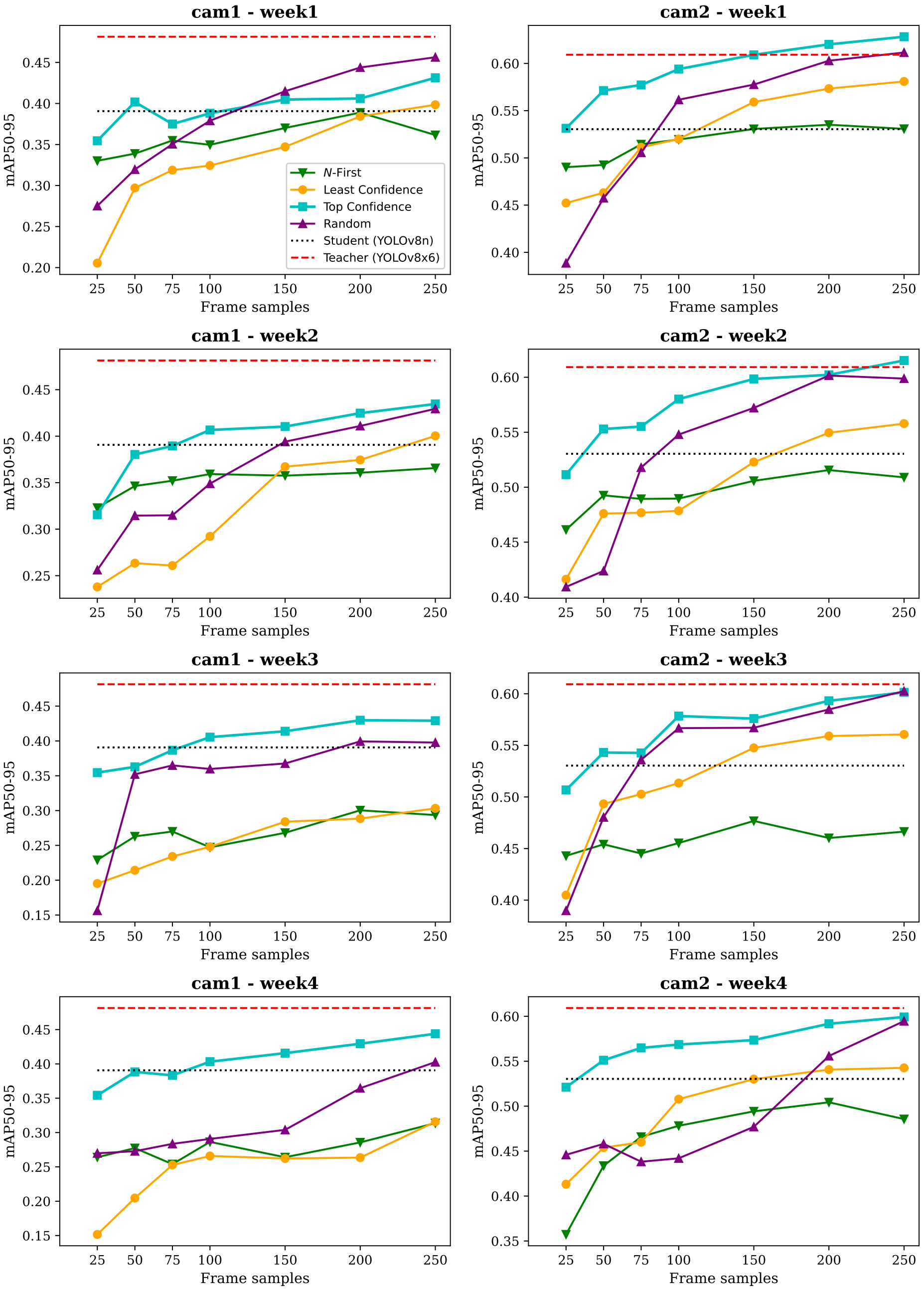


Figure 3. Weekly analysis on the first two cameras of WALT.

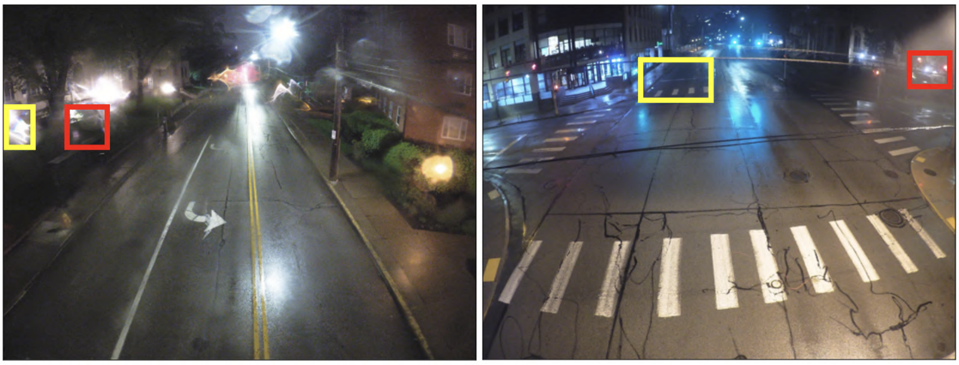


Figure 4. Two difficult examples (one for each camera) that lead to *confirmation bias*: when the student requests highly uncertain images based on its predictions (in yellow), wrong pseudo labels are revealed (in red).

References

[1] Sharat Agarwal, Himanshu Arora, Saket Anand, and Chetan Arora. Contextual diversity for active learning. In *European Conference on Computer Vision (ECCV) 2020*, 8 2020. 3

[2] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020. 1

[3] Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *2020 International Conference on Learning Representations*, 6 2019. 3

[4] Cenk Baykal, Khoa Trinh, Fotis Iliopoulos, Gaurav Menghani, and Erik Vee. Robust active distillation. *arXiv preprint arXiv:2210.01213*, 2022. 2, 3

[5] Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. Knowledge distillation: A good teacher is patient and consistent. *CoRR*, abs/2106.05237, 2021. 1

[6] Cristian Buciluundefined, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, page 535–541, New York, NY, USA, 2006. Association for Computing Machinery. 2, 5

[7] Davide Cacciarelli and Murat Kulahci. A survey on online active learning. *arXiv preprint arXiv:2302.08893*, 2023. 2

[8] Davide Cacciarelli, Murat Kulahci, and John Tyssedal. Online active learning for soft sensor development using semi-supervised autoencoders. In *ICML 2022 Workshop on Adaptive Experimental Design and Active Learning in the Real World*, 2022. 4

[9] Davide Cacciarelli, Murat Kulahci, and John Sølve Tyssedal. Stream-based active learning with linear models. *Knowledge-Based Systems*, 254:109664, 10 2022. 4

[10] Davide Cacciarelli, Murat Kulahci, and John Sølve Tyssedal. Robust online active learning. *arXiv preprint arXiv:2302.00422*, 2023. 4

[11] Andrea Castellani, Sebastian Schmitt, and Barbara Hammer. Stream-based active learning with verification latency in non-stationary environments. In *Artificial Neural Networks and Machine Learning 2022*, 4 2022. 4

[12] Jiwoong Choi, Ismail Elezi, Hyuk-Jae Lee, Clément Farabet, and Jose M. Alvarez. Active learning for deep object detection via probabilistic modeling. *CoRR*, abs/2103.16130, 2021. 3

[13] Anthony Cioppa, Adrien Deliege, Maxime Istasse, Christophe De Vleeschouwer, and Marc Van Droogenbroeck. Arthus: Adaptive real-time human segmentation in sports through online distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 2

[14] Gui Citovsky, Giulia DeSalvo, Claudio Gentile, Lazaros Karydas, Anand Rajagopalan, Afshin Rostamizadeh, and Sanjiv Kumar. Batch active learning at scale. In *Conference on Neural Information Processing Systems*, 7 2021. 3

[15] Sanjoy Dasgupta, Adam Kalai, and Claire Monteleoni. Analysis of perceptron-based active learning. In *Lecture Notes in Computer Science*, volume 10, 12 2005. 4

[16] Ehsan Elhamifar, Guillermo Sapiro, Allen Yang, and S. Shankar Sasrty. A convex optimization framework for active learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 209–

216. Institute of Electrical and Electronics Engineers Inc., 2013. 3

[17] Di Feng, Xiao Wei, Lars Rosenbaum, Atsuto Maki, and Klaus Dietmayer. Deep active learning for efficient training of a lidar 3d object detector. In *30th IEEE Intelligent Vehicles Symposium*, 2019. 3

[18] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning*, 2017. 3

[19] Amirhossein Habibian, Haitam Ben Yahia, Davide Abati, Efstratios Gavves, and Fatih Porikli. Delta distillation for efficient video processing, 2022. 2

[20] Wei Huang, Shuzhou Sun, Xiao Lin, Dawei Zhang, and Lizhuang Ma. Deep active learning with weighting filter for object detection. *Displays*, page 102282, 1 2022. 3

[21] Chieh-Chi Kao, Teng-Yok Lee, Pradeep Sen, and Ming-Yu Liu. Localization-aware active learning for object detection. In *Asian Conference on Computer Vision (ACCV) 2018*, 1 2018. 3

[22] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. 5

[23] Salman Mohamadi, Gianfranco Doretto, and Donald A Adjeroh. Deep active ensemble sampling for image classification. In *16th Asian Conference on Computer Vision (ACCV 2022)*, 2022. 3

[24] Ravi Teja Mullapudi, Steven Chen, Keyi Zhang, Deva Ramanan, and Kayvon Fatahalian. Online model distillation for efficient video inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3573–3582, 2019. 2

[25] Viraj Prabhu, Arjun Chandrasekaran, Kate Saenko, and Judy Hoffman. Active domain adaptation via clustering uncertainty-weighted embeddings. In *International Conference on Computer Vision (ICCV) 2021*, 2020. 3

[26] N. Dinesh Reddy, Robert Tamburo, and Srinivasa G. Narasimhan. Walt: Watch and learn 2d amodal representation from time-lapse imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9356–9366, June 2022. 2, 4

[27] Carlos Riquelme, Ramesh Johari, and Baosen Zhang. Online active linear regression via thresholding. In *31st AAAI Conference on Artificial Intelligence*, 2017. 4

[28] Daniel Rivas, Francesc Guim, Jordà Polo, Pubudu M Silva, Josep Ll Berral, and David Carrera. Towards automatic model specialization for edge video analytics. *Future Generation Computer Systems*, 2022. 2

[29] Dan Roth and Kevin Small. Margin-based active learning for structured output spaces. In *European Conference on Machine Learning (ECML*, 2006. 3

[30] Soumya Roy, Asim Unmesh, and Vinay P Namboodiri. Deep active learning for object detection. *29th British Machine Vision Conference(BMVC)*, 2018. 3, 4

[31] Sebastian Schmidt, Qing Rao, Julian Tatsch, and Alois Knoll. Advanced active learning strategies for object detection. In *2020 IEEE Intelligent Vehicles Symposium*, 2020. 3

[32] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *ICLR 2018*, 8 2017. 3

[33] Burr Settles. Active learning literature survey. *Computer Sciences Technical article, University of Wisconsin–Madison*, 2009. 2

[34] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5972–5981, 2019. 3

[35] Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum self-paced learning for cross-domain object detection. *Computer Vision and Image Understanding*, 204:103166, 2021. 1

[36] Jiaxi Wu, Jiaxin Chen, and Di Huang. Entropy-based active learning for object detection with progressive diversity constraint. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2022. 3

[37] Chenhongyi Yang, Lichao Huang, and Elliot J. Crowley. Plug and play active learning for object detection. *http://arxiv.org/abs/2211.11612*, 2022. 3

[38] Donggeun Yoo and In So Kweon. Learning loss for active learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5 2019. 3

[39] Weiping Yu, Sijie Zhu, Taojiannan Yang, and Chen Chen. Consistency-based active learning for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2021. 3

[40] Tianning Yuan, Fang Wan, Mengying Fu, Jianzhuang Liu, Songcen Xu, Xiangyang Ji, and Qixiang Ye. Multiple instance active learning for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4 2021. 3

Bibliography

[1] X. He and R. Zemel, “Learning hybrid models for image annotation with partially labeled data,” *Advances in Neural Information Processing Systems*, vol. 21, 2008.

[2] S. Budd, E. C. Robinson, and B. Kainz, “A survey on active learning and human-in-the-loop deep learning for medical image analysis,” *Medical image analysis*, vol. 71, p. 102062, 2021.

[3] E. Weigl, W. Heidl, E. Lughofer, T. Radauer, and C. Eitzinger, “On improving performance of surface inspection systems by online active learning and flexible classifier updates,” *Machine Vision and Applications*, vol. 27, pp. 103–127, 1 2016.

[4] I. S. Ramírez, F. P. G. Márquez, and M. Papaelias, “Review on additive manufacturing and non-destructive testing,” *Journal of Manufacturing Systems*, vol. 66, pp. 260–286, 2023.

[5] D. Reker and G. Schneider, “Active-learning strategies in computer-assisted drug discovery,” *Drug discovery today*, vol. 20, no. 4, pp. 458–465, 2015.

[6] K. Fowler, K. Kokilepersaud, M. Prabhushankar, and G. AlRegib, “Clinical trial active learning,” in *The 14th ACM Conference on Bioinformatics, Computational Biology and Health Informatics (ACM-BCB)*, 2023.

[7] S. Hanneke and R. Nowak, “Active learning from theory to practice,” 2019.

[8] B. Settles, “Active learning literature survey,” *Technical Report 1648, University of Wisconsin-Madison Department of Computer Sciences*, 2009.

[9] B. Settles, “Active learning literature survey,” 2010.

[10] S. Dasgupta, “Two faces of active learning,” *Theoretical Computer Science*, vol. 412, no. 19, pp. 1767–1781, 2011. Algorithmic Learning Theory (ALT 2009).

[11] M. M. Gaber, A. Zaslavsky, and S. Krishnaswamy, “Mining data streams: a review,” *ACM Sigmod Record*, vol. 34, no. 2, pp. 18–26, 2005.

[12] P. R. Freeman, “The secretary problem and its extensions: A review,” *International Statistical Review*, vol. 51, pp. 189–206, 1983.

[13] Flaticon.com *https://www.flaticon.com*, 2023.

[14] L. Fortuna, S. Graziani, A. Rizzo, and M. G. Xibilia, *Soft sensors for monitoring and control of industrial processes*, vol. 22. Springer, 2007.

[15] L. L. T. Chan, Q. Y. Wu, and J. Chen, "Dynamic soft sensors with active forward-update learning for selection of useful data from historical big database," *Chemometrics and Intelligent Laboratory Systems*, vol. 175, pp. 87–103, 4 2018.

[16] J. Zheng and Z. Song, "Mixture modeling for industrial soft sensor application based on semi-supervised probabilistic pls," *Journal of Process Control*, vol. 84, pp. 46–55, 12 2019.

[17] Z. Ge, "Active probabilistic sample selection for intelligent soft sensing of industrial processes," *Chemometrics and Intelligent Laboratory Systems*, vol. 151, pp. 181–189, 2 2016.

[18] Z. Ge, "Active learning strategy for smart soft sensor development under a small number of labeled data samples," *Journal of Process Control*, vol. 24, pp. 1454–1461, 2014.

[19] S. Shalev-Shwartz, "Online learning and online convex optimization," *Foundations and Trends in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2012.

[20] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games.* Cambridge University Press, 3 2006.

[21] D. Zha, Z. P. Bhat, K.-H. Lai, F. Yang, Z. Jiang, S. Zhong, and X. Hu, "Data-centric artificial intelligence: A survey," *arXiv preprint arXiv:2303.10158*, 2023.

[22] Y. Roh, G. Heo, and S. E. Whang, "A survey on data collection for machine learning: a big data-ai integration perspective," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 4, pp. 1328–1347, 2019.

[23] T. Fredriksson, D. I. Mattos, J. Bosch, and H. H. Olsson, "Data labeling: An empirical investigation into industrial challenges and mitigation strategies," in *International Conference on Product-Focused Software Process Improvement*, pp. 202–216, Springer, 2020.

[24] S. García, J. Luengo, and F. Herrera, *Data preprocessing in data mining*, vol. 72. Springer, 2015.

[25] R. D. Kjærsgaard, M. G. Grønberg, and L. K. H. Clemmensen, "Sampling to improve predictions for underrepresented observations in imbalanced data," *Workshop on Data-Centric AI (NeurIPS 2021)*, 11 2021.

[26] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 2, pp. 539–550, 2008.

[27] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

[28] D. Di Nardo, F. Pastore, and L. Briand, “Generating complex and faulty test data through model-based mutation analysis,” in *2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST)*, pp. 1–10, IEEE, 2015.

[29] B. Krawczyk, B. Pfahringer, and M. Wozniak, “Combining active learning with concept drift detection for data stream mining,” *2018 IEEE International Conference on Big Data (Big Data)*, pp. 2239–2244, 12 2018.

[30] D. C. Montgomery, *Design and Analysis of Experiments.* John Wiley & Sons, Inc., 1 2012.

[31] C. C. Drovandi, C. Holmes, J. M. McGree, K. Mengersen, S. Richardson, and E. G. Ryan, “Principles of experimental design for big data analysis,” *Statistical science: a review journal of the Institute of Mathematical Statistics*, vol. 32, no. 3, p. 385, 2017.

[32] S. Karlin and W. J. Studden, “Optimal experimental designs,” *The Annals of Mathematical Statistics*, vol. 37, pp. 783–815, 1966.

[33] M. R. Gahrooei, K. Paynabar, M. Pacella, and B. M. Colosimo, “An adaptive fused sampling approach of high-accuracy data in the presence of low-accuracy data,” *IISE Transactions*, vol. 51, no. 11, pp. 1251–1264, 2019.

[34] K. Liu, Y. Mei, and J. Shi, “An adaptive sampling strategy for online high-dimensional process monitoring,” *Technometrics*, vol. 57, no. 3, pp. 305–319, 2015.

[35] A. M. E. Gómez, D. Li, and K. Paynabar, “An adaptive sampling strategy for online monitoring and diagnosis of high-dimensional streaming data,” *Technometrics*, vol. 64, no. 2, pp. 253–269, 2022.

[36] D. C. Montgomery, *Introduction to statistical quality control.* John wiley & sons, 2019.

[37] D. Cacciarelli and M. Kulahci, “Active learning for data streams: a survey,” *Machine Learning*, 2023.

[38] M. Reid, “Reliability – a python library for reliability engineering,” 2022.

[39] N. A. Heckert, J. J. Filliben, C. M. Croarkin, B. Hembree, W. F. Guthrie, P. Tobias, and J. Prinz, “Handbook 151: NISTe-handbook of statistical methods,” 2002.

[40] A. Wald, *Sequential analysis.* Courier Corporation, 2004.

[41] S. Gajjar, M. Kulahci, and A. Palazoglu, "Real-time fault detection and diagnosis using sparse principal component analysis," *Journal of Process Control*, vol. 67, pp. 112–128, 7 2018.

[42] E. Vanhatalo and M. Kulahci, "Impact of autocorrelation on principal components and their use in statistical process control," *Quality and Reliability Engineering International*, vol. 32, pp. 1483–1500, 6 2016.

[43] E. Vanhatalo, M. Kulahci, and B. Bergquist, "On the structure of dynamic principal component analysis used in statistical process monitoring," *Chemometrics and Intelligent Laboratory Systems*, vol. 167, pp. 1–11, 8 2017.

[44] J. MacGregor and T. Kourti, "Statistical process control of multivariate processes," *Control Engineering Practice*, vol. 3, 3 1995.

[45] H. Hotelling, "Multivariate quality control," *Techniques of Statistical Analysis*, 1947.

[46] D. Cacciarelli and M. Kulahci, "A novel fault detection and diagnosis approach based on orthogonal autoencoders," *Computers & Chemical Engineering*, vol. 163, p. 107853, 7 2022.

[47] R. M. Ueda and A. M. Souza, "An effective approach to detect the source(s) of out-of-control signals in productive processes by vector error correction (vec) residual and hotelling's t2 decomposition techniques," *Expert Systems with Applications*, vol. 187, 1 2022.

[48] H. Sabahno, A. Amiri, and P. Castagliola, "Evaluating the effect of measurement errors on the performance of the variable sampling intervals hotelling's t2 control charts," *Quality and Reliability Engineering International*, vol. 34, pp. 1785–1799, 12 2018.

[49] B. R. de Almeida Moreira, V. H. Cruz, M. L. C. Oliveira, and R. da Silva Viana, "Full-scale production of high-quality wood pellets assisted by multivariate statistical process control," *Biomass and Bioenergy*, vol. 151, 8 2021.

[50] N. L. Chong, M. B. Khoo, A. Haq, and P. Castagliola, "Hotelling's t2 control charts with fixed and variable sample sizes for monitoring short production runs," *Quality and Reliability Engineering International*, vol. 35, pp. 14–29, 2 2019.

[51] Y. Ruan, J. Yang, and Y. Zhou, "Linear bandits with limited adaptivity and learning distributional optimal design," *STOC 2021: Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, 7 2020.

[52] J.-Y. Audibert and R. Munos, "Best arm identification in multi-armed bandits," *COLT - 23th Conference on Learning Theory*, 2010.

[53] K. Jamieson and R. Nowak, "Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting," *2014 48th Annual Conference on Information Sciences and Systems (CISS)*, pp. 1–6, 3 2014.

[54] M. Soare, A. Lazaric, and R. Munos, "Active learning in linear stochastic bandits," *Bayesian Optimization in Theory and Practice*, 2013.

[55] M. Soare, A. Lazaric, and R. Munos, "Best-arm identification in linear bandits," *27th Conference on Neural Information Processing Systems (NeurIPS 2014)*, 2014.

[56] Y. Jedra and A. Proutiere, "Optimal best-arm identification in linear bandits," *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020.

[57] M. J. Azizi, B. Kveton, and M. Ghavamzadeh, "Fixed-budget best-arm identification in structured bandits," *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)*, 6 2022.

[58] S. Filippi, O. Cappe, A. Garivier, and C. Szepesvári, "Parametric bandits: The generalized linear case," *Advances in Neural Information Processing Systems 23 (NIPS 2010)*, 2010.

[59] J. E. Van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine learning*, vol. 109, no. 2, pp. 373–440, 2020.

[60] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[61] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," *Advances in neural information processing systems*, vol. 27, 2014.

[62] Z. Chen and B. Liu, "Lifelong machine learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 12, no. 3, pp. 1–207, 2018.

[63] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019.

[64] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[65] A. Robins, "Catastrophic forgetting, rehearsal and pseudorehearsal," *Connection Science*, vol. 7, no. 2, pp. 123–146, 1995.

[66] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, “Progressive neural networks,” *arXiv preprint arXiv:1606.04671*, 2016.

[67] A. Dekhovich, O. T. Turan, J. Yi, and M. A. Bessa, “Cooperative data-driven modeling,” *Computer Methods in Applied Mechanics and Engineering*, vol. 417, p. 116432, 2023.

[68] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, pp. 1126–1135, 2017.

[69] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, “Matching networks for one shot learning,” *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[70] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” pp. 41–48, 2009.

[71] T. S. Ferguson, “Who solved the secretary problem?,” *Statistical science*, vol. 4, no. 3, pp. 282–289, 1989.

[72] C. Riquelme, R. Johari, and B. Zhang, “Online active linear regression via thresholding,” *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[73] K. Zhang, A. T. Bui, and D. W. Apley, “Concept drift monitoring and diagnostics of supervised learning models via score vectors,” *Technometrics*, vol. 65, no. 2, pp. 137–149, 2023.

[74] R. B. Gramacy and H. K. Lee, “Cases for the nugget in modeling computer experiments,” *Statistics and Computing*, vol. 22, pp. 713–722, 2012.

[75] T. J. Santner, B. J. Williams, W. I. Notz, and B. J. Williams, *The design and analysis of computer experiments*, vol. 1. Springer, 2003.

[76] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* John Wiley & Sons, 2012.

[77] C. E. Rasmussen and H. Nickisch, “Gaussian processes for machine learning (gpml) toolbox,” *The Journal of Machine Learning Research*, vol. 11, pp. 3011–3015, 2010.

[78] M. Wunder, M. L. Littman, and M. Babes, “Classes of multiagent q-learning dynamics with epsilon-greedy exploration,” in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 1167–1174, 2010.

[79] N. O'Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, L. Krpalkova, D. Riordan, and J. Walsh, "Deep learning vs. traditional computer vision," in *Advances in Computer Vision: Proceedings of the 2019 Computer Vision Conference (CVC), Volume 1 1*, pp. 128–144, Springer, 2020.

[80] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, "Deep learning for smart manufacturing: Methods and applications," *Journal of manufacturing systems*, vol. 48, pp. 144–156, 2018.

[81] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *CVPR*, 2001.

[82] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[83] G. N. Chaple, R. Daruwala, and M. S. Gofane, "Comparisions of robert, prewitt, sobel operator based edge detection methods for real time uses on fpga," in *2015 International Conference on Technologies for Sustainable Development (ICTSD)*, pp. 1–4, IEEE, 2015.

[84] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, 2015.

[85] A. Canziani and Y. LeCunn, "NYU deep learning course," 2021.

[86] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[87] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *CVPR*, 2016.

[88] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NIPS*, 2015.

[89] X. Lu, Q. Li, B. Li, and J. Yan, "Mimicdet: Bridging the gap between one-stage and two-stage object detection," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pp. 541–557, Springer, 2020.

[90] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pp. 21–37, Springer, 2016.

[91] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *CVPR*, 2016.

[92] K. Technology, "Data labeling services price Q3 2023 benchmark," 2023. Accessed: 12/10/2023.

[93] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, “A survey of deep active learning,” *ACM Computing Surveys*, vol. 54, pp. 1–40, 12 2022.

[94] C. Baykal, K. Trinh, F. Iliopoulos, G. Menghani, and E. Vee, “Robust active distillation,” *arXiv preprint arXiv:2210.01213*, 2022.

[95] S. Roy, A. Unmesh, and V. P. Namboodiri, “Deep active learning for object detection,” *29th British Machine Vision Conference(BMVC)*, 2018.

[96] C.-A. Brust, C. Käding, and J. Denzler, “Active learning for deep object detection,” *ArXiv Preprint*, 9 2018.

[97] N. D. Reddy, R. Tamburo, and S. G. Narasimhan, “Walt: Watch and learn 2d amodal representation from time-lapse imagery,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9356–9366, June 2022.

[98] C. Baykal, K. Trinh, F. Iliopoulos, G. Menghani, and E. Vee, “Robust active distillation,” 10 2022.

ISBN 978-82-326-7964-5 (printed ver.)
ISBN 978-82-326-7963-8 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

DTU

NTNU
Norwegian University of
Science and Technology