Robust Concept Drift Handling for Industrial Applications

Continuous Maintenance of Productive Machine Learning Regressors

Zur Erlangung des akademischen Grades eines

DOKTORS DER INGENIEURWISSENSCHAFTEN
(Dr.-Ing.)

von der KIT-Fakultät für
Maschinenbau
des Karlsruher Instituts für Technologie (KIT)

genehmigte

DISSERTATION

von

M.Sc. Martin Trat
geb. in Bayreuth

Tag der mündlichen Prüfung:	27.02.2026
Hauptreferent:	Prof. Dr. Dr.-Ing. Dr. h. c. Jivka Ovtcharova
Korreferent:	Prof. Dr.-Ing. Benjamin Menz

Zusammenfassung

Maschinelles Lernen hat sich zu einem zentralen Enabler der datengetriebenen industriellen Transformation entwickelt. Die langfristige Aufrechterhaltung der Leistungsfähigkeit von Machine-Learning-Modellen in Produktivumgebungen stellt jedoch weiterhin eine erhebliche Herausforderung dar. Nach ihrer Inbetriebnahme sind Modelle kontinuierlichen Datenströmen, also potenziell endlosen Sequenzen von Datenpunkten, ausgesetzt. Im Allgemeinen ist davon auszugehen, dass sich deren statistische Eigenschaften im Zeitverlauf verändern, was auch als Concept Drift bezeichnet wird. Unbeachtet kann dieser die Genauigkeit und Zuverlässigkeit von Modellen erheblich beeinträchtigen. Diese Arbeit befasst sich daher mit der kontinuierlichen Wartung von Modellen in industriellen Anwendungskontexten. Hierbei liegt der Fokus auf dem robusten Umgang mit Concept Drift, insbesondere im Kontext von Regressionsproblemen, was bislang nur unzureichend erforscht ist.

Entwickelt wird daher eine Referenzarchitektur, die kohärent die Modellierung und den Entwurf von Machine-Learning-Modellsystemen mit robustheitsbewahrenden Komponenten unterstützt. Darüber hinaus setzt die Arbeit die Exploration von Concept Drift Detection Ensembles fort, indem sie eine umfassende Literaturanalyse, eine Taxonomie sowie ein Entwurfsschema für deren effektive Konfiguration bereitstellt. Aufbauend auf diesen Erkenntnissen wird die neuartige Methode für Concept-Drift-Adaption in Regressionsszenarien (SAM-CDAR) entwickelt. Sie integriert die Detektor-Ensemble-Struktur mit einem Machine-Learning-Modell-Ensemble, das die Robustheit von Regressoren mittels naturinspirierter Mechanismen zur Concept-Drift-Adaption aufrechterhält.

Die vorgeschlagene Methode wird in zwei realen industriellen Regressionsszenarien experimentell validiert. In diesen wird die Machine-Learning-gestützte Prädiktion von Prozessschrittdauer-Angaben in einem Freiformschmiedebetrieb sowie von Lieferzeiten in der Möbelproduktlogistik betrachtet. Die Auswertung zeigt, dass SAM-CDAR im Vergleich zu mehreren Baseline-Ansätzen deutliche Reduktionen der Regressionsfehler erzielt, mit Verbesserungen von bis zu 43,85% bzw. 60,82%. Diese Ergebnisse unterstreichen die Wirksamkeit der Methode bei der Aufrechterhaltung der Modellrobustheit unter dem Einfluss unterschiedlicher Arten von Concept Drift und belegen zugleich ihren praktischen Nutzen.

Abstract

Machine learning has become a key enabler of data-driven industrial transformation. Yet, sustaining the performance of machine learning estimators in productive environments still poses a persistent challenge. Once deployed, estimators are exposed to data streams, i.e. potentially endless sequences of data points. Generally, one must assume that their statistical properties evolve over time, which is also known as the phenomenon referred to as concept drift. If left unaddressed, concept drift can severely degrade estimator performance and reliability. This thesis therefore addresses the continuous maintenance of estimators in industrial contexts by focusing on robust concept drift handling with a particular emphasis on regression problems, which marks a problem that remains critically under-researched.

To this end, a reference architecture that provides a coherent framework for modeling and designing machine learning systems with robustness-sustaining components is proposed. The thesis also further explores concept drift detection ensembles by contributing a comprehensive literature analysis, taxonomy and design blueprint for their effective configuration. Building upon these insights, the novel stream-applicable method for concept drift adaptation in regression scenarios (SAM-CDAR) is developed. It integrates this detector ensemble structure with a novel machine learning ensemble approach maintaining regressor robustness through nature-inspired concept drift adaptation mechanisms.

The proposed method is experimentally validated in two real-world industrial regression scenarios involving the estimation of manufacturing process step durations in a free-form forging company and of delivery times in furniture product logistics. The evaluation demonstrates that SAM-CDAR achieves substantial reductions in regression error compared to multiple baseline approaches, with improvements of up to 43.85% and 60.82%, respectively. Furthermore, the findings underline the method's capacity to preserve regressor robustness under diverse types of concept drift, thereby affirming its practical value.

Acknowledgments

This thesis was developed during my time as a research scientist in the division Intelligent Systems and Production Engineering at the FZI Research Center for Information Technology. My heartfelt gratitude goes to Prof. Dr. Dr.-Ing. Dr. h. c. Jivka Ovtcharova, who made this work possible by granting me the opportunity to pursue it, and whose guidance and vision accompanied me through every stage of its development. Her encouragement, insightful critique as well as generous freedom in shaping the research path have deeply influenced not only this thesis but also my way of thinking as a scientist.

I would also like to express my sincere appreciation to Prof. Dr.-Ing. Benjamin Menz for his thoughtful engagement as a member of the examination committee and for his continued mentorship. His perspective and advice have been of great value.

My sincere thanks go to my colleagues at the FZI and beyond, who made this time both stimulating and enjoyable. I would particularly like to mention Dr.-Ing. Janek Bender, Dr.-Ing. Jana Deckers, Matthias Jaenicke, Chenwei Sun, Mine Felder, Burçin Moumin, Dr.-Ing. Matthes Elstermann and Dr.-Ing. Victor Häfner for their support, expertise and steady motivation. Working alongside such dedicated minds has been a privilege.

Finally, my deepest gratitude belongs to my dear wife Sonja. Her unwavering support and patience have been my anchor throughout this endeavor. Her continued encouragement has been both a comfort and a driving force.

Karlsruhe, December 2025 *Martin Trat*

Contents

List of Figures

List of Tables

Lists of Abbreviations, Symbols and Notations

Acronyms

ADWIN	Adaptive Windowing
ARFR	adaptive random forest regressor
CDDE	concept drift detection ensemble
CRISP-DM	Cross-Industry Standard Process for Data Mining
CRISP-ML(Q)	Cross-Industry Standard Process Model for the Development of Machine Learning Applications with Quality Assurance Methodology
DDM	Drift Detection Method
DT	delivery time
EFER	early-find-early-report vote
Exp	experiment
SAM-CDAR	stream-applicable method for concept drift adaptation in regression scenarios
HTR	Hoeffding tree regressor
LT	lead time
MAE	mean absolute error
MAPE	mean absolute percentage error
MBPD	mean batch processing duration
MSE	mean squared error

MV	majority vote
PAISE	Process Model for Artificial-Intelligence-Systems Engineering
PH	Page-Hinkley test
RFR	random forest regressor
RMSE	root mean squared error
SBD	subject behavior diagram
SID	subject interaction diagram
SRPR	streaming random patches regressor
STD	standard deviation
TV	threshold vote

Symbols

a	accretion rate
b	batch size
D	base concept drift detector
$\mathbf{D}$	set of base concept drift detectors
E	ensemble construct
f	algorithmic model
M	base machine learning estimator
n	number of members in an ensemble construct
o	offset
O	complexity
P	probability
$\mathbf{R}$	set of reactive members of an ensemble

r	reactive member of an ensemble
s	stable member of an ensemble
S	strategy
w	weight
X	data queue
$\vec{x}$	multidimensional data point
Y	label queue
y	label value
z	number of bins for discretization
Δ	confidence
θ	parametrization
κ	maximum training data volume
μ	maturity threshold
τ	threshold

General Indices and Notations

agg	referring to data aggregation
$class$	referring to machine learning classification
det	referring to concept drift detection
$dist$	referring to data distribution
est	referring to machine learning estimation
$i,\ j$	running index
m	quantity
max, min	referring to a maximum or minimum
$t,\ u,\ v$	time

Mathematical Notations

$\mathbb{N}$	set of natural numbers
$\mathbb{R}$	set of real numbers
∞	infinity
$X,\ x$	variable
$\mathcal{X}$	random variable
$\vec{x}$	vector
$\mathbf{X}$	set
$\lvert\vec{x}\rvert$	cardinality of a vector
$\lvert\mathbf{X}\rvert$	cardinality of a set
$\hat{y}$	predicted value

1 Introduction

Artificial intelligence and machine learning have become enabling technologies for the digital transformation of industrial domains. In manufacturing, logistics, healthcare and beyond, the capability to extract value from continuously produced data offers great potential for process optimization and decision support among others. Developing and operating machine learning systems, i.e. the software contexts operating machine learning estimators, that remain performant in the long run, however, is far from trivial. The path from data to productive applications requires the investment of substantial efforts regarding machine learning estimator development and experimentation as well as careful system engineering.

Practically oriented frameworks such as the Process Model for Artificial-Intelligence-Systems Engineering (PAISE), which is presented in Figure 1.1, have been introduced to guide these activities in an end-to-end fashion within industrial contexts. Despite being among the more recently proposed instances, PAISE still lacks targeted guidelines for sustainably integrating and maintaining estimators in production environments. More specifically, it relies on handovers between development and operation teams, possibly across company boundaries, and does not provide explicit measures ensuring the long-term robustness of deployed estimators (Hasterok et al. 2021). This shortcoming also extends to other recent process model representatives. As a result, the aforementioned maintenance-related issues but also the question of concrete personnel responsibilities remain largely unresolved.

1.1 Motivation

In industrial practice, the deployment of machine learning estimators represents merely an early phase of their lifecycle. Once exposed to real-world data streams, i.e. potentially endless data series instead of complete datasets, numerous challenges arise that can critically undermine their performance. Among these are perturbations such as noise, outliers, and especially the phenomenon of concept drift, which describes changes in the statistical relationship between input data and target variables. If not properly addressed, concept drift can cause performance decline in estimators and may eventually render them unusable (Gama et al. 2014, Webb et al. 2016). This problem can be interpreted as another type of hidden technical debt (Sculley et al.

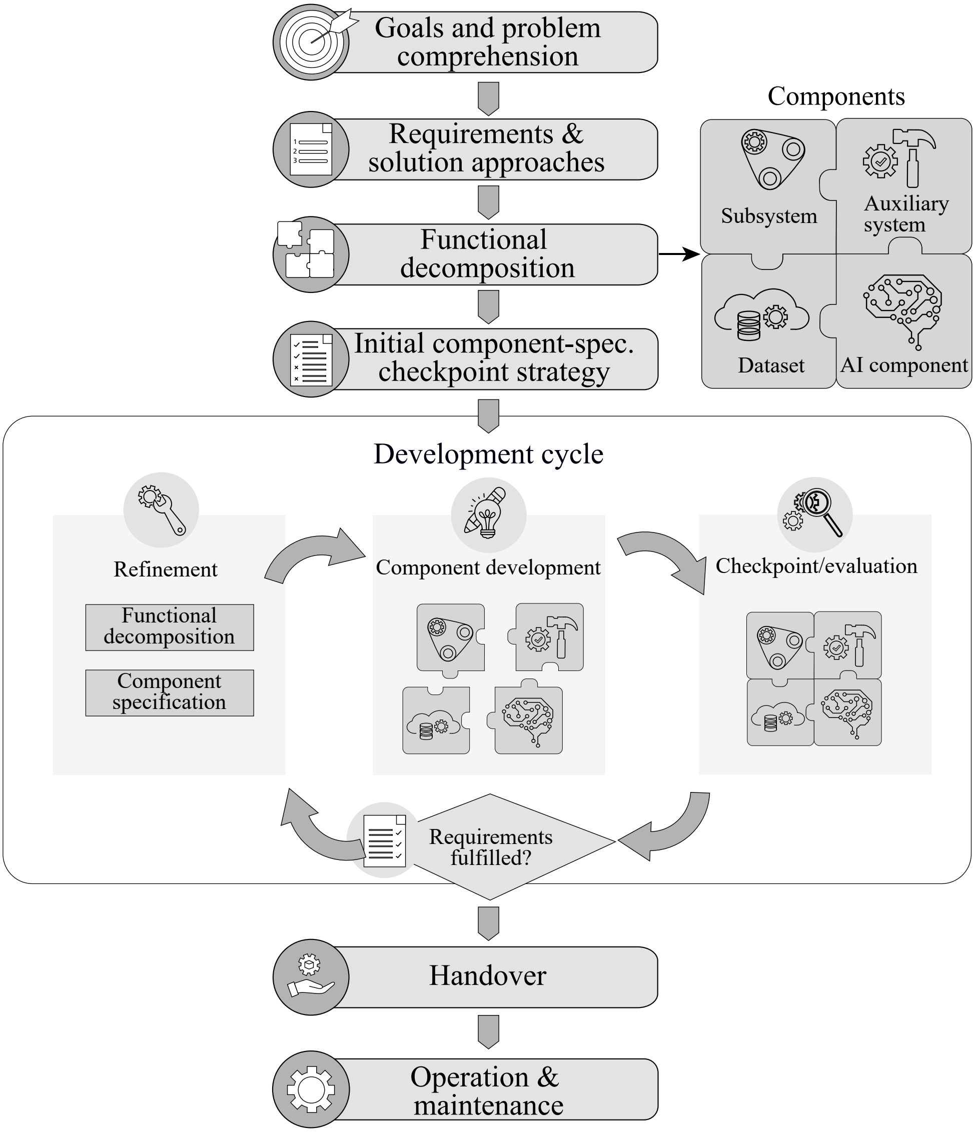


Figure 1.1: The Process Model for Artificial-Intelligence-Systems Engineering (PAISE). It aims at guiding design activities of entire machine-learning-based systems for practical use cases in an end-to-end fashion. Figure based on Hasterok et al. (2021).

2015) and severely complicates the successful adoption of machine learning, especially for small and medium-sized enterprises.

Reports from recent years indicate that this problem is still widespread across various industrial domains. Although many organizations succeed in training high-performing estimators, only a fraction of them manage to sustain this performance in production. Typical obstacles include limited access to siloed-away historical as well as live data, changes in compute environments, missing strategies for model maintenance and the availability of sufficiently skilled personnel (Algorithmia Research 2020, Anaconda Inc. 2021). The few successful companies often attribute their achievements to clear procedures for continuously updating their machine learning systems (Chui et al. 2022).
A similar pattern emerges with regard to the currently trending adoption of generative artificial intelligence, as companies often struggle to realize the expected return on investment. There is evidence that the employed models tend to stagnate in performance, which may be due to them being stuck to often outdated knowledge and simultaneously highly difficult to adapt (Challapally et al. 2025).

Addressing this issue requires methods that foster robust productive machine learning. More specifically, this demands for systems capable of maintaining high estimation quality despite non-stationary data and operational variability. This thesis aims to contribute to this goal by proposing concept drift adaptation approaches that strengthen the long-term robustness of machine learning estimators and the systems operating them. A particular focus is placed on regression problems, as these very frequently occur in industrial contexts yet remain under-represented in research. To this end, this thesis makes extensive use of machine learning ensemble methods, which exploit diversity among multiple model components to improve resilience against concept drift and other disturbances. At the same time, it challenges the associated literature's prevailing, often self-imposed and arguably unfounded constraint that algorithms need to fulfill extreme demands for computation efficiency. Instead, the impact of using such methods is examined in real-world production environments, which often provide ample but underutilized compute resources (Read and Žliobaitė 2023).

1.2 Practical Challenges

Despite notable advances in the research fields on concept drift detection and adaptation, several practical and scientific challenges remain unresolved. These challenges form the basis of the problems addressed in this thesis and are outlined in the following. They are closely tied to dynamically evolving real-world industrial scenarios, in which machine learning regressors are tasked to estimate process step durations in a free-form forging company and delivery durations of furniture products.

Modeling productive machine learning systems. As machine learning becomes an integral part of industrial processes, system architectures need to be both technically sound and documented in a form devoid of ambiguities. However, existing modeling approaches for describing machine learning systems often lack in this regard. Such shortcomings hinder a shared understanding among developers, operators and domain experts. In the examined industrial scenarios, this becomes evident when integrating estimators into heterogeneous production environments with varying data acquisition and organizational responsibility contexts. Also, the systematic integration of robustness-related aspects is proving particularly difficult.

Effectively detecting concept drift in practice. While a range of concept drift detectors exist, selecting and applying them within real-world scenarios is difficult (Barros and Santos 2018). Different detectors exhibit varying sensitivities and reaction times depending on the type and magnitude of concept drift, and no single detector performs best across all problems. This difficulty becomes particularly apparent in regression problems, which are prevalent in various industrial contexts such as manufacturing and logistics, yet remains severely under-researched (Lima et al. 2021, 2022). For instance, in the aforementioned forging and furniture trade scenarios, already subtle variations in material characteristics, process timing, or logistical conditions can result in abruptly or gradually altered data distributions. If untreated, this may lead to declining model performance.

Maintaining estimator robustness through explicit concept drift adaptation. Even if accurate concept drift detection were readily available, deciding on how to adapt estimators remains a complex task. Explicit concept drift adaptation strategies characterize the detector-informed approach to sustain the robustness of estimators. Suitable methods for combining both detection and adaptation in an integrated, model-agnostic manner and approaches to aptly parametrize them are scarce. This gap is also particularly evident in real-world regression scenarios, characterized by often evolving data and environmental conditions (Lima et al. 2022). As concretely observable in the production environments of the industrial scenarios addressed in thesis, maintaining reliable estimations of process and delivery durations is challenging. Achieving this requires not only timely concept drift detection but also targeted adaptation measures that preserve regressor stability.

1.3 Research Questions and Contributions

This thesis is the first to address a range of research aspects within the context of concept drift handling in real-world industrial scenarios. On the one hand, a set of research questions govern the scientific approach of this thesis, bringing forth several novel contributions to science. On the other hand, several contributions are documented in this thesis in order to demonstrate the research progress achieved by it. Both aspects are addressed in detail below and additionally visualized in Figure 1.2.

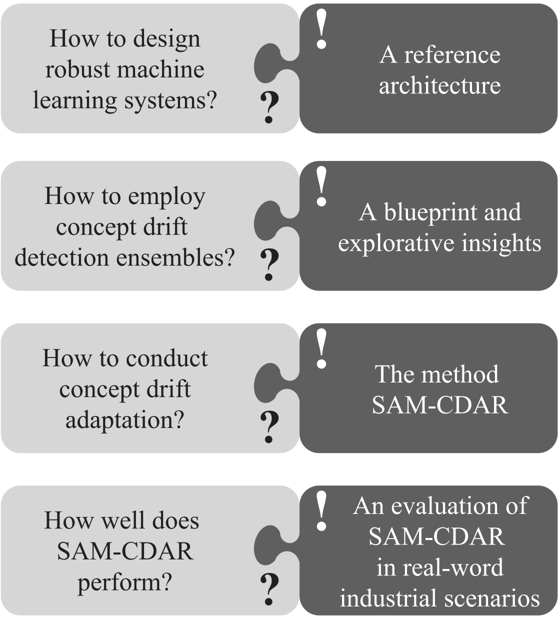

How to design robust machine learning systems?
?
!
A reference architecture
How to employ concept drift detection ensembles?
?
!
A blueprint and explorative insights
How to conduct concept drift adaptation?
?
!
The method SAM-CDAR
How well does SAM-CDAR perform?
?
!
An evaluation of SAM-CDAR in real-word industrial scenarios

Figure 1.2: The research questions and respective contributions proposed by this thesis.

1.3.1 Research Questions

The research questions of this thesis are outlined on a high level in the following. Throughout the chapters of this thesis, to facilitate addressing them in a profound way, they are operationalized. This is done through the assignment of more fine-grained sub-questions to each research question and the pursuit of their answers via gathering associated evidence and insights.

Initially, the research questions **RQ I** and **RQ II** are raised. Their aim is to critically evaluate the suitability of the model-centric reference architecture approach for addressing machine learning robustness problems prevailing in practice by supporting machine learning system design with an emphasis on long-term robustness.

RQ I What are the challenges in sustaining the robustness of productive machine learning estimators from a practical model-centric perspective?

RQ II Can a reference architecture support to sustain robustness of machine learning estimators from a practical model-centric perspective?

Robust machine learning systems can be further improved by employing concept drift detection ensembles (CDDEs), which can, in essence, be understood as diversity-enhanced structures of concept drift detectors. Research questions **RQ III** and **RQ IV** therefore aim at creating a fundamental understanding of these structures as well as of best practices for their design:

RQ III How extensively are concept drift detection ensembles studied?

RQ IV How can one design concept drift detection ensembles?

The responses to the previously stated research questions generate insights that are extended towards their application in real-world data streams by this thesis. Continuing the emphasis on machine learning robustness, **RQ V** is posed and primarily influences the design of a method apt for regression problems of the industrial practice.

RQ V How can one sustain the robustness of practically applied machine learning regressors against concept drift?

Finally, **RQ VI** puts the aforementioned method to the test. Observations of its performance in real-world concept-drifting industrial scenarios are gathered and evaluated.

RQ VI To what extend does the proposed method sustain the robustness of productively applied machine learning regressors in real-world evolving industrial data streams?

1.3.2 Research Contributions

In alignment with the above-introduced research questions, based on condensed existing knowledge from the literature, this thesis suggests several novel contributions to science. These are conceptualized as well as evaluated within the context of various industry research projects as outlined throughout this thesis. This ensures their practical relevance and makes them available for further practical use. The most notable contributions are outlined in the following.

A reference architecture for robust machine learning systems. A review of existing literature suggesting models that describe machine learning systems with concept drift adaptation components reveals an urgent need for a more coherent modeling approach. This thesis therefore proposes a reference architecture and provides it as an artifact that practically supports the design of robust systems. Furthermore, it can complement data science process models while simultaneously introducing front-loading capabilities. This thesis also provides guidelines for the reference architecture's specialization to further practical use cases and integration in organizational contexts.

The development of CDDEs. With this thesis being the first work to review the state of science on CDDEs, it lays the groundwork for further research. Resulting from a systematic analysis of existing publications, a definition, taxonomy and insights into the scientific progression of CDDEs is contributed to the body of research. Additionally, this thesis provides a structural perspective on their design in the form of a blueprint comprehensively outlining relevant design aspects.

A method for sustaining the robustness of productively applied machine learning regressors. Based on aforementioned artifacts, namely the reference architecture and the CDDE blueprint, combined with a nature-inspired explicit concept drift adaptation approach, a novel method is suggested: The stream-applicable method for concept drift adaptation in regression scenarios (SAM-CDAR). It consists of a CDDE that is developed to be applicable for regression scenarios and able to detect robustness threats of regressors, as well as a machine learning ensemble method for continuously updating such to represent volatile concepts via mechanisms imitating natural processes of evolutionary biology. Internalizing principles of robust machine learning systems by utilizing the reference architecture, it is apt to be utilized for data stream mining. Furthermore, its components can also be employed in a standalone fashion for concept drift detection and regression, respectively.

The experimental evaluation of SAM-CDAR in real-world industrial scenarios. This thesis documents in a detailed fashion how the proposed novel method SAM-CDAR is applied in data streams to sustain the robustness of regressors. Simultaneously, an approach to determine its suitable parametrization is suggested. The subsequent careful evaluation of the obtained results involves the thorough measurement of SAM-CDAR's performance. This way, its feasibility in real-world industrial problems is validated, while considering the compute restrictions and requirements that apply within such environments.

1.4 Outline

The following paragraphs provide an overview of the structure of this thesis by outlining the main content aspects of all following chapters. Additionally, remarks are made for chapters containing research that is separately published. Figure 1.3 additionally visualizes this structure.

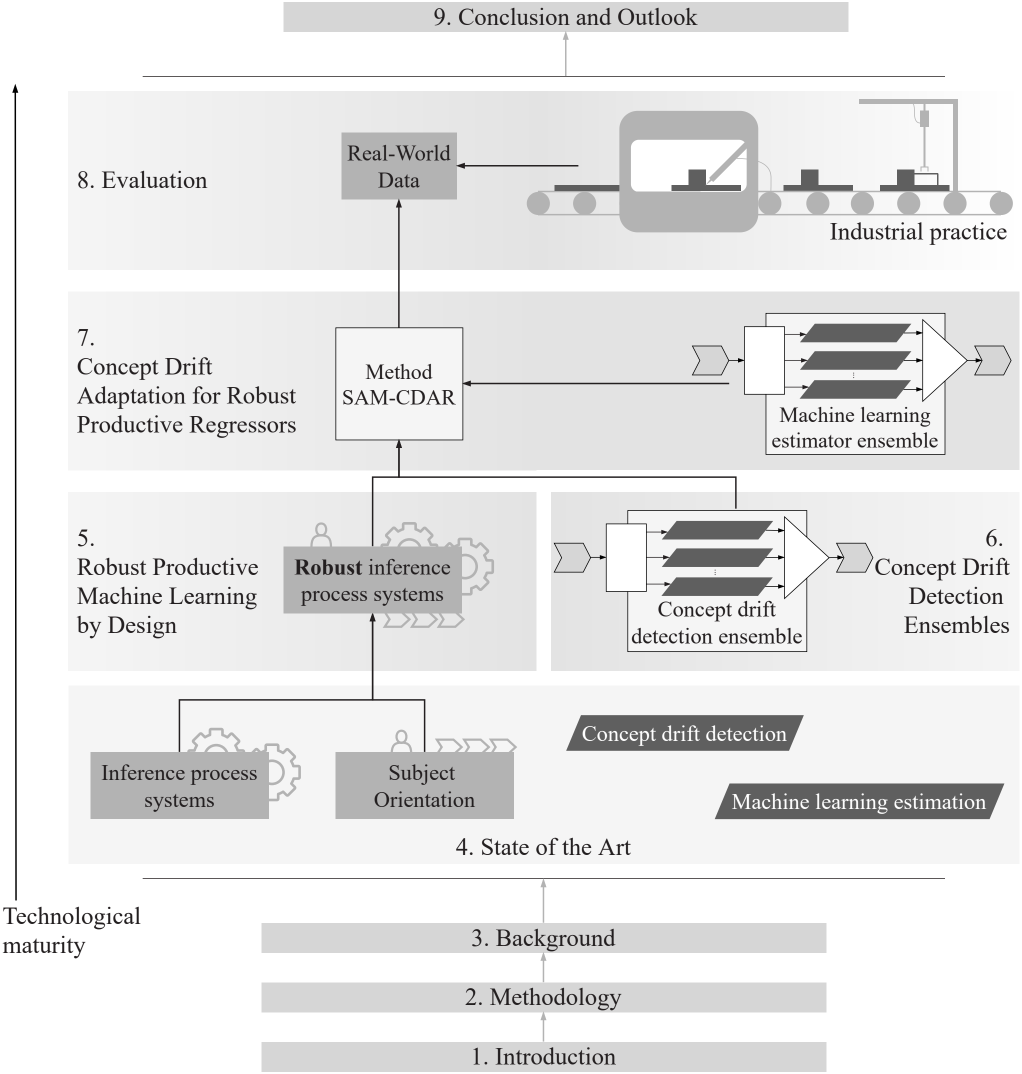

9. Conclusion and Outlook
8. Evaluation
Real-World Data
Industrial practice
7.
Concept Drift Adaptation for Robust Productive Regressors
Method SAM-CDAR
Machine learning estimator ensemble
5.
Robust Productive Machine Learning by Design
Robust inference process systems
Concept drift detection ensemble
6.
Concept Drift Detection Ensembles
Inference process systems
Subject Orientation
Concept drift detection
Machine learning estimation
4. State of the Art
Technological maturity
3. Background
2. Methodology
1. Introduction

Figure 1.3: Schematic representation of the structure of this thesis. It visualizes the interaction of its chapters and its effort to advance a set of technologies to higher levels of maturity. The numbers correspond to the chapter numbering.

Methodological foundations of this thesis, a contextualizing background and the relevant state of the art are provided in Chapters 2, 3 and 4, respectively. Subsequently, the following Chapters 5 through 8 constitute the scientific core of this thesis.

In Chapter 5, basic robustness and machine-learning-system related formalizations are provided for use throughout this thesis. Additionally, several aspects relevant with regard to the reference architecture approach for the design of robust systems are presented. This chapter extends and contextualizes the following publication:

> Martin Trat, Matthes Elstermann, Jana Deckers, and Jivka Ovtcharova. Modeling a Reference Architecture for Concept Drift Adaptation Systems. In Tung Bui, editor, *Proceedings of the 58th Hawaii International Conference on System Sciences*, Proceedings of the Annual Hawaii International Conference on System Sciences. Hawaii International Conference on System Sciences, 2025. doi: 10.24251/HICSS.2025.121[1]

Chapter 6 provides a detailed and profound introduction of CDDEs, outlines and analyzes existing research and offers guidelines for their careful design. Parts of the research presented in this chapter are also included in the following publication:

> Martin Trat and Jivka Ovtcharova. Designing Concept Drift Detection Ensembles: A Survey. In *2023 IEEE 10th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10. IEEE, 2023. ISBN 979-8-3503-4503-2. doi: 10.1109/DSAA60987.2023.10302492

Chapter 7 integrates the methods explored in previous chapters to form a novel one. It is designed to achieve high concept drift detection performance and to sustain the robustness of machine learning regressors.

Chapter 8 validates the novel method in real-world industrial use cases via conducting a series of experiments. The following publication by the author of this thesis introduces a component of this method, which is augmented in Chapter 7, and evaluates this component on real-world data streams:

> Martin Trat, Philipp Bergmann, Andreas Ott, and Jivka Ovtcharova. A Nature-Inspired Concept Drift Adaptation Method for Industrial Data Stream Regression. In Yi-Chi Wang, Siu Hang Chan, and Zih-Huei Wang, editors, *Flexible Automation and Intelligent Manufacturing: Manufacturing Innovation and Preparedness for the Changing World Order*,

[1] Nominated as best paper of the 58th Hawaii International Conference on System Sciences 2025.

Lecture Notes in Mechanical Engineering, pages 3–13. Springer Nature Switzerland, Cham, 2024. ISBN 978-3-031-74481-5. doi: 10.1007/978-3-031-74482-2_1

Lastly, this thesis is concluded in Chapter 9. This chapter also provides an outlook on potential future work in its scientific fields.

For selective approaches to walk through this thesis, Figure 1.4 provides a dependency graph resolving how the Chapters 2 – 8 build upon each other. For instance, if one desires to study the reference architecture for robust inference process systems introduced in Chapter 5, one should priorly familiarize oneself with methodic foundations applied for surveying associated literature as well as the description method Subject Orientation provided in Chapters 2 and 3, respectively.

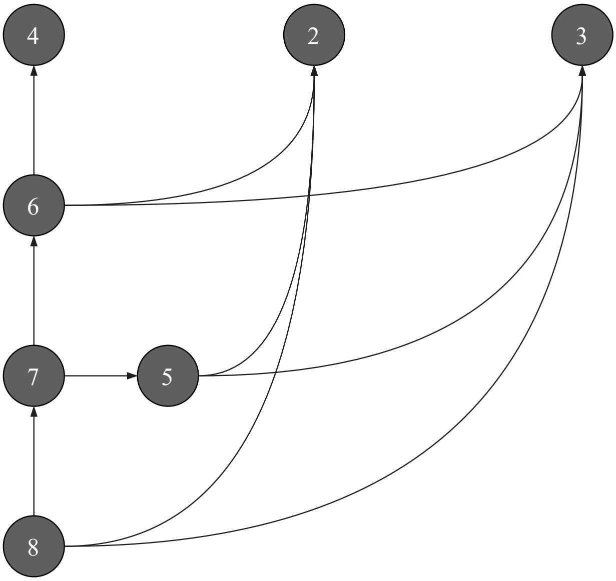


Figure 1.4: Dependency graph representation of the Chapters 2 – 8 of this thesis. The numbered nodes reflect the chapter numbering while the directed edges represent dependency relations, each rooted in the node possessing the dependency and pointing to the one that it depends on.

2 Methodology

The scientific approach of this thesis is guided by a set of research questions, which are defined in Section 1.3.1. Several principles and methods that support this approach in a methodologically sound and consistent fashion are introduced and detailed in this chapter. These involve the selected and refined methods for systematically extracting and condensing insights from scientific literature as well as for aggregating such in model-based forms, which are introduced in Section 2.1 and 2.2, respectively. The principles relevant for experimental evaluations conducted in this thesis are concretized in Section 2.3.

2.1 Systematic Literature Review

Systematic literature reviews conducted for various matters throughout this thesis consistently follow a three-step process: Gathering potentially relevant publications, filtering out irrelevant ones and analyzing the remaining ones. In its fundamentals, this approach is based on Tranfield et al. (2003). It is visually outlined in Figure 2.1.

Gathering

During the initial step, relevant publications are retrieved from a suitable literature database. For this purpose, it is suitable if it exclusively contains high-quality peer-reviewed research. Typically, indexed parts of stored publications, such as the title, abstract, keywords, or full text are matched with a search query string, defined by the user. Therefore, terms relevant for the respective matter are assembled to such a string. Optionally, the search query is augmented with additional conditions. Such may, for instance, limit the publication year to reduce the number of works to a manageable quantity. Applying the search query, a set of publications is produced that must then be filtered based on specific content requirements.

Filtering

The objective of this step is to reduce the set of publications retrieved in the previous step by discarding those not relevant to the respective matter. In order to ensure consistency, a set of content requirements is defined. By reading selected parts of the retrieved publications and comparing the obtained insight with the content requirements, a decision can be made as to whether it should be included in the set of relevant publications. To consider as much relevant research as possible, the exclusion of publications from further consideration is done carefully: In case its content focus appears vague or an association to the respective matter can not be ruled out with certainty, it is retained within the set of relevant publications. Furthermore, filtering can be conducted in an iterative manner, beginning with relatively abstract parts of a publication, such as the abstract, and progressing to more specific and comprehensive sections. This way, ambiguities concerning a publication's relevance can be reduced. During this step, the query string from the preceding one may also be refined based on the accumulation of knowledge about the matter. This may result in a higher density of relevant publications among the initially gathered ones.

Analysis

Finally, all publications identified as relevant are subject of analysis. The type of analysis varies and is selected depending on the respective matter. If feasible, data can e.g. be extracted from the relevant publications based on a set of analysis criteria. The method for compiling such is based on Webster and Watson (2002). Otherwise, insights from the publications can be obtained in a qualitative manner.

2.2 Model-Based Description

Several contributions of this thesis are derived by following structured processes of aggregating knowledge retrieved from existing bodies of scientific literature. This aims at enhancing understanding and generating insights by systematically analyzing principles of the respective target research subjects as well as their presentation in a manner that maximizes the utility for the scientific community.

To achieve this, generated insights are organized and refined based on carefully applying model-based perspectives or methods. This way, insights become embedded within structured, condensed, and persistent artifacts. Additionally, they are designed to facilitate further research activities, such as developing novel methods or algorithms, by providing a clear theoretical foundation and structure.

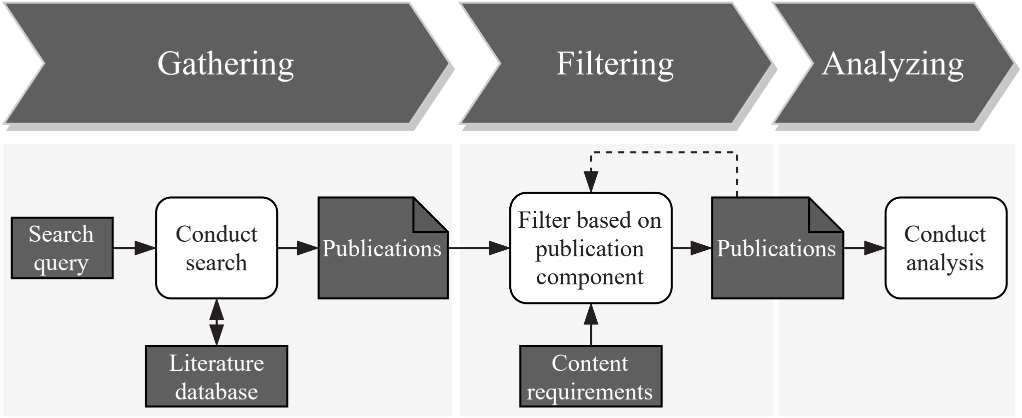


Figure 2.1: Flowchart representation of systematic literature reviews applied in this thesis. Each review is a process that starts with gathering publications from a literature database by applying a search query. Filtering of retrieved publications can be done iteratively, which is represented by a dashed arrow. It involves obtaining insight from reading increasingly comprehensive parts of publications. Finally, the set of relevant publications serves as basis for various analysis approaches.

The specific instances of model-based artifacts proposed in this thesis are the following:

- A reference architecture for characterizing, analyzing and implementing complex robust inference process systems (cf. Chapter 5).
- A schematic blueprint for conceptualizing and parametrizing an algorithmic construct (cf. Chapter 6).

2.3 Experimental Evaluation

In the following, basic principles for experimentally evaluating novel algorithmic methods on real-world data are outlined. These are applied in Chapter 8, which furthermore provides respective parametrizations to ensure the reproducibility of all conducted evaluations.

Performance Scoring

In this thesis, proposed as well as baseline methods are scored by consistently computing comparable performance metrics. In order to match the properties of data stream mining, prequential evaluation is selected as scoring framework. It can be characterized as a test-then-train approach following a deliberately selected order of evaluation and training-related steps:

1. Once a new data batch arrives in a data stream, the estimator, in its current state, receives it as input and computes an output.
2. Assuming immediate access to labels for evaluation purposes, the output is used to compute a performance metric value related to the new data batch.
3. The performance metric value can be included in calculating an aggregated performance metric. The latter quantifies the estimator's long-term performance in an incremental fashion, e.g. as a data-stream-global or a sliding window average.
4. The new data batch can be utilized for estimator development.

This approach ensures that evaluation consistently occurs on data that the estimator has not seen before. Yet, the same data can still be employed for training and is not wasted. As a result, a holdout data purely utilized for testing is not required (Wares et al. 2019, Bifet et al. 2018). This evaluation framework is visualized in Figure 2.2.

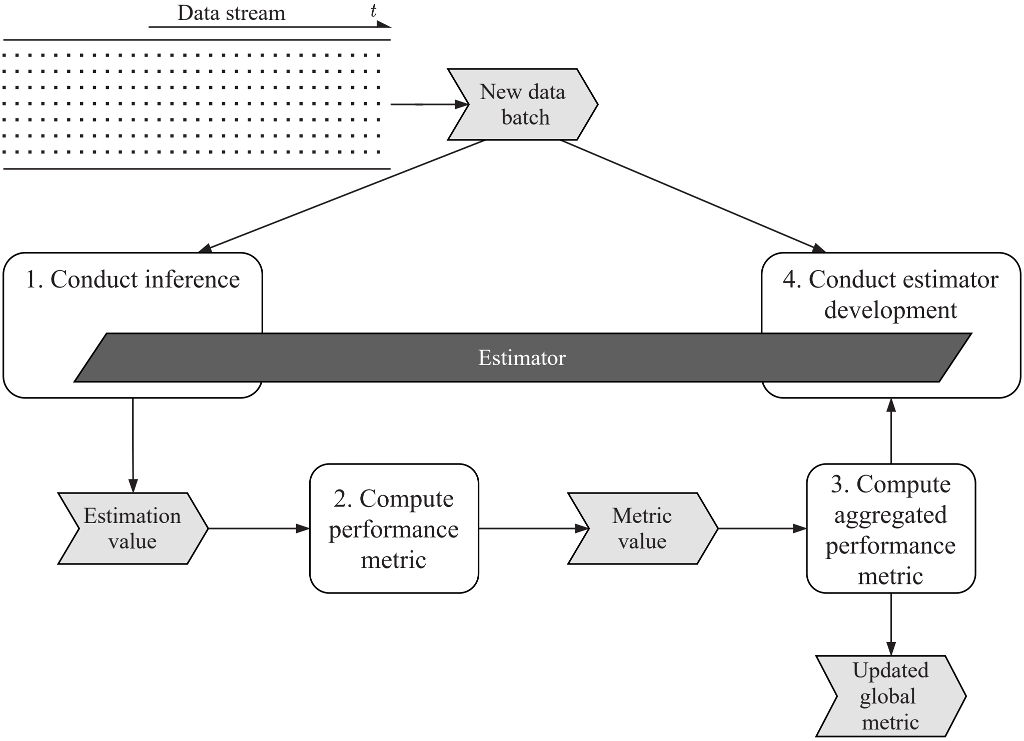


Figure 2.2: Prequential evaluation in a test-then-train fashion. To ensure economic data usage and sound evaluation, a new data batch is first used for performance scoring and then for estimator development.

Analysis via Optimization

The primary goal of experiments conducted in this thesis is to enable carrying out a detailed analysis of concept drift adaptation capabilities of various methods. As parametrization of all considered methods is crucial for this goal, hyper-parameter optimization is selected as experimental approach. Based on a pre-defined set of parameters, associated boundary conditions and a target metric, it approximates a function that supports the search for an optimal parameter configuration. Also, based on obtained results, insights into the performance range as well as associated bounds can be gained (Bergstra and Bengio 2012).

To diligently ensure the analysis of the full combinatorics of all parameter configurations, hyper-parameter optimization is consistently applied in a grid-search-based fashion in this thesis. Certain benefits from alternatively applying a randomized search are conceivable (Bergstra and Bengio 2012). However, despite possibly being costly, the former guarantees the exhaustion of a parameter space that is manually defined via specifying the ranges of all parameters that are varied throughout an experiment.

2.4 Summary

This chapter concisely informs the reader about the methodological foundations of this thesis. Several methods and principles that relate to gathering and formalizing knowledge as well as conducting experimental evaluations are introduced in an abstract fashion. Beyond outlining individual procedures, this chapter establishes a coherent methodological framework that guides the scientific approach of deriving evidence from various sources of information as well as the structured documentation of resulting insights.

More specifically, by consolidating the selected approaches for systematic literature reviews, model-based knowledge aggregation and experimental evaluation, this chapter provides a consistent basis for the subsequent research contributions. Consequently, this lays the groundwork for Chapters 5 through 8, which apply and further specify these methods in concrete contexts. In doing so, the methodological principles presented here are instantiated on real-world data, adapted to the respective research objectives and ultimately serve to support the development, analysis and validation of the contributions of this thesis.

3 Background

This chapter introduces theoretic aspects that are built upon in later ones. It begins with laying out basic machine learning fundamentals in Section 3.1 and extends these into more advanced contexts. On the one hand, these fundamentals are elevated to production-relevant data stream environments in Section 3.2. In the same course, associated boundary conditions that may be challenging due to non-stationary properties are outlined. On the other hand, it considers these aspects within the context of data science projects and guiding principles for associated processes in Section 3.3. To additionally support descriptive approaches within this thesis, the rigorous modeling method Subject Orientation is introduced in Section 3.4. Throughout this chapter, classification-related aspects are consistently yet briefly referred to. Emphasis, however, is placed on regression problems due to their high relevance in this thesis.

3.1 Machine Learning Algorithmics

Machine learning is among the most prominent disciplines to generate added value from information bound to data. This section introduces a range of associated theoretic concepts. These include estimators as algorithmic structures internalizing data patterns in a model-based form, approaches to combine such in ensembles for more robust outputs as well as to measure their performance.

3.1.1 Estimators

The term machine learning estimator is employed as an umbrella term for classifiers and regressors in this thesis. The following introduces theoretical aspects of their formalization, exemplary representatives and training strategies.

Basic Principles and Prominent Algorithms

A machine learning estimator M applies an algorithmic learning model f^{est} to extract and internalize patterns from training data to solve an arbitrary task via inference on previously unseen data. In this context, input data can be described as a volume of multidimensional data points formalized as vectors $\vec{x}$. The dimensions of each $\vec{x}$ refer to the set of features it is defined over (James et al. 2023, p. 9), with each feature $x_j \in \mathbb{R}$ and $j \in \{0, ..., |\vec{x}| - 1\}$. Within the industrial domain, exemplary features may include material characteristics, processing temperatures, machine types or personnel attendance. Abstractly, a task can be formalized as an estimation problem $\vec{x} \mapsto y$, with data-associated labels y (James et al. 2023, pp. 15–25). The range of feasible tasks is abundant and spans e.g. object recognition, machine fault prediction or the estimation of physical properties (Mitchell 1997, pp. 2–5), (Awad and Khanna 2015, p. 4). Figure 3.1 depicts a schematic visualization of these estimator aspects.

A classifier is an estimator that provides a qualitative inference output. More specifically, it assigns data points $\vec{x}$ to typically discrete classes y, e.g. with $y \in \mathbb{N} \cup \{0\}$. Popularly employed algorithms f^{est} are Naïve Bayes, K-Nearest Neighbors and neural network classifiers (James et al. 2023, p. 27,135,136).
A regressor, in contrast, is an estimator providing quantitative inference outputs (James et al. 2023, p. 27). In associated problems, such can generally be formalized as continuous variables, i.e. $y \in \mathbb{R}$ (Awad and Khanna 2015, p. 67). Prominent representatives of regressor f^{est} are the tree-based methods random forest regressor (RFR) and Hoeffding tree regressor (HTR). The former describes an ensemble that aggregates the output of a specific number of regression trees while leveraging randomness to obtain robust estimations. Each tree is typically trained on a random sample of the training data considering only a feature subset by recursively partitioning the data at nodes based on feature value splits that optimize certain criteria (Breiman 2001). The latter is an adaptation of the regression tree optimized for use in fast flowing data streams. It supports the formation of nodes by applying the statistical Hoeffding bound measure (Gomes et al. 2020, Domingos and Hulten 2000).

Training Types

The f^{est} can be trained based on different types of algorithms as shown in Figure 3.1. They can be categorized as either batch-based or continuous. Mixed types of both exist (Read et al. 2012, Gama et al. 2014, Krawczyk et al. 2017).

Batch-based training requires the entire data mass to be passed to the f^{est} coherently in a single pass (James et al. 2023, pp. 15–18). Without any further strategies in place, no subsequent

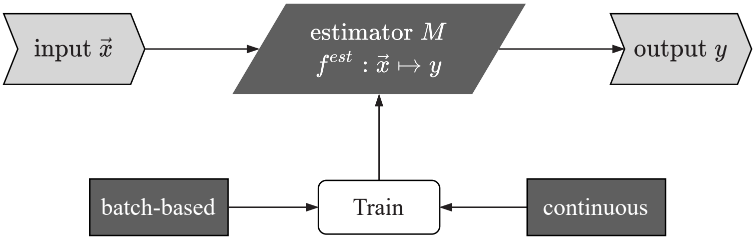


Figure 3.1: An abstract schematic visualization of a machine learning estimator M and feasible training types.

retraining is possible, thus permitting merely the static use of an estimator trained this way. Across many industrial domains, this is still common practice (Gupta et al. 2023).

Given the often evolving nature of data stream environments, as expanded upon further below, incrementally training or retraining estimators is among the most promising strategies to sustain their robustness. **Purely continuous**-training-type algorithms enable f^{est} to consecutively incorporate a notion of knowledge gain from sequentially received data points or batches. Nevertheless, the available selection of associated f^{est} is highly limited and they may require great volumes of labeled data to remain robust in data streams if no additional measures are in place[2] (Read et al. 2012).

To support the use of arbitrary, widely well understood batch-based f^{est} in data streams, retraining can be done in an **emulated-batch-continuous** fashion. It requires repeated from-scratch training of models on a theoretically infinitely growing buffered dataset. This, however, becomes infeasible at a certain point in practical scenarios due to resource constraints (Gama et al. 2014, Read et al. 2012).

3.1.2 Ensembles

Arriving at well-founded decisions can be achieved by bringing together experts with diverse specializations, i.e. significant knowledge in different specific areas. Such a gathering of individual persons can be referred to as a committee or ensemble. Examples of such can be found in a plethora of contexts, including political election scenarios, legal court juries, technical panel discussions, medical decision panels or scientific peer reviews. Transferred to the machine learning context, this approach can be leveraged (Polikar 2012, pp. 1–2). In the following, associated basic principles as well as algorithmic approaches for their compilation are introduced.

[2] Even the authors of a widely employed Python library for data stream mining express doubts whether such f^{est} are proper approaches for most use cases (`https://github.com/online-ml/river`, accessed: 01.06.2025).

Basic Principles

In the field of machine learning, ensembles can be formed by combining multiple base, i.e. combinable estimators and aggregating their individual outputs. In this thesis, such structures may be referred to as machine learning estimator ensembles, or simply ensembles for short. Based on Kuncheva (2004, p. 105) and Polikar (2012, p. 4), Definition 1 formally characterizes them, while Figure 3.2 complements it via a schematic visualization.

> **Definition 1** (Machine learning estimator ensemble)**.** A machine learning estimator ensemble is an algorithmic construct E^{est} combining n^{est} base machine learning estimators M_i as members differing in machine learning estimation algorithm f^{est} or parametrization θ, with $i \in \{1...n^{est}\}$. The construct employs an aggregation strategy S^{est}_{agg} that defines the consolidation of the output of each M_i into a single ensemble-wide output. Consequently, E^{est} can be considered an estimator M in its own right.

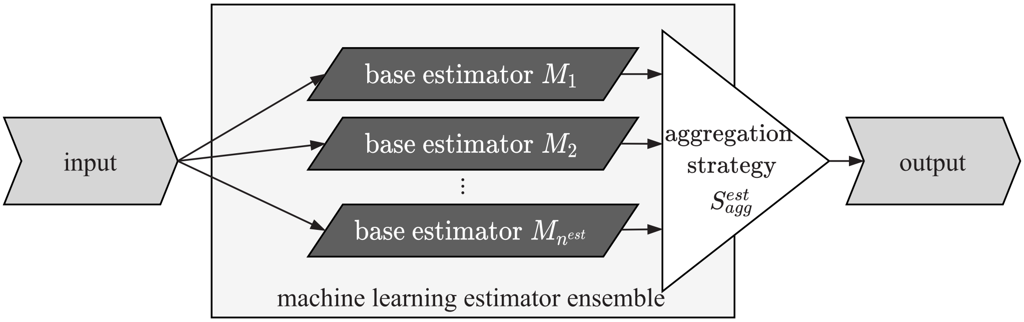


Figure 3.2: An abstract schematic visualization of a machine learning estimator ensemble E^{est}. It combines multiple base machine learning estimators M_i. The individual outputs of members are consolidated into an ensemble-wide output based on an aggregation strategy S^{est}_{agg}. Based on Krawczyk et al. (2017).

From a theoretical perspective, ensembling can be counter-intuitive at first glance. For instance, based on the Occam's razor principle, simple estimation models should be preferred over such characterized by higher model complexity (Witten et al. 2011, p. 351). However, in practice, ensembling has the potential to work very well. It can reduce the risk of ending up with an unfit single estimator and might even critically improve the estimation performance in a broad set of data and inference-related problems by reducing the estimation variance. For an artificial data classification scenario, this is depicted in Figure 3.3. Analogously, this reasoning holds for regressors as well (Polikar 2012, pp. 1–3) (Witten et al. 2011, pp. 353–355). Furthermore, the intuition behind ensembling is supported by the No-Free-Lunch theorem (Wolpert and Macready

1997). Simply put, it states that there is no single estimator that performs best on the set of all possible problems, but it may maintain a certain specialization. Ensembling can therefore improve the overall performance by combining estimators that counterbalance each other's weaknesses, i.e. their respective lack of specialization in various estimation-relevant aspects of a given problem. The goal, therefore, is to introduce diversity into an ensemble, e.g. manifesting in the difference of various estimators' algorithmic, feature-based, or model structural properties. This consequentially shifts responsibility to the user for carefully selecting a diverse set of estimators (Krawczyk et al. 2017).

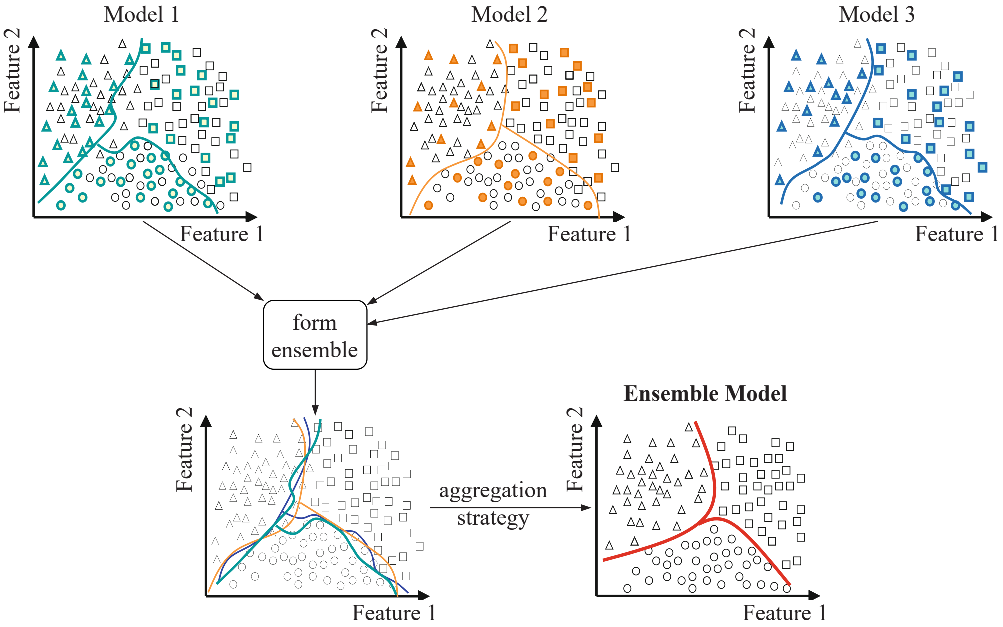


Figure 3.3: Reduction of estimation variance by combining multiple machine learning estimators in an ensemble. An artificial data classification scenario containing two features exemplifies this effect. Separate estimation models, depicted as data-separating boundaries in subplots Model 1 through 3, exhibit distinct specializations but commit certain mis-classifications, respectively. However, a mutual agreement on correct classifications holds in the case of a set of carefully selected estimation models. As a result introduced by averaging over this set, the ensemble-aggregated model, depicted as red data-separating boundary in subplot Ensemble Model, leads to an improved estimation performance. Based on Polikar (2012, p. 3).

However, if characterized by an exceedingly high model complexity, practical applicability may elude ensembles. As an example, a data science competition originated by the entertainment content streaming service Netflix[3]can be referenced. The highest-performing solution in that

[3] `https://www.kdnuggets.com/2015/06/ensembles-kaggle-data-science-competition-p2.html`, accessed: 04.10.2024

competition is an ensemble, however, containing hundreds of members. While, from a theoretical standpoint, this is an achievement, it implies significant drawbacks regarding its practical application. Firstly, such a configuration makes inference computationally expensive as a large number of computations need to be executed for each estimation. Secondly, it is difficult to analyze such an ensemble in detail, as e.g. performance analysis becomes increasingly complicated with an increasing number of members. Deriving measures to improve their performance then becomes a non-trivial task (Witten et al. 2011, p. 352).

Meta Algorithms

Apart from implementing ensembles based on Definition 1, one can alternatively form ensembles using meta algorithms. This way, data sampling schemes or individual base estimator's strengths can be exploited. They reduce the user's effort but simultaneously block her possibility to intervene with respect to a large set of ensemble design aspects. The most popular examples of ensemble meta algorithms are bagging, boosting and stacking.

Bagging. This technique primarily aims at reducing the variance of estimators. Its name is a portmanteau of the terms *bootstrap* and *aggregation*. Consequently, it involves bootstrapping, i.e. randomly sampling multiple equally sized subsets from the available training data with replacement (Polikar 2012, pp. 5, 12). Subsequently, a separate estimator is trained on each of these subsets. Already small differences in the sampled subsets can have a strong influence on the resulting estimators. This approach therefore exploits the instability of learning algorithms when applied to different data to produce and aggregate diverse members to form an ensemble. Typically, the members in the resulting ensemble are weighted equally (Witten et al. 2011, pp. 353–356). A popular approach applying bagging is the algorithm behind the random forest estimator, which has variants for classification and regression problems (Breiman 2001).

Boosting. In contrast to bagging, boosting involves differently weighing estimators of an ensemble. This is meant to promote the ensemble-wide impact of high-performing members and decrease that of low-performing ones (Witten et al. 2011, p. 352). Members are iteratively added to an ensemble with the goal of complementing the performance of their predecessors (Witten et al. 2011, pp. 358–362). From a theoretical standpoint, already weak estimators fulfil the requirement to become an ensemble member as long as they perform better than guessing at random. They are trained on subsets from the available training data, which are sampled based on a probability distribution favoring data points misclassified by members that are already in the ensemble (Polikar 2012, pp. 5, 13–15).

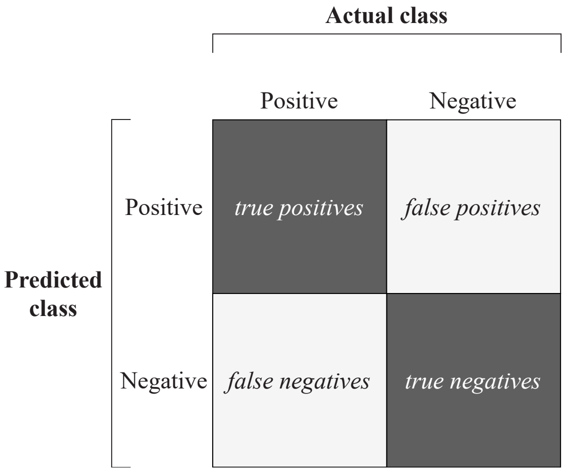
Stacking. This approach is applied by replacing the aggregation of member outputs based on vote and average-based approaches with a trainable, usually simple estimation model. Within this context, while ensemble members are referred to as tier-1 estimators, the aggregating estimator is referred to as tier-2 estimator. Stacking is different from the two previously outlined meta algorithms with respect to two major aspects: Firstly, this technique is highly suitable to combine tier-1 estimators using different models (Witten et al. 2011, pp. 369–371), which may introduce an additional source of diversity. Secondly, it unlocks the potential to achieve superior aggregation strategies by better processing the output of the tier-1 estimators. This can be argued as the tier-2 estimator leverages a machine learning algorithm to model the proficiency of the tier-1 estimators with respect to e.g. different regions of the feature space (Wolpert 1992), (Polikar 2012, pp. 15–16).

3.1.3 Performance Metrics

For a wide range of motivations, quantifying the performance of estimators or estimator ensembles is vital. Among others, these include tracking training progress, hyper-parameter optimization, and their monitoring during productive operation. Approaches to measure the performance via computing specific metrics of estimator correctness or error are introduced in the following.

Within the context of **classification** problems, correctness scores can be computed for estimators. Initially assuming the existence of solely two classes, namely an abstractly positive and a negative class, helps defining a range of core aspects that may seem to have only subtle differences yet serve different purposes. The feasible cases, i.e. correct and incorrect assignments to both classes, can be attributed to the quadrants of a confusion matrix, which is displayed in Figure 3.4. Certainly, these aspects can also be transferred to more complex classification problems (Fawcett 2006). The precision score describes the fraction of the number of data points correctly classified as positive and all actually belonging to the positive class. The recall considers also the number of data points correctly classified as positive but puts them into the relation of merely all classified as positive, correctly or not. The F_1 score is computed as the harmonic mean of both aforementioned scores. It can, however, be generalized to reflect different trade-offs between them. The widely employed accuracy score is computed as the share of data points correctly assigned to any class in relation to all data points. For concrete formulae, the reader is referred to e.g. Fawcett (2006) and Awad and Khanna (2015, pp. 2–3).

For **regression** problems, error scores are popularly considered. Similar ones can also be transferred to classification contexts (Mitchell 1997, pp. 128–130). This is, however, not considered here. In this regard, an error quantifies the difference between the actual regression target value y_u and its regressor-provided prediction $\hat{y}_u$, formalized as $y_u - \hat{y}_u$ with u representing the temporal context. Put into the relation of a volume of b data points, one can e.g. compute the mean



		Actual class	
		Positive	Negative
Predicted class	Positive	*true positives*	*false positives*
	Negative	*false negatives*	*true negatives*

Figure 3.4: The confusion matrix. In its quadrants, it localizes the feasible cases of a two-class classification problem. Based on Fawcett (2006).

absolute error (MAE) by averaging the absolute of the error over b (James et al. 2023, p. 437), as formalized in (3.1).

$$\text{MAE}(y_u, \hat{y}_u) = \frac{1}{b} \sum_u |y_u - \hat{y}_u| \tag{3.1}$$

Instead of the absolute, the mean squared error (MSE) resorts to squaring the error, which emphasizes large instances of such (James et al. 2023, p. 28):

$$\text{MSE}(y_u, \hat{y}_u) = \frac{1}{b} \sum_u (y_u - \hat{y}_u)^2 \tag{3.2}$$

By taking the square root of the MSE, the root mean squared error (RMSE) computes an error notion sharing the same unit of the MAE, as shown in (3.3) (James et al. 2023, p. 28), (Hyndman and Koehler 2006).

$$\text{RMSE}(y_u, \hat{y}_u) = \sqrt{\text{MSE}(y_u, \hat{y}_u)} \tag{3.3}$$

The mean absolute percentage error (MAPE), in contrast, pursues another approach by considering the fraction of the error and the actual target value in its core, as formalized in (3.4). It may be scaled with 100 to emphasize its notion being remotely similar to a percentage (Hyndman and Koehler 2006).

$$\text{MAPE}(y_u, \hat{y}_u) = \frac{1}{b} \sum_u |\frac{y_u - \hat{y}_u}{y_u}| \tag{3.4}$$

3.2 Evolving Data Streams

Especially in the industrial context, an increasing number of processes attain increasing levels of digitization. This leads to highly heterogeneous data being generated in rough quantities at great velocities. The various disciplines around mining such data streams using machine-learning-based approaches require employed models to robustly process them indefinitely (Žliobaitė et al. 2014, 2012). Despite considerable effort and resources being invested in developing and deploying such approaches for productive use, many fail as they are designed with insufficient awareness of the evolving characteristics of such streams (Chui et al. 2022, Anaconda Inc. 2021).
In the following subsections, formalizations and properties but also risks and challenges that are associated with data streams are introduced. Also, based on a formal characterization of concepts within data streams, the foundations of concept drifts as well as means to identify and handle their destructive potential towards machine-learning-based approaches are illustrated.

3.2.1 From Datasets to Data Streams

Training machine learning estimators to perform specific tasks typically requires data. It represents a resource that is used to build up experience (Mitchell 1997, pp. 2–6). If a considerate data volume is statically available prior to training, it can be referred to as a dataset. It can be formalized as a set $\mathbf{X}$ of multidimensional data points $\vec{x_i}$ defined over a set of features, with $\vec{x_i} \in \mathbf{X}$ and running index $i \in \{0, ..., |\mathbf{X}| - 1\}$. For supervised training, labels y_i, with $y_i \in \mathbf{Y}$, assigned to the $\vec{x_i}$ are required.

While static datasets can be employed for the experimental estimator development, data streams become increasingly wide-spread in application environments. Data streams can be formalized as $\vec{x_t} \sim \mathcal{X}$, i.e. potentially infinite time series of multidimensional data points $\vec{x_t}$ drawn from a random variable $\mathcal{X}$ at various timestamps t, with $t \in \{0, ..., \infty\}$. Analogously, labels can be understood as values y_t drawn at various t from a random variable $\mathcal{Y}$ (Webb et al. 2016), formalized as $y_t \sim \mathcal{Y}$. As an example, one might consider a sensor measuring the temperature of a machine component. It continuously produces data over its lifetime that can, among other features, serve as input $\vec{x_t}$ for a machine learning model. A label y_t could then be a classification of the machine component's current health state. The values for y_t could then be such that represent the state being e.g. functional or impaired. Labels can e.g. be provided by experts other other trusted entities, such as high-quality sensors. The difference between a static dataset and a data stream is additionally visualized in Figure 3.5.

In general, from the perspective of a receiving entity, data or labels can arrive in batches, which can be formalized as arbitrarily sized sets. In this regard, a single data point $\vec{x_t}$ or label y_t can be viewed as a special case of a data batch $\mathbf{X}_t$ or label batch $\mathbf{Y}_t$ with a size $|\mathbf{X}_t|$ or $|\mathbf{Y}_t|$ of 1. For notational consistency, the remainder of this thesis uniformly refers to chunks of streaming data as batches.

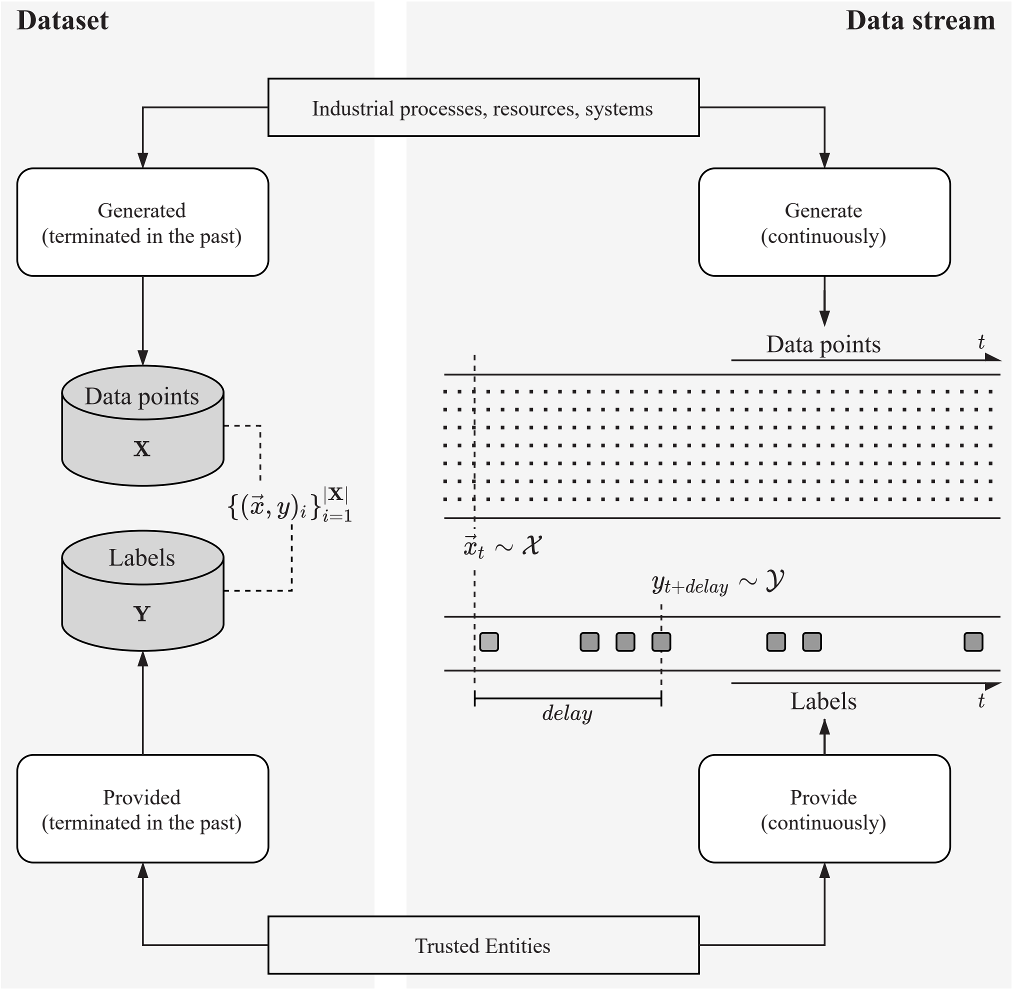

Dataset
Data stream
Industrial processes, resources, systems
Generated (terminated in the past)
Generate (continuously)
Data points
X
$\{(\vec{x}, y)_i\}_{i=1}^{|\mathbf{X}|}$
Labels
Y
Data points
t
$\vec{x}_t \sim \mathcal{X}$
$y_{t+delay} \sim \mathcal{Y}$
delay
Labels
t
Provided (terminated in the past)
Provide (continuously)
Trusted Entities

Figure 3.5: The difference between static datasets (left) and data streams (right). The former can be understood as a fixed-size set of historical data and label pairs $(\vec{x_i}, y_i)$. The latter, in contrast, can be formally represented as a potentially infinite series of associated data points $\vec{x_t}$ and labels $y_{t+delay}$ drawn from random variables $\mathcal{X}$ and $\mathcal{Y}$, respectively. The $delay$ refers to the general case of labels not being immediately available.

3.2.2 Concepts and Concept Drift

While the notion of random variables is well suited for data stream formalization and considers the existence of noise, stream-generated data primarily follows patterns determined by reality. Such patterns need not be static.

Concepts

Referring to the machining example above, a certain machine operation pattern typically results in temperature values within a certain range. In machine learning, such a pattern is termed a concept (Webb et al. 2016). In technical terms, it can be defined as follows:

Definition 2 (Concept)**.** A concept is the joint probability distribution $P(\mathcal{X}, \mathcal{Y})$ of input data $\mathcal{X}$ and labels $\mathcal{Y}$. In unsupervised estimation problems, i.e. in the absence of labels, a concept can be represented as a probability distribution $P(\mathcal{X})$ (Webb et al. 2016).

During training, machine learning models attempt to algorithmically find formal representations for concepts to internalize them (Mitchell 1997, pp. 2–6). To augment the example above, a model might be trained to solve the task of identifying an abnormal machine health state, hence to find a mapping between $\vec{x_t}$ and y_t adhering to a real-world concept. Among other features, temperature might provide evidence for this task. Initially, it serves as training input and, during application, a model uses it as input $\vec{x_t}$ to infer the current health state y_t.

Concept Drift

As long as a concept remains unchanged, so too do the statistical properties of $P(\mathcal{X}, \mathcal{Y})$. Given the volatile nature of the processes generating data, underlying data distributions, however, need to be assumed to be non-stationary. Transferred to the machining example, many factors might invalidate previously identified temperature ranges indicating abnormal health states. Just a few of the large body of conceivable factors are the replacement of machine components, e.g. leading to better cooling capabilities, changes in the temperature environment conditions the machine is operated in or the deterioration of employed sensors. Therefore, a notion considering the volatility of concepts is required. This is fulfilled by the rationale of concept drift (Webb et al. 2016), which is defined as follows:

Definition 3 (Concept drift)**.** Concepts can be considered as dynamic with respect to a certain point in time. A concept drift is an event causing a certain concept $P_t(\mathcal{X}, \mathcal{Y})$, which is valid at time t, to become invalid and be replaced with a different concept $P_u(\mathcal{X}, \mathcal{Y})$, which is valid at time u. This can be formalized as $P_t(\mathcal{X}, \mathcal{Y}) \neq P_u(\mathcal{X}, \mathcal{Y})$, with $u > t$ (Webb et al. 2016). The event itself occurs or starts at a generally unknown point in time v, with $t < v < u$.

Different types of concept drift exist and manifest as different **dynamic profiles**, **magnitudes**, **subjects** and **feature-spatial expansions** along associated dimensions. Mixed types, i.e. a superposition of different concept drift types, can occur (Khamassi et al. 2018).

Regarding **dynamic profiles**, concept drifts can be described as either abrupt or extended. The latter can furthermore be characterized by specific transition types, such as gradual, incremental or probabilistic (Webb et al. 2016, Khamassi et al. 2018, Iwashita and Papa 2019). As pointed out by Definition 3, an abrupt concept drift can formally be assigned a point in time v characterizing its temporal context. For an extended one, however, a start time v can formally be defined. Several profiles are visualized in Figure 3.6.

Concept drift **magnitudes** can be quantified and thus discriminated with respect to their severity. This can e.g. be done using distribution dissimilarity metrics quantifying a notion of distance between two concepts. Magnitudes can also be discretized into different severity levels, such as warnings and changes (Webb et al. 2016), as visualized in Figure 3.9.

Regarding concept drift **subjects**, most relevantly, one needs to distinguish between real and virtual concept drift, which is visualized in Figure 3.7. If changes manifest in an evolving prior distribution $P_u(\mathcal{X})$, i.e. are observable exclusively within features of $\mathcal{X}$, virtual concept drift can be assumed. If exclusively the posterior distribution $P_u(\mathcal{Y}|\mathcal{X})$ is impacted, i.e. mappings of features onto labels, concept drifts are referred to as real. Mixed types also exist with respect to this concept drift dimension (Webb et al. 2016, Khamassi et al. 2018, Iwashita and Papa 2019). While real concept drift is prominently known to harm estimator performance (Webb et al. 2016), this may also hold for the virtual type (Sobolewski and Woźniak 2013, Woźniak et al. 2016a). Moreover, virtual concept drift can be an early sign of the real one, i.e. evolve into it (Khamassi et al. 2018).

Concept drifts can also have different **feature-spatial expansions**. They can be present in the entire feature space of the data, which is referred to as global concept drift. Otherwise, it might also be limited to certain features or subsets of such, which is referred to as local concept drift (Khamassi et al. 2018).

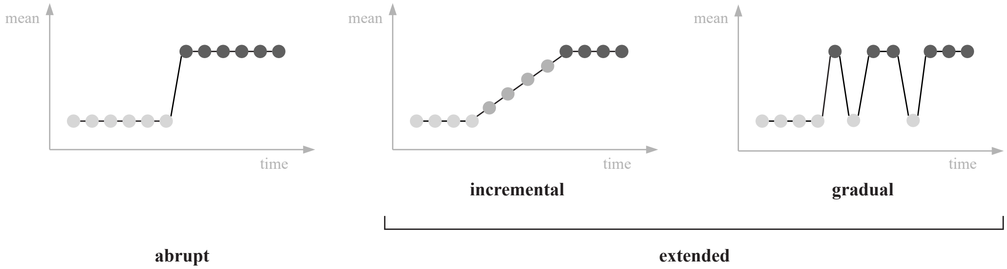


Figure 3.6: Prominent instances of dynamic concept drift profiles. An abrupt concept drift describes an instantaneous change from one concept to another. In contrast, extended concept drifts describe changes featuring transitions between concepts. The latter can, among others, be further classified into incremental and gradual manifestations. To schematically illustrate each of these profiles, a concept-drift-impacted mean value of an arbitrary variable is plotted against a temporal axis. Figure based on Lemaire et al. (2015).

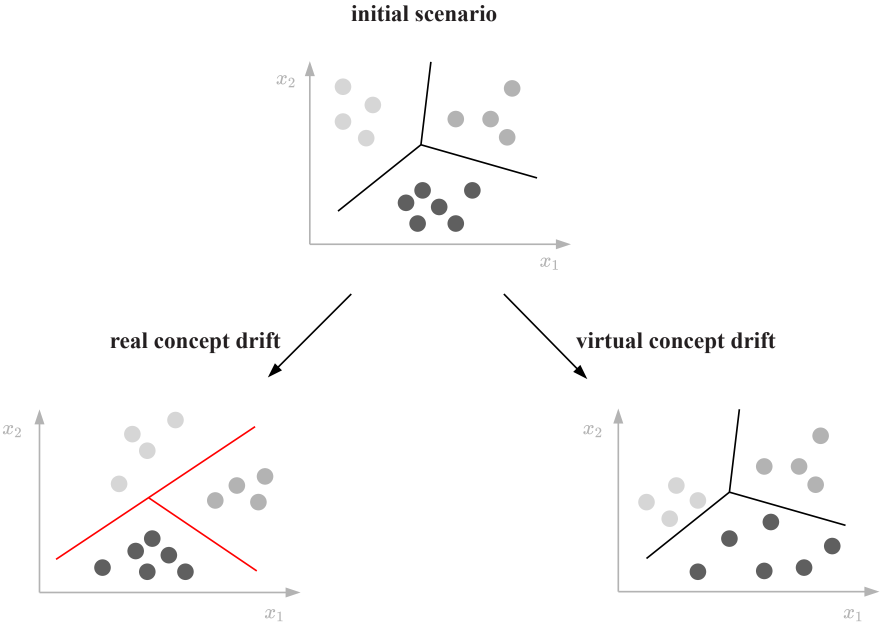


Figure 3.7: Concept drift subjects illustrated with a fictional two-dimensional data classification scenario, where data dimensions $x_i \in \mathbb{R}$ with $i \in \{1, 2\}$. In the case of real concept drift, changes invalidate a previously learned model and demand the formulation of a new one, which is depicted as a set of black and red lines, respectively. In contrast, virtual concept drift causes data distribution changes that do not invalidate the employed model. Figure based on Khamassi et al. (2018).

3.2.3 Concept Drift Adaptation

In the industrial practice, the problem of concept drift is often ignored and lacks concrete established best practices (Vela et al. 2022). Given the destructive potential emanating from both real and virtual concept drift, as well as the democratization of machine-learning-based solutions, this is surprising. Concept drift adaptation can be employed as a solution to this problem.

Basic Principles and Scientific Progress

Concept drift adaptation can be regarded as a set of strategies aiming at sustaining the robustness of machine learning estimators by administering appropriate responses treating concept drift occurrences (Gama et al. 2014, Khamassi et al. 2018). The scientific field of concept drift adaptation therefore delves into the intricacies of designing associated algorithms. Such typically involve estimator retraining (Gama et al. 2014), employing approaches appropriate for the respective learning algorithm type (cf. Section 3.1.1), or ensemble-based constructs (Krawczyk et al. 2017).

The research on concept drift adaptation progresses rather slowly (Vela et al. 2022) and furthering its applicability for a wider range of problems is a matter of ongoing work. For instance, as demonstrated by the author of this thesis, a novel approach based on concept drift adaptation can also be applied to guide and improve the efficacy of the fine-tuning process of deep-learning-based transformer architectures (Sturm et al. 2024). In essence, this approach closely observes the loss resulting during individual training iterations. If an increasing loss is identified via concept drift detection, one can trigger responses intervening in the training process. Specifically, such might involve e.g. repeating certain iterations, changing parameters or other interventions.

Implicit versus Explicit Approaches

Concept drift adaptation can be done in an implicit or explicit fashion (Iwashita and Papa 2019). The former, which is also referred to as blind adaptation, attempts to treat concept drift occurrences without any information about their timing or whether such have occurred at all. Approaches of this type usually involve estimator retraining in regular intervals or when labels become available (Gama et al. 2014, Krawczyk et al. 2017). In contrast, the latter, which is the focus of this thesis, employs informed measures and is visually represented in Figure 3.8. Any triggered actions are based on prior explicit identification of concept drifts occurrences and their timing by exploiting the output of concept drift detectors (Gama et al. 2014). Unlike implicit adaptation, with such controlled responses, the explicit approach has the potential to improve an economized utilization of compute resources and labels (Sethi and Kantardzic 2015) as well as allows the deliberate treatment of concept drift occurrences (Minku and Yao 2012). Furthermore, explicit concept drift adaptation can potentially utilize more in-depth information on concept drift characteristics to realize guided adaptation measures (Gama et al. 2014).

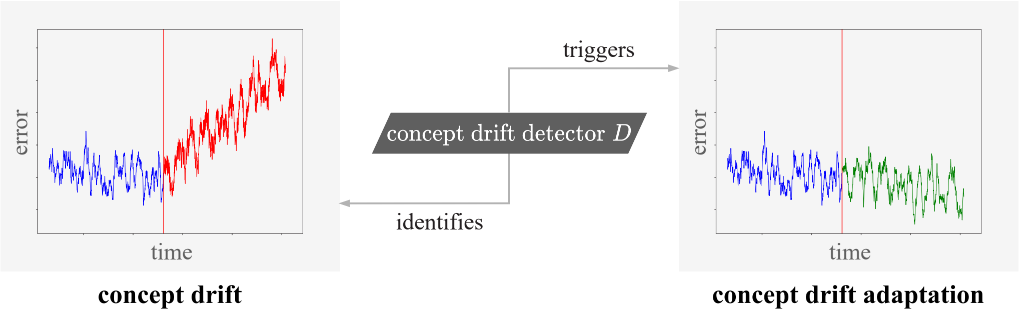


Figure 3.8: An abstract schematic visualization of the detection and adaptation of concept drift. The point in time a concept drift occurs at is represented as red vertical line inside plots showing the error (blue) of a machine learning regressor over time. Untreated, the error might increase, which is represented by red error curve. A concept drift detector D can identify that point in time by receiving the error as input. This information can subsequently be used to treat the concept drift by employing a concept drift adaptation strategy. This has the chance to avoid an increasing error, represented by a green error curve.

3.2.4 Concept Drift Detection

Explicit adaptation requires concept drifts to be identified as outlined above and visualized in Figure 3.8. Concept drift detection describes the set of structured approaches to achieve this goal.

Basic Principles and Evaluation

Many concept drift detection methods are based on a detector construct D featuring a specific detection algorithm f^{det} that continuously monitors similarities or evolving patterns of data characteristics or attributes (Agrahari and Singh 2022, Khamassi et al. 2018, Gama et al. 2014). Within the context of this thesis, the input for D is consistently referred to as concept drift evidence and its output as alert. Ground truth knowledge of the locations of concept drifts in the data, and optionally of their types, is referred to as concept drift labels.

It is critical to distinguish concept drift from anomalies or outliers. While only the former should be treated, i.e. via adaptation, the latter should not (Gama et al. 2014). Apart from being processed for estimator maintenance, detected concept drifts can also be employed for various other purposes, such as concept drift understanding (Mahdi et al. 2024, Agrahari and Singh 2022, Lu et al. 2018).

Furthermore, a set of metrics can be employed to evaluate the performance of D instances. These can support their parametrization by aiming at finding solutions within the trade-off between high detection quality and missed detections. Typically, these metrics quantify notions of the frequency of false or absent alerts, e.g. the missed detection rate, as well as of the temporal delay between a concept drift occurrence and its detection, e.g. the mean time to detection (Iwashita and Papa 2019, Bifet 2017).

Supervised versus Unsupervised Approaches

Regarding the utilization of estimation labels, a D can be supervised or unsupervised. In the former case, a D is employed to probe measurements of an estimator's performance for concept drift occurrences (Agrahari and Singh 2022, Khamassi et al. 2018, Gama et al. 2014). Figure 3.9 schematically illustrates the deterioration of an estimator's performance caused by concept drift. Certain points, i.e. a warning and a change-level magnitude of deterioration, can be detected by a well parametrized D. In the latter case, the stream of incoming data or model characteristics are directly probed for concept drift occurrences. In contrast to supervised approaches, unsupervised ones do not utilize estimator performance data (Gemaque et al. 2020).

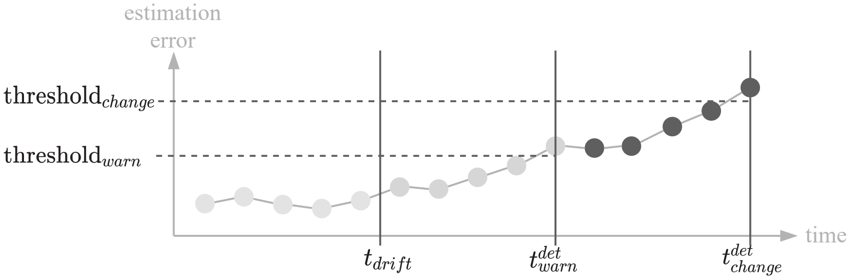


Figure 3.9: Supervised concept drift detection. A schematic error signal computed from an estimator's outputs exhibits evidence of concept drift. Time t_{drift} marks the actual time of the concept drift occurrence. A detector identifies it with delay and emits a warning and a change alert at times t^{det}_{warn} and t^{det}_{change}, respectively, once the associated error thresholds are exceeded. Figure based on Krawczyk et al. (2017).

Prominent Algorithms

Popularly employed D representatives are Adaptive Windowing (ADWIN), the Page-Hinkley test (PH) and the Drift Detection Method (DDM). These are continuously being employed in practice or often used as basis for specialized approaches.

Adaptive Windowing. This approach provides rigorous performance guarantees for detecting various kinds of concept drifts in supervised and unsupervised settings. It maintains two sliding windows of data or performance values and, if they exhibit great-enough statistical differences, emits an alert. The detector's sensitivity can be parametrized by the user (Bifet and Gavaldà 2007).

Page-Hinkley test. This statistical-test-based method operates by monitoring a sequence of data or performance value observations and calculating a cumulative sum based on these. Once it exceeds a user-parametrized threshold, it may indicate the occurrence of concept drift in the data stream (Page 1954).

Drift Detection Method. Based on the probably approximately correct learning model, which refers to an estimators capacity to represent a hypothesis with minimal error (Mitchell 1997, pp. 205–207), DDM assumes a classifier's stable or increasing performance. It monitors and statistically models the correctness of observed classifications and emits an alert upon sufficiently many false ones. The user can parametrize statistical measures of dispersion to control the method's sensitivity (Gama et al. 2004).

3.3 Data Science Process Models

Data science is the discipline that considers the application of machine learning estimators from the respective problem formulation motivated e.g. by its business context, through their development, to their productive utilization. Data science process models further define process-centric iterative frameworks that ensure consistency, repeatability and sustainable collaboration in data science projects. This way, they render such projects manageable by formalizing workflows and offering best practices (Provost and Fawcett 2013, pp. 19, 27–34).

Cross-Industry Standard Process for Data Mining. The possibly most popular data science process model is the Cross-Industry Standard Process for Data Mining (CRISP-DM). It was introduced in the year 2000 and aims at being applicable independent of the type of the target industry as well as of the specific technology and software being utilized. As shown in Figure 3.10, it features six steps that are formally traversed linearly and in an iterative fashion. Nevertheless, in practice, traversing them in a different order is possible. Each of these steps represents the highest level of the hierarchically defined underlying process model and can be broken down into generic tasks. Each task again contains various specific tasks itself (Wirth and Hipp 2000, Chapman et al. 2000).

Process Model for Artificial-Intelligence-Systems Engineering. Later, other data science process models building on CRISP-DM are introduced. For instance, developed in 2021, the Process Model for Artificial-Intelligence-Systems Engineering (PAISE) aims at developing entire machine-learning-based systems instead of mere models. It is displayed in Figure 1.1. It bases

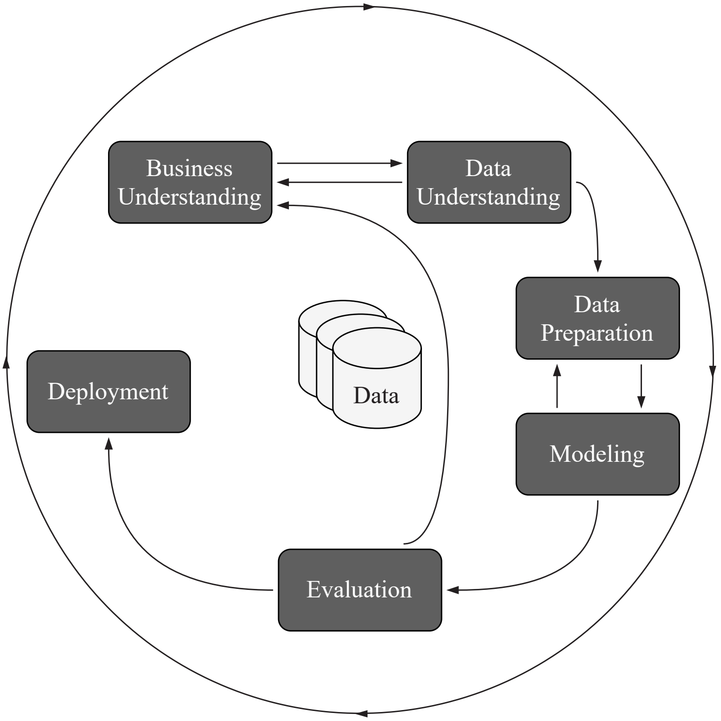


Figure 3.10: The Cross-Industry Standard Process for Data Mining (CRISP-DM). By providing a high-level iterative phase-based framework, it aims at making the complex nature of data science projects more manageable. Figure based on Wirth and Hipp (2000).

on knowledge from traditional systems engineering and includes best-practices from software engineering and data-driven development to meet the specific requirements of such systems. Similar to CRISP-DM, PAISE also enables iterative development approaches (Hasterok et al. 2021).

Cross-Industry Standard Process Model for the Development of Machine Learning Applications with Quality Assurance Methodology. Proposed around the same time as PAISE, the Cross-Industry Standard Process Model for the Development of Machine Learning Applications with Quality Assurance Methodology (CRISP-ML(Q)) also more heavily considers the requirements specific to machine-learning-based systems. Notably, it emphasizes the importance of the operation phase, particularly to address potential performance degradation of estimators over extended periods. This is achieved through the integration of quality assurance methodologies, which represent a significant enhancement over CRISP-DM. Additionally, CRISP-ML(Q) incorporates tasks relevant to machine learning estimator development, reflecting advancements from more recent literature (Studer et al. 2021). Figure 3.11 shows the model with its major phases designated at the top.

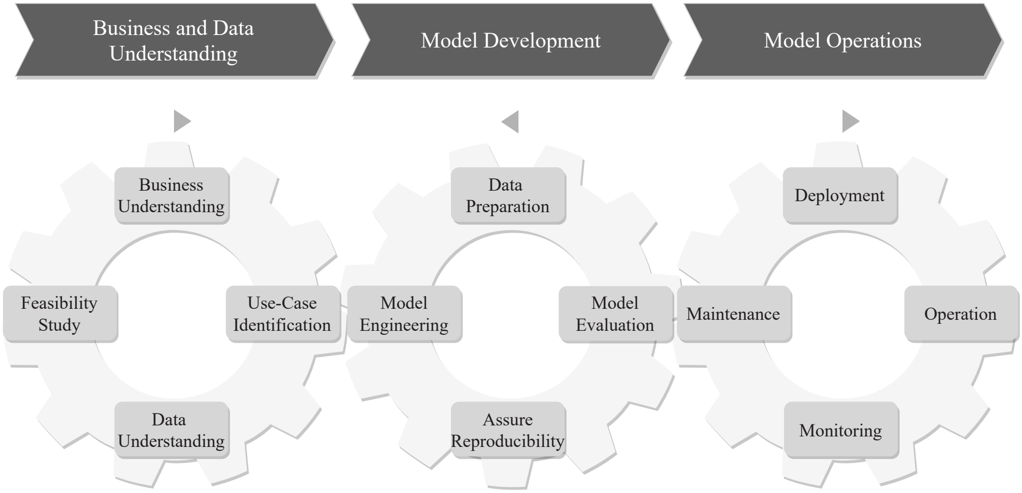


Figure 3.11: The Cross-Industry Standard Process Model for the Development of Machine Learning Applications with Quality Assurance Methodology (CRISP-ML(Q)). While the three major phases are denoted at the top, detailed tasks are indicated on interlocked cog wheels symbolically representing these phases. They express the notion of the phases being highly dependent on each other. Triangles mark the cog wheel's rotation direction and start of each major phase. Figure based on Studer et al. (2021).

Other instances. Over the years, established technology companies have also introduced data science process models to guide machine learning estimator development. With the Team Data Science Process, Microsoft promotes an openly developed and community-driven framework aimed at supporting the development of machine learning solutions (Tabladillo 2024). While it minimizes explicit references to proprietary products, other companies have released data science process models more closely tied to their respective offering. For instance, IBM's Analytics Solutions Unified Method for Data Mining (IBM 2016) and the Well-Architected Machine Learning Lifecycle introduced by the company Amazon Web Services (AWS 2023a,b) incorporate features of their cloud infrastructure and service portfolios. These models provide templates and workflows that address e.g. performance optimization and cost management during solution development. Consequently, by focusing on respective product integrations, their degree of technology independence is reduced.

3.4 Subject Orientation

Subject Orientation is a method to descriptively model process systems and can be seen as a part of the wider discipline of Subject-Oriented Business Process Management. It differentiates between active entities and passive elements and refers to them as subjects and objects, respectively. All activities that are part of a process need to be placed in the direct context of subjects. The interaction between subjects thus becomes the center of a model's consideration and explicitly represents their exchange of information as a form of communication. This way, it stands out when compared to predominantly procedural and activity-focused modeling techniques, such as the Business Process Model and Notation (Elstermann 2020, pp. 66–67, 79–82), (Fleischmann et al. 2012, pp. 63–66).

Core Principles and Components

The Parallel Activity Specification Schema is a graphical modeling language that enables the design of descriptive models strictly adhering to Subject Orientation principles. Applying it requires modeling two interrelated diagrams: the subject interaction diagram (SID) and the subject behavior diagram (SBD). Associated examples are presented in sub-figures (a) and (b) of Figure 3.12, respectively. In the SID, all active entities, referred to as subjects, that are relevant for a particular process system, as well as their interaction are modeled. Interaction can e.g. involve the exchange of data objects, which are represented as yellow envelopes in sub-figure (a) of Figure 3.12. While interface-type subjects are only represented in SIDs, more detailed information on regular subjects, such as workflows, are contained in associated SBDs. Each SBD describes the different feasible states a subject can be in as well as the transitions between these (Elstermann 2020, pp. 83–89), (Fleischmann et al. 2012, pp. 289–230). They are represented as rectangles and arrows, respectively, in sub-figure (b) of Figure 3.12. Three different types of states are distinguished (Elstermann 2020, pp. 86–88):

- Do-states (yellow) describe the actions a subject performs.
- Send-states (green) describe the sending of data objects to other subjects.
- Receive-states (red) describe the waiting of a subject for receipt of data objects from other subjects.

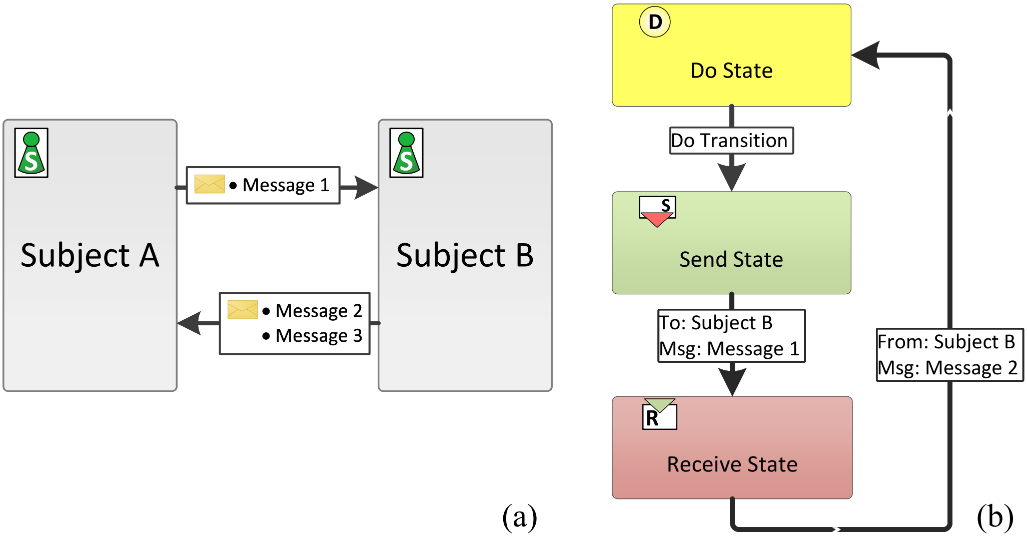


Figure 3.12: Purely exemplary model created using the method Subject Orientation. The subject interaction diagram (SID) (a) shows all relevant subjects and their interaction via messages. The subject behavior diagram (SBD) (b) further defines the subjects and their workflows, using do, send and receive states as well as respective transitions.

Rigorous Application and Validation

When applying Subject Orientation, a user is forced to follow syntactical rules. The correctness of a resulting model can be evaluated using a simple simulation algorithm. It involves the construction of a chance and duration tree for each SBD. This way, the traversing through sequences of states can be modeled probabilistically, with either uniform or user-defined probability distributions assigned to all state transitions. Also, suitable boundary conditions, such as a maximum recursion depth for cycles, are defined. Then, the total process duration is iteratively computed by summing up transition, execution and waiting times across differently probable paths through all SBDs. Simulation errors, such as unreferenced elements of the SID, lead to immediate failure of the validation algorithm, which ultimately aims at supporting sound modeling (Elstermann and Ovtcharova 2018).

It can be argued that Subject Orientation is well suited to enclose even high degrees of process complexity within descriptive models. It enables observers to grasp intricacies in an intuitively understandable way for a wide range of application fields (Elstermann 2020, p. 278), (Elstermann et al. 2021, Bönsch et al. 2022).

3.5 Summary

The content of this chapter provides the reader with the relevant information on the methods researched and applied in this thesis. This involves a discussion of the ongoing transition from a static and often batched type of machine learning to a more dynamic one characterized by evolving and potentially infinite data streams. In highlighting this transition, this chapter also addresses methodological tensions between established machine learning paradigms and the demands of today's real-world environments. Given the decades of research that went into batch-based learning, this thesis puts emphasis on supporting its continued use in data stream scenarios despite the destructive impact emanating from the phenomenon concept drift, which is introduced in this chapter. The discussion of this phenomenon and associated adaptation mechanisms establishes a theoretical foundation later chapters base assessments of machine learning estimator robustness in production scenarios on.

In addition, the abstract concept of forming ensembles is introduced. It is employed at multiple points in the thesis to improve the performance of machine learning estimators as well as of approaches to detect concept drift by incorporating diversity. By outlining the algorithmic principles ensemble methods build on, this chapter creates the basis for understanding how complementary model behaviors can be combined to mitigate uncertainty and enhance their performance.

Other aspects that lay the groundwork for the practical orientation of the research results documented in the following are additionally outlined. Firstly, commonly applied models in data science practice are characterized and discussed, offering a structured perspective on how machine-learning-based solutions are developed from an initial problem formulation to their productive deployment. Secondly, the modeling method Subject Orientation, which is selected as description approach for various model-based artifacts proposed later on in this thesis, is presented. Its conceptual rigor and emphasis on considering the human as integral element within technical systems provide a descriptive vocabulary that supports the formalization and discussion of these artifacts.

4 State of the Art

This chapter extends the background knowledge provided in the preceding one by discussing those scientific works that define the state of the art relevant for this thesis. These tend to various aspects belonging to the fields of concept drift detection and adaptation, as detailed in Sections 4.1 and 4.2, respectively. Section 4.3 then precisely identifies the research gaps that persist.

4.1 Concept Drift Detection Methods

The body of research on concept drift detection, i.e. the identification of concept drift occurrences as basis for e.g. explicit adaptation-related responses, primarily focuses on applying a set of well-understood detectors in rather ordinary settings such as classification problems. A series of highly impactful surveys, such as those of Gemaque et al. (2020), Iwashita and Papa (2019) and Khamassi et al. (2018), provide a profound taxonomic understanding and characterization of such methods as well as introduce their first applications. In contrast, among other aspects, rather sparsely explored are concept drift detection methods for regression problems and the formation of detector ensembles. These sub-fields are introduced in the following.

4.1.1 Concept Drift Detection for Regression Problems

The application of machine learning for regression problems is an under-researched field (Lima et al. 2022, Read and Žliobaitė 2023). This gap is even more severe regarding concept drift detection on that field. Only few scientific works explore the design and application of detectors and are restricted to merely a small number of benchmark datasets (Cavalcante et al. 2016, Bayram et al. 2022). These and further regression-specific aspects, such as concept drift adaptation, evaluation and prevailing challenges, are subject of a systematic literature review conducted by Lima et al. (2022). In the following, unsupervised as well as supervised approaches are overviewed.

Unsupervised Approaches

Unsupervised concept drift detection approaches do not require access to ground truth information. Ikonomovska et al. (2009) for instance employ the Page-Hinkley test (PH) (cf. Section 3.2.4) in an approach specialized on detecting concept drifts when employing online regression tree estimators. They suggest a loss metric computed by continuously comparing estimations at a tree's leafs or parent nodes of such. Consequently, local and global concept drifts can be detected this way. Despite the term *loss* typically referring to a machine learning estimator performance measure computed in a supervised fashion, the formalization of their approach reflects no indication of any ground truth use. They report good performance with respect to detection of abrupt and gradual concept drifts.

Other works evaluate the use of estimator-agnostic unsupervised concept drift detectors for regression problems. For instance, the statistical hypothesis detector exponentially weighted moving average for concept drift detection (Ross et al. 2012) can be employed on regression data streams. It takes as input a distance measure computed from eight statistical features describing correlation, kurtosis and variance-related input data properties (Cavalcante et al. 2016). Similarly, detectors monitoring distribution dissimilarity measures of the input data stream can be utilized (Song et al. 2023, da Costa et al. 2016).

Supervised Approaches

Several works design specialized concept drift detectors that implement supervised approaches (Lima et al. 2022), i.e. such that require ground truth information. For instance, Mayaki and Riveill (2022) suggest exploiting properties of self-exciting auto-regressive models, i.e. non-linear time series forecasting methods, for concept drift detection purposes. To compute the detector inputs, regression error metrics that are not further specified are employed. Existing concept drift detectors can also be used, with the Drift Detection Method (DDM) (cf. Section 3.2.4) being most referred to in scientific works (Lima et al. 2022).

Pronouncedly few works directly utilize regression error scores as input for concept drift detectors, which is visually represented in Figure 4.1. In another work, Lima et al. (2021) conduct a comparative analysis of a set of detectors that use data properties as well as such scores in a supervised fashion. They utilize a small number of artificial-drift-induced as well as real-world benchmark datasets and analyze whether all concept drifts can be identified and the associated delay. Apart from that, they also analyze machine learning estimator scores as indirect indicators of the impact of explicit concept drift adaptation on estimator performance.

Other works design and employ rather simple concept drift detectors. As an example, one can

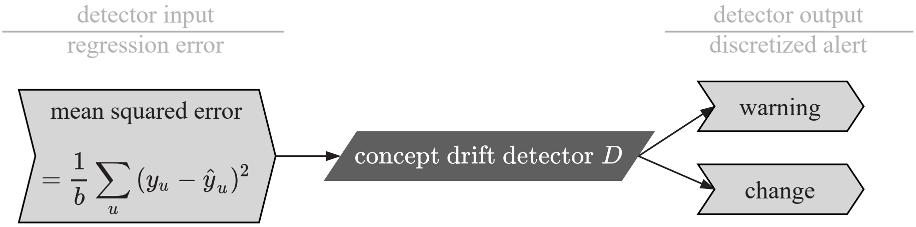


Figure 4.1: Utilizing machine learning regression errors as concept drift detector input. This can, for instance, be the mean squared error, which is a metric computed over the predictions $\hat{y}_u$ for a data batch with size b and associated labels y_u (Hastie et al. 2009, p. 24). A detector identifying abnormal patterns in these errors emits alerts e.g. on warning or change level.

compare the root mean squared error of a set of models, with only one of them being actively used for estimations. Once one of the background models exhibits an error that falls below the one of the actively used model, it replaces the latter. In doing so, this incident that can be referred to as an error intersection is exploited as concept drift detection alert (Baier et al. 2020). Straightforwardly, concept drift alerts can also be emitted based on a regression error metric exceeding a user-specified threshold (Liu et al. 2019, Cavalcante and Oliveira 2015). E.g. using the mean squared error, these alerts can be used in machine learning ensembles to adapt weights or to replace members (Soares and Araújo 2015).

4.1.2 Concept Drift Detection Ensembles

The last decade brought forth the first research on CDDEs. Despite the straightforward motivation of leveraging ensemble diversity, i.e. combining the individual strengths of multiple different concept drift detectors, the number of works is limited. In the following, these are outlined. It shall, however, be remarked that further details on these works are provided in a separate publication (Trat and Ovtcharova 2023). With the author of this thesis being the originator of these insights, there may be verbatim equivalents of certain passages. Additionally, Chapter 6 presents a systematic literature review as well as a deep analysis on CDDEs.

Member Algorithm Variety

A considerable set of works conceptualize CDDEs with the goal of leveraging ensemble diversity to increase concept drift detection performance. The diversity potential of a CDDE is strongly influenced by the variety of the concept drift detector algorithms it combines: Several works (Sobolewski and Woźniak 2013, Maciel et al. 2015, Woźniak et al. 2016a, Zhang et al. 2020, Perez et al. 2020, Xu and Klabjan 2021) suggest heterogeneous CDDEs, which comprise multiple different detection algorithms. In contrast, homogeneous ones comprise multiple differently parametrized instances of one detection algorithm. This type of CDDE is also subject of several works (Woźniak et al. 2016b, Bu et al. 2016, Pesaranghader et al. 2018, Korycki and Krawczyk 2019, Okawa and Kobayashi 2021, Nguyen et al. 2022, Komorniczak et al. 2022). As pointed out later in Chapter 6, this CDDE design aspect is of high significance with respect to the associated field's progression as well as various other design aspects.

Concept Drift Type Coverage

A few works build on the existing research and specifically explore the benefits of ensemble diversity for increasing a CDDE's coverage of concept drift types (cf. Section 3.2.2), i.e. the capability to detect different types. This aspect is schematically visualized in Figure 4.2. To achieve this, a substantial amount of consideration is invested into base concept drift detector selection. Du et al. (2015) propose a two-step heuristic for selecting base concept drift detectors from a set of various supervised ones to form a CDDE. Firstly, they group multiple detectors based on the target concept drift type they are respectively specialized on. Secondly, only the best representative of each group is selected based on benchmarking all detectors on a dataset with known concept drift evidence. Hu et al. (2018) select base detectors based on prior knowledge of their respective concept drift type specialization. Additionally, detections are gathered and processed for each target concept drift type separately. In a later work, Hu and Kantardzic (2022) repeat the experimental evaluation of their CDDE while including another different concept drift detector to further increase the coverage of drift types. The three aforementioned works choose heterogeneous CDDE architectures. In a further thesis, Hu (2022) employs heterogeneous CDDEs on various popularly employed benchmark classification problems and synthetic datasets. Also, methods to handle class imbalance as well as visualization approaches facilitating explainability in streaming data concepts are proposed in this work.

Concept Drift Adaptation Potential

It is also possible to leverage CDDE ensemble diversity to improve concept drift adaptation performance. Despite the significant importance of this goal for practical applications relying on productively employed machine learning models, only two works focus on it. Lapinski et al. (2018) employ a homogeneous CDDE architecture. They consider the approach to have different base detectors separately focusing on local concept drifts by associating it to separate data stream features. Toumi et al. (2020), in contrast, employ a heterogeneous architecture suggested by the aforementioned work of Maciel et al. (2015). The work of Toumi et al. (2020) is the only one among CDDE works evaluated on a real-world case study. In this case study, they attempt to

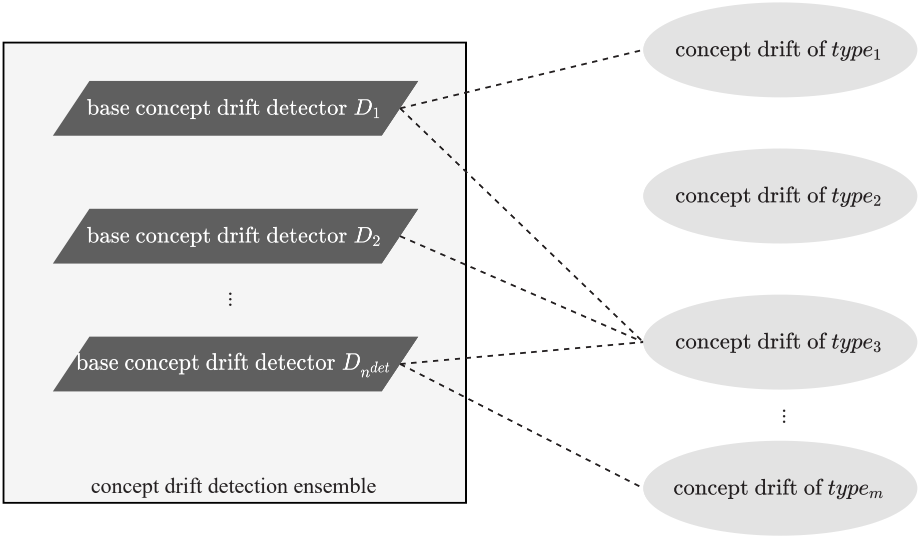

base concept drift detector D_1
base concept drift detector D_2
base concept drift detector $D_{n_{det}}$
concept drift detection ensemble
concept drift of $type_1$
concept drift of $type_2$
concept drift of $type_3$
concept drift of $type_m$

Figure 4.2: A concept drift detection ensemble's capability to cover various concept drift types. This goal can be reflected in the ensemble design by specifically considering the member-individual specializations to detect certain concept drift types. These specializations are represented by dashed lines. While, in this schematic example, $type_2$ is not covered by any detector's specialization, the remaining types are. Figure based on Hu et al. (2018).

estimate discrete levels of virtual machine load in a cloud computing scenario and intend to trigger load balancing actions, while keeping memory usage low at all times.

Performance Potential

The verdict of all aforementioned studies regarding the performance of the individually suggested CDDEs is overwhelmingly positive. All succeed in leveraging ensemble diversity for their individual goals or at least propose specific incitations to compensate respective use-case-specific drawbacks as future work.

4.2 Explicit Concept Drift Adaptation Methods

Overall, concept drift adaptation is a highly relevant subject of active research. While implicit approaches are reasonably well described in science and practice, explicit approaches are lacking in several respects. The research field of machine-learning-estimator-ensemble-based methods for the latter approach, for instance, advances rather slowly. Also, studies on deployable machine learning system solutions containing components implementing such approaches are rare. Merely a range of specific and central aspects that need to be considered in such solutions can be identified to a certain extent. The following provides an overview of these topics, extracted from scientific literature.

4.2.1 Continuous Ensembles

A promising approach to conduct concept drift adaptation is to leverage capabilities of machine learning estimator ensembles, which are introduced in Section 3.1.2. When augmented by algorithmics for continuous learning in data streams, they are referred to as continuous ensembles[4] (Krawczyk et al. 2017). The majority of existing scientific works suggest implicit concept drift adaptation approaches, i.e. such not utilizing concept drift detectors alongside machine learning estimator ensembles (Krawczyk et al. 2017). This thesis, however, has a pronounced focus on explicit approaches and provides an overview over such in the following paragraphs. To allow for a high degree of member customization, an emphasis is placed on estimator-model-agnostic ensembles. For the sake of completeness, a comprehensive overview of the state of science of both approaches is additionally provided in Trat et al. (2024). Certain following passages may be verbatim equivalents as the author of this thesis is the originator of this information.

Classification Scenarios

Several ensembles are suggested for the explicit concept drift adaptation in classification scenarios. One might, for instance, consider approaches that leverage the capability of ensembles to introduce diversity. In the work of Minku and Yao (2012), different diversity magnitudes are exploited via separate sub-ensembles. More specifically, on the one hand, one might fit classifiers in a low-diversity manner and add them as members to a certain sub-ensemble to represent new concepts. Long-lasting concepts, on the other hand, can be exploited by maintaining sub-ensembles containing members trained in a high-diversity manner in the past. In this regard, diversity can be quantified and optimized using a measure of agreement between classifiers for a set of data examples. For low-magnitude incremental concept drifts, this approach is observed to work well (Minku and Yao 2012).

Other ensemble approaches utilize concept drift detections as trigger to add new members. Such might only initially be trained on the concept-drift-triggering batch while their weights are

[4] Despite being commonly referred to as *continuous* ensembles, such constructs are note limited to purely continuous-training-type algorithms but can also employ e.g. emulated-batch-continuous ones.

constantly updated based on measurements of their performance. Members might be discarded once their weight reaches zero (Deckert 2011). Alternatively, new members can be trained on larger data batches and be kept indefinitely causing the ensemble to grow constantly (cf. Figure 4.3) (Nishida et al. 2005). These approaches exhibit good performance for abrupt or gradual drifts and recurring concepts, respectively, and highlight the ensemble capability of leveraging weighing dynamics.

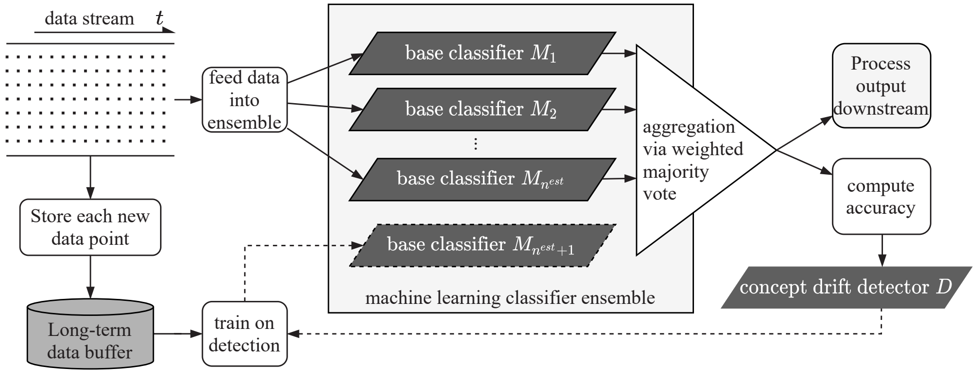

data stream t
feed data into ensemble
base classifier M_1
base classifier M_2
⋮
base classifier $M_{n^{est}}$
base classifier $M_{n^{est}+1}$
aggregation via weighted majority vote
machine learning classifier ensemble
Process output downstream
compute accuracy
concept drift detector D
Store each new data point
Long-term data buffer
train on detection

Figure 4.3: A continuous-ensemble-based concept drift adaptation approach for classification scenarios from the state of science by Nishida et al. (2005). A buffer saves data and provides it on concept drift detections to train additional members. Inference is conducted based on a weighted majority vote. Figure based on Nishida et al. (2005).

Regression Scenarios

The research gap of ensembles approaches for explicit concept drift adaptation emerges as even more severe if one exclusively considers works focusing on regression problems. Even across the wider discipline of machine learning, its application for regression problems is still a heavily under-researched aspect (Lima et al. 2022, Read and Žliobaitė 2023). Among the most popularly employed methods are the streaming random patches regressor (SRPR) and the adaptive random forest regressor (ARFR). Both are limited to the use of continuous-training-type machine learning estimator models. To increase diversity, the SRPR maintains several feature subsets and trains members on these. The subsets are formed by drawing from the full feature set at random. After ensemble deployment, each of these members is continuously monitored by one concept drift detector. If one detector emits a warning, a new background estimator is created and trained continuously. However, at that point, this estimator's output is not considered for computing the ensemble-global prediction. On a change detection, the member is replaced by its associated background member. The member weights are computed based on measurements of their performance (Gomes et al. 2019, 2020).

The ARFR is mostly identical to SRPR except the following: It exclusively employs members with tree-based continuous-training-type regressor models, i.e. is not model-agnostic. It also maintains randomly assembled feature subsets but these have influence only on node splits within trees instead of the training of entire members (Gomes et al. 2018).

4.2.2 Application in Production

The highly influential work on the field of concept drift adaptation of Gama et al. (2014) implies a wide range of concept drift adaptation strategies. However, it describes them from a predominantly functional perspective without providing actionable considerations regarding their implementation in machine learning system solutions. The following paragraphs provide an overview of initial advances towards the productive application of explicit concept drift adaptation. This is done by referring to exemplary ones proposing associated approaches as components of such system solutions. Figure 4.4 additionally provides an overview of technical aspects covered by them.

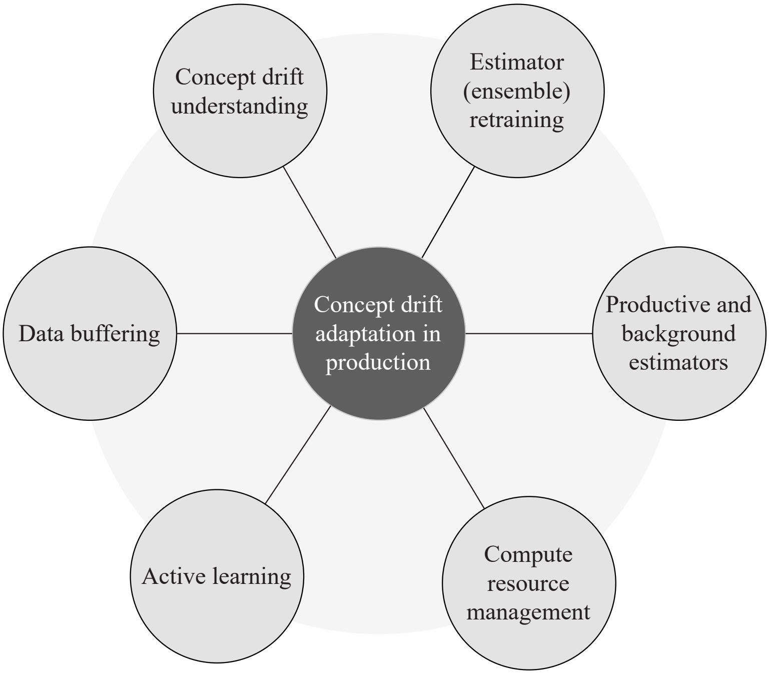

Concept drift understanding
Estimator (ensemble) retraining
Data buffering
Concept drift adaptation in production
Productive and background estimators
Active learning
Compute resource management

Figure 4.4: Technical aspects of concept drift adaptation in production scenarios covered by the recent state of science of this field.

Training, Inference and Retraining

Trivially, a central aspect of above-referenced systems is their application of machine-learning-based estimators to process data input and to provide associated outputs. The former might stem from e.g. regression or classification problems the estimators are designed for. Apart from the frequent use of base estimators, machine learning estimator ensembles are also employed (Casado et al. 2023, Li and Zhao 2024, Wilson et al. 2023). Before commencing the processing of a data stream, estimator models are trained on historical data (Angbera and Chan 2024), or training is done on the first batches of the stream. From then, sustaining a model's robustness is crucial. On high-magnitude concept drift detections, i.e. such indicating grave changes in the data, a productively employed estimator can be replaced directly. In case of less severe detections, which may be referred to as warning-level detections, one or more separate background estimators can be instantiated. Before replacing the productively employed estimator due to a specific condition being fulfilled, the background ones can continuously be trained (Jin and Zhang 2023, Wilson et al. 2023).

Concept Drift Understanding

Activities of concept drift understanding, i.e. further analysis of concept drift occurrences to leverage associated evidence can be utilized by downstream applications or to improve concept drift adaptation strategies. Such include, for instance, information on the timing, location, or reasoning of concept drifts (Mahdi et al. 2024, Agrahari and Singh 2022, Lu et al. 2018). Utilizing this type of insight can, furthermore, increase the compute resource efficiency by identifying and only updating parts of estimator models that are affected by a concept drift (Dong et al. 2022).

Data Buffering

The buffering of input data or labels is rarely considered or even frowned upon (Read and Žliobaitė 2023). With regard to input data, only few approaches, e.g. those of Hu (2022) and Shayesteh et al. (2022), suggest concept drift adaptation approaches requiring a buffer. In their cases, a buffer is employed to gather labels within the context of an active learning strategy and to collect sufficient data for a reinforcement-learning-based estimation approach, respectively. With regard to label buffering, a possible reason for this being done quite rarely might be the popular but often unrealistic assumption of immediate label availability (Fahy et al. 2023).

4.3 Research Gaps

The preceding sections highlight a range of promising approaches and ongoing developments in the fields of concept drift detection and adaptation. At the same time, several limitations in scope, methodological coverage, and applicability remain. The following paragraphs summarize the most relevant research gaps, which are additionally visualized in Figure 4.5, motivating the practically relevant and novel contributions of this thesis.

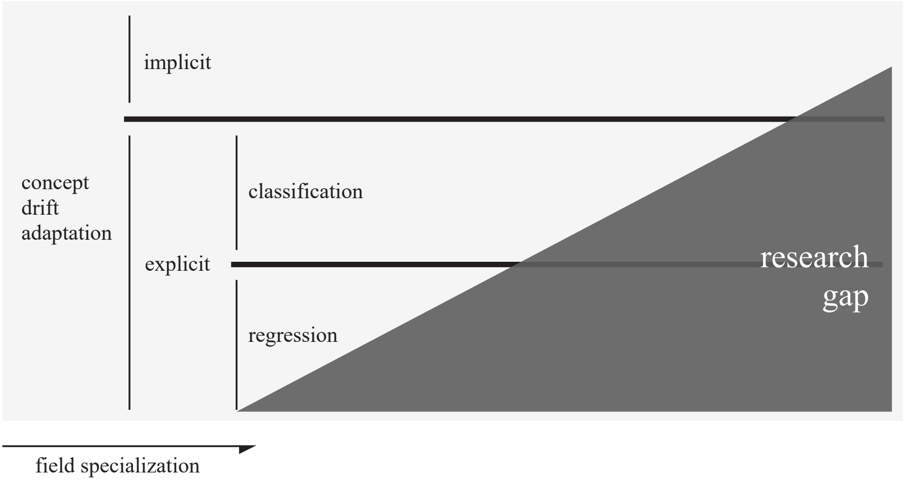


Figure 4.5: The research gap present in the scientific field of concept drift adaptation. This field is increasingly concretized into its mutually exclusive sub-fields from left to right. The area covered by the triangular shape represents the research gap with respect to these sub-fields. The more area is covered, the more the respective field is under-researched. This illustrates that explicit concept drift adaptation approaches for regression scenarios are most critically under-researched.

- The scientific field of **concept drift detection for regression problems** is still in its early stages, whereas classification has already been extensively examined. Existing methods for the former are heavily restricted to a limited selection of benchmark datasets and often rely on simple extensions of supervised detection methods. Regression-specific detector designs, evaluation approaches and scalability aspects thus remain open to systematic exploration.

- Despite the demonstrated performance potential of machine-learning-ensemble-based approaches, research on **concept drift detection ensembles (CDDEs)** is severely limited. Initial works evaluate the diversity-induced potential of such constructs but focus almost exclusively on their detection performance while largely neglecting their adaptation-related one. The gap is particularly significant regarding the application of CDDEs for regression problems, with no existing work tending to this aspect, and on real-world data in general. Also, their design space remains under-explored.

- In the domain of explicit concept drift adaptation, **continuous ensembles** offer a promising approach but are rarely model-agnostic and also still predominantly developed for classification problems. Instances of such approaches are also rarely benchmarked in real-world problem settings.

- The application of concept drift adaptation in **productive machine learning system solutions** is under-researched. While theoretical strategies for detection and adaptation begin to receive research attention, their integration into system architectures to sustain estimator robustness is rarely addressed and associated descriptions critically lack clarity and standardization.

4.4 Summary

This chapter introduces existing state-of-the-art concept drift detection and adaptation approaches, with foci on the use of machine learning ensemble methods and applications for regression problems. It highlights that recent research attends to the detection of concept drift occurrences in data streams with varying access to ground truth. In addition, it addresses how well a range of properties and benefits attributed to forming concept drift detection ensembles are explored so far. Finally, strategies proposed to maintain the robustness of productively employed estimators under evolving conditions are overviewed.

Beyond outlining the landscape of established approaches, the chapter also details how different research strands interact and where their respective boundaries lie. These primarily concern algorithmic design aspects, overarching learning paradigms as well as system-level considerations. In doing so, it becomes evident that several sub-fields have matured considerably. Most notably, these include classification-oriented concept drift detection and implicit adaptation. Particularly regression-oriented methods, concept drift detection ensembles and practically applicable explicit adaptation strategies, in contrast, remain comparatively under-researched.

In a precise fashion, the associated concrete research gaps are then formulated. These gaps not only delineate the limitations of existing work but also provide a structured motivation for the practical and methodological relevance of the contributions developed in this thesis. The following chapters therefore directly build upon the shortcomings identified here, addressing them both conceptually and empirically.

5 Robust Productive Machine Learning by Design

"That we do not discover reality but rather invent it is quite shocking."
— Paul Watzlawick, scientist on the field of radical constructivism[5]

In the productive, i.e. industrial, machine learning context, an estimator is required to constantly provide high-quality outputs over extended periods of time, despite the destructive impact of concept drift (Woźniak 09.10.2023). Sustaining robustness by treating concept drift is therefore vital in e.g. high-stakes applications such as medical diagnostics, autonomous driving, but also with respect to decision-making in business environments (Bifet et al. 2018, pp. 3–7). In these scenarios, unreliable estimators can cause misdiagnoses, endanger lives, or incur substantial costs due to erroneous outputs (Javed et al. 2025, Kwak et al. 2024, Li 2024).

Although inference process systems, i.e. machine-learning-based system solutions, become increasingly widespread in practice (Chui et al. 2022), concept drift adaptation approaches are rarely considered in their design. Several factors play a role for this shortcoming, as additionally visualized in Figure 5.1: Firstly, the research on the field of concept drift adaptation progressed slowly leading to a decelerated consideration of associated strategies (Vela et al. 2022). Secondly, adopting insights from this research field is non-trivial, as many components of existing inference process systems exhibit complex interdependencies with prototypic concept-drift-adaptation-related ones. Far-reaching system architecture changes are therefore to be expected (Kreuzberger et al. 2023), while appropriate standards, which would support the transfer into practice, do not yet exist as pointed out in the course of this chapter. Furthermore, given that inference process systems are profoundly different from other application contexts (Amershi et al. 2019), the obvious potential of reusing insights from traditional software design practice is severely limited, and sufficiently skilled personnel is still quite rare (Anaconda Inc. 2021).

[5] Radical constructivism is a framework for the philosophy discipline epistemology. In essence, this framework as well as Watzlawick's statement represent the view that individuals learn, i.e. construct new knowledge, strictly based on their existing knowledge. This, inherently, detaches them from an objective reality but helps them to function in their environment by maintaining a subjective interpretation of reality conceived via experiences (von Glasersfeld 1990).

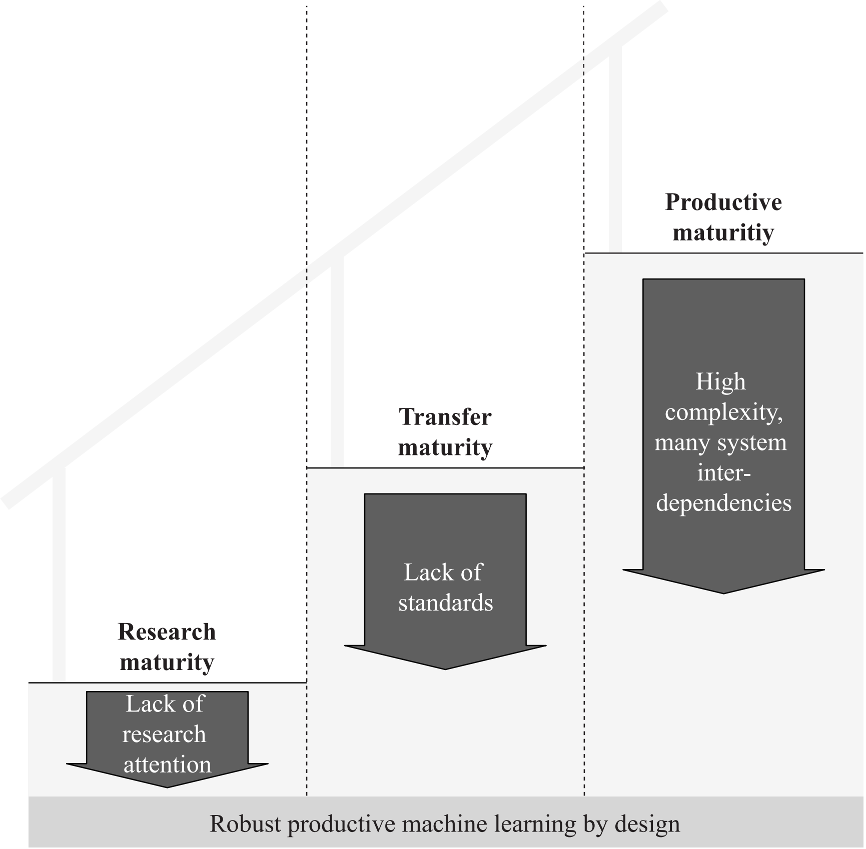

Productive maturitiy
High complexity, many system inter-dependencies
Transfer maturity
Lack of standards
Research maturity
Lack of research attention
Robust productive machine learning by design

Figure 5.1: Influencing factors for the hesitant adoption of concept drift adaptation in practically employed inference process system design. The aggravating effect of these factors is figuratively represented by the downwards-pointing arrows on the demanding ascent towards productive maturity after successively reaching research and transfer maturity.

To effectively support appropriate productive system design, among other things, it is vital to better capacitate responsible engineering practitioners of various backgrounds with respect to robustness. They need to be able to implement robustness-preserving measures during the design of inference process systems as well as acquire deep comprehension of the associated operational intricacies. To achieve this, the author of this thesis posits that adopting a model-centric perspective is expedient to better transfer concept drift adaptation research into practice. This chapter, hence, suggests an approach to structure concept-drift-adaptation-related insights that go beyond merely algorithmic advances. In the form of artifact-based and visually accessible results, descriptive models can systematically condense such aspects and provide these for various system-related concerns. These include the guidance of practical system development, design, management and maintenance, but also, especially given the fact that multiple experts need to cooperate, the avoidance of ambiguities, misconceptions and miscommunications.

The research questions addressed in this chapter are operationalized in Section 5.1. Central concepts around robustness and inference process systems are formally characterized in Section 5.2. The approach for exploring the state of science and practice on robust inference process systems and for identifying associated limitations, as well as gathered insights are presented in Section 5.3 and 5.4, respectively.
A systematically designed descriptive model in the form of a reference architecture constitutes this chapter's core. This reference architecture implements the above-described model-centric perspective with its practical guidance capabilities by means of consistent, clear and informative modeling. Thereby, it addresses the identified limitations in a standardized way. Apart from its design process, the reference architecture is characterized with all of its components and is additionally supplemented by concrete guidelines for practical process and organization-related integration in Section 5.5.
This chapter's Sections 5.4.2 and 5.5 partly contain verbatim passages from a work previously published by the author of this thesis (Trat et al. 2025). Associated passages are not individually cited when the author is responsible for the intellectual content and research insights. Contributions from co-authors as well as figures and tables are explicitly cited.

5.1 Research Questions

While this thesis suggests multiple avenues to sustain productive estimator robustness, this chapter explores them within a system-design-oriented context while adopting a primarily model-centric perspective. Two high-level research questions therefore group consecutive scientific concerns associated to machine learning robustness as it is contextualized here. They accomplish the separation between identifying the problem space and proposing as well as evaluating conceptualized solutions. More concretely, one focuses on formally describing robustness within this context to equip the other with the necessary foundation to propose a practically applicable framework. They are formulated as follows:

RQ I What are the challenges in sustaining the robustness of productive machine learning estimators from a practical model-centric perspective?

RQ I-1 How can robustness be formally characterized?

RQ I-2 What are the limitations of existing frameworks supporting the design of robust machine learning solutions?

Research question **RQ I** is directed towards prevailing challenges regarding robustness being sustained from a model-centric perspective. It is operationalized by firstly posing **RQ I-1** to explore fundamental concepts and make these formally accessible for this thesis. Research question **RQ I-2** then identifies the limitations of the state of science against this formalized background.

RQ II Can a reference architecture support to sustain robustness of machine learning estimators from a practical model-centric perspective?

RQ II-1 Can Subject Orientation be employed to formally model a reference architecture for inference process systems with robustness-sustaining components?

RQ II-2 What are the practical advantages of such a reference architecture?

RQ II-3 How can such a reference architecture be integrated into practice?

Research question **RQ II** pursues reference architectures to overcome the above-outlined challenges. To provide evidence to this research question, first **RQ II-1** needs to be approached to validate the selected descriptive modeling tool. Only then, **RQ II-2** and **RQ II-3** can be approached to validate the resulting artifact, which takes the form of a reference architecture, as well as carefully examine its suitability in practical contexts.

5.2 Robust Inference Process Systems

After deploying an estimator to a production environment, concept drift can critically erode its robustness. To establish a consistent understanding throughout this thesis, and guided by **RQ I-1**, the following subsection formalizes the notion of robustness. Additionally, the system context that integrates estimators is defined.

5.2.1 A Definition of Robustness for Machine Learning Estimators

The notion of robustness shall not only be formalized from a theoretical perspective but also by identifying quantifiable aspects of it. When assessing the robustness of estimators, one might consider the performance of the detection approach employed to identify harmful concept drifts. The literature suggests a range of metrics to quantify this performance. They typically set the detected concept drifts and ground-truth information on concept drifts or detection delays in relation to each other (Cerqueira et al. 2022, Bifet et al. 2013). However, such ground-truth information on concept drift occurrences in the data is typically difficult or even impossible to obtain (Pesaranghader et al. 2018).

As highlighted by Bifet (2017), exclusively considering concept-drift-detection-related metrics might result in a one-sided perspective and even be misleading with respect to estimator performance. Therefore, machine learning performance metrics, such as accuracy, F1-score for classification and a range of error metrics for regression, enable more fundamental insights into robustness. Section 3.1.3 provides an overview of such metrics. These metrics provide a means for evaluating estimator performance under various conditions and tracking it over time. They also allow the evaluation of concept drift adaptation measures.

Definition 4 attempts to combine the existing considerations of Bifet et al. (2018, p. 7), Studer et al. (2021) and Ott (2024). It is used to formally describe the notion of robustness for the purpose of this thesis and provides a response to **RQ I-1**.

Definition 4 (Machine learning estimator long-term robustness)**.** The robustness of a machine learning estimator M is maintained if it consistently achieves high performance on unseen data from a stream that exhibits evolving characteristics. This holds when performance, quantified using appropriate metrics, remains high with minimal variance over time.

5.2.2 A Definition of Inference Process Systems

Generally, within application contexts, machine learning estimators cannot operate in isolation. To serve as a deployable solution, they rely on a functional context that includes essential components to manage incoming and outgoing data. For this reason, it is useful to conceptualize structures that are referred to as inference process systems within this thesis. Building on the work of Bifet et al. (2018, pp. 6-11) and Kulbach et al. (2022), an inference process system can be defined as follows:

Definition 5 (Inference process system)**.** An inference process system is a structure that encapsulates a machine learning estimator M. It includes interfaces for accepting incoming data batches $\mathbf{X}_t$ and for making estimates y_u available for downstream processing at times t or u. It also includes routines and substructures for preprocessing data into a form suitable for inference as expected by M.

To further enhance robustness, an inference process system satisfying Definition 5 and integrating components designed to conduct concept drift adaptation can be referred to as robust inference process system.

5.3 Systematic Literature Review

A systematic literature review, as introduced in Section 2.1 is conducted to create the basis for researching descriptive models of robust inference process systems. With its three steps, namely gathering potentially relevant publications, filtering out irrelevant ones and analyzing the remaining ones, it is visualized as flowchart in Figure 5.2. This flowchart specifies the abstract one in Figure 2.1 for application of the process in this instance.

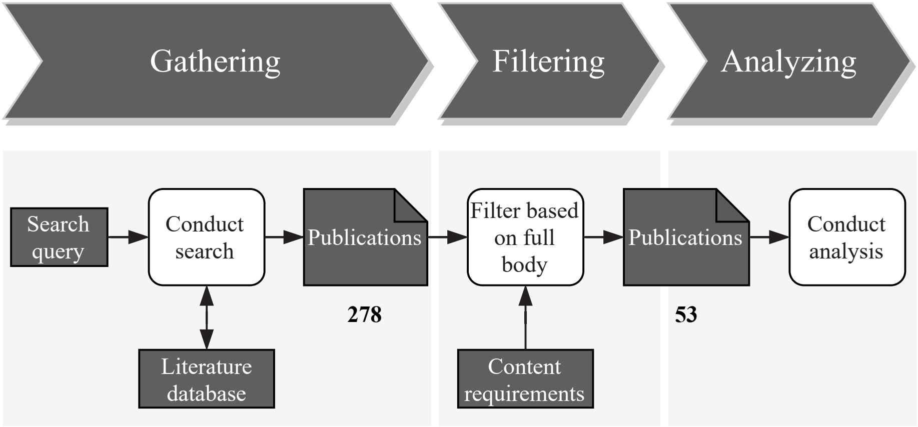


Figure 5.2: Flowchart representation of the systematic literature review applied to survey the state of science on concept drift adaptation approaches. It involves gathering, filtering and analyzing publications. Being a specification of Figure 2.1, details of the filtering-related sub-step are named in the respective flowchart elements and quantities of retrieved, filtered and analyzed publications are provided as bold-formatted numbers.

Gathering

To retrieve potentially relevant and high-quality peer-reviewed publications, the curated abstract and citation database Scopus[6] is used. It indexes a large number of publication databases such as ACM, IEEE and Springer. This is done by using the query string printed in Table 5.1. To be selected via this query, publications have to employ the term *concept drift* in their title, abstract

[6] Available at `https://www.scopus.com/search/form.uri`, accessed: 14.02.2024.

or keywords. Furthermore, terms describing process systems, such as *system*, *architecture* and *process*, are required. Following the assumption that the density of publications suggesting practically implementable process systems increases as time progresses and to limit the number of publications, only such published in 2021 or later are considered. This way, 278 publications are gathered.

Table 5.1: Details on gathering publications on concept drift adaptation approaches. Using the meta research database Scopus, a search query is designed to identify specifically such approaches that aim at increasing the long-term robustness of machine learning estimators in implementable process systems.

Literature database	Scopus

String	`TITLE-ABS-KEY(concept PRE/0 drift)`
	`AND TITLE-ABS-KEY(drift W/15 adapt*)`
	`AND TITLE-ABS-KEY(system OR architecture OR process)`

Conditions	Publication during or after year 2021

Filtering

Within the context of this review, a publication provides relevant research if it

- suggests a process system that employs explicit concept drift adaptation.
- suggests a process system that is implementable, i.e. comprises all components required for the full range of functions to conduct inference in a data stream. This requirement is not fulfilled, if integral components are omitted.
- provides a description for the suggested process system. It is either visualized in the publication body, appendix or in any external document referenced by it.
- is written in English language.

By scrutinizing the initially retrieved 278 publications with respect to these relevance criteria, 53 publications are identified as relevant. As process system descriptions typically appear in the text body of the publication, this component is screened during filtering. Primarily the absence of such descriptions in a considerable portion of the publications leads to these being filtered out.

Analysis

The 53 publications identified as relevant are then subject to a qualitative and a component-oriented analysis. They are conducted to further the understanding as well as the development of descriptive models of robust inference process systems. To gather a common data basis for these, a set of content attributes are defined and populated for each publication. They are provided in Table A.1 in the appendix of this thesis.

5.4 Limitations of Existing Descriptive Models

Model-driven approaches are employed in practice to reduce complexity, facilitate collaboration and enable reproducibility in the development of inference process systems. Furthermore, augmenting such systems with robustness-sustaining measures and providing associated descriptive models is subject of active research. Motivated by **RQ I-2**, this section characterizes these approaches and highlights their shortcomings.

5.4.1 Data Science Process Models

Developing a machine learning estimator and deploying it to a production environment as a component of an inference process system is a highly complex process, involving a broad range of tasks and responsibilities beyond merely estimator-related ones. Data science process models are descriptive models that integrate these tasks into process-centric iterative frameworks that aim at guiding data science projects. Several instances from the literature and practice are detailed in Section 3.3. In the following, an abstract view as well as limitations of these models are outlined.

High-Level Phases of Data Sciences Process

Data science process models can provide valuable starting points for developing inference process systems in a technology and industry-independent manner. In general, their frameworks need to consider five phases (Provost and Fawcett 2013, pp. 27–33), (Awad and Khanna 2015, pp. 5–6). They are illustrated in Figure 5.3 and detailed in the following paragraphs.

Business context analysis and problem formulation. Initially, a profound comprehension of the underlying business objectives is required. This supports to clearly define the problem the machine learning estimator is intended to solve.
Data-related processes. These processes start by gathering associated data, e.g. from enterprise

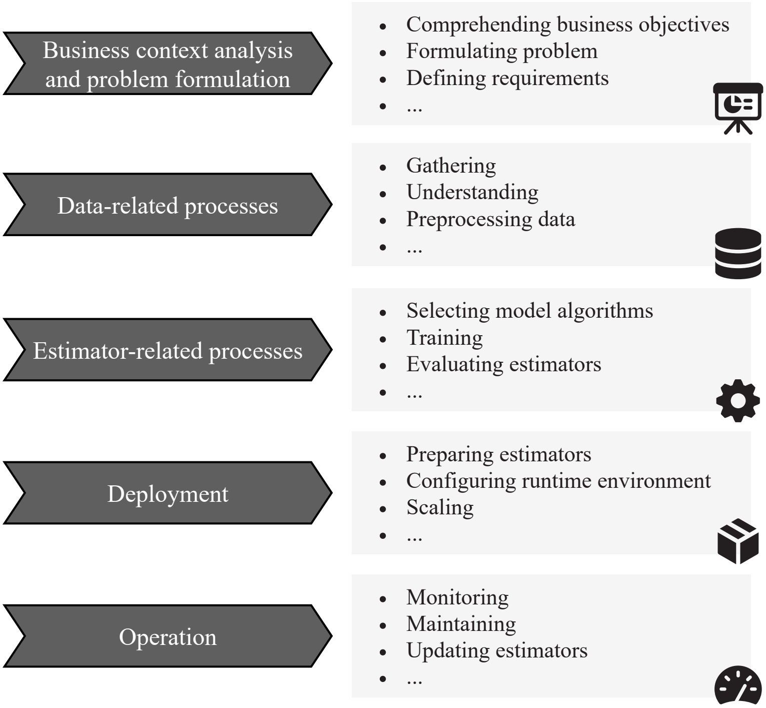


Figure 5.3: High-level phases of data science processes. On an abstract level, a data science project can be modeled as a process that traverses five phases, depicted in the left column, to output a productive machine learning estimator. As depicted in the right column, each phase can be broken down to a broad range of associated tasks.

systems or sensors. Subsequently, the data needs to be understood, evaluated in terms of quality and preprocessed for further use.

Estimator-related processes. Often, a significant share of time and effort is invested in processes associated to machine learning estimators. These involve the possibly automated selection of suitable algorithms as well as estimator training and evaluation.

Deployment. The onset of this phase marks a turning point, as an estimator is considered ready for being transferred to a production environment. This involves integrating it with existing infrastructure as well as ensuring its scalability.

Operation. After successful deployment, an estimator commences its productive phase. During this phase, monitoring its condition and updating it is vital as several issues, such as concept drift, may arise.

Shortcomings of Data Science Process Models

Data science process models are designed to provide general guidance rather than practically applicable granular advice. Therefore, they do have several critical limitations with regard to sustaining long-term robustness. Table 5.2 evaluates popular instances of data science process models with respect to several inference-process-system-related aspects relevant during its operation. Most instances do not explicitly account for the complexities that come with the occurrence of concept drift. Most severely, the danger of it causing performance degradation is mainly neglected. In large part, they also fall short in supporting the concise formulation of robustness-preserving maintenance strategies. CRISP-DM, one of the most widely adopted instances, considers performance-monitoring-related tasks but effectively terminates its process with the deployment phase (Wirth and Hipp 2000, Chapman et al. 2000). Partially as a response to CRISP-DM's shortcomings, the more recent instances of data science process models PAISE and CRISP-ML(Q) stand out by providing increased support for operating inference process systems. They explicitly consider the potential impact of concept drift and declare the necessity of countermeasures as part of maintenance strategies (Hasterok et al. 2021, Studer et al. 2021). However, they do so on a high level and do not specify detailed approaches.

Apart from that, most instances do not define any personnel roles and responsibilities with respect to inference process systems during the operation phase. This omission leaves monitoring and maintenance-related tasks unassigned, resulting in gaps in accountability. In contrast, CRISP-DM and PAISE include the handover of inference process systems with the associated responsibilities to a third party. In these models, however, they are transferred to the third party as a whole. It is then up to the latter to establish specific personnel responsibilities (Wirth and Hipp 2000, Chapman et al. 2000, Hasterok et al. 2021).

Beyond that, no data science process model suggests system architecture design approaches for sustaining long-term robustness during the operation of inference process systems. While not leading to an erosion of the models' technology independence, suggesting such would hardly be compatible with their process-model-centric approach. Therefore, one might argue that considering a system-model-centric approach can be a suitable addition to data science process models.

5.4.2 Robust Inference Process Systems

The scientific field of concept drift adaptation has only in recent years begun to emerge as a topic of interest. In particular, explicit approaches can help increasing the long-term robustness of machine learning estimators in inference process systems (Gama et al. 2014). An introduction to the field of concept drift adaptation is provided in Section 3.2.3.

Table 5.2: Coverage of post-deployment machine-learning-estimator-related aspects within data science process models. The first column lists a set of data science process models, chronologically sorted. The second through fourth column evaluate their consideration of performance degradation, their specific reference to maintenance-related tasks and responsibilities, respectively. The more segments of a circle are colored in black, the more the respective aspect is fulfilled by a model.

	Performance degradation	Maintenance-related tasks	Explicitly assigned responsibilities
CRISP-DM	◔	○	◔
Analytics Solutions Unified Method for Data Mining	◔	◑	○
PAISE	●	◕	◔
CRISP-ML(Q)	●	◕	○
Team Data Science Process	◔	○	○
Well-Architected Machine Learning Lifecycle	◔	○	○

Designing robust inference process systems or augmenting existing ones with robustness-sustaining components is a highly complex task. It is therefore crucial to descriptively model such robust systems in a self-sufficient way devoid of ambiguities. In the following, a qualitative analysis of associated approaches from scientific literature is presented. From this, insights into their limitations are obtained.

Scope of the Analysis

The qualitative analysis is conducted on the publications identified as relevant, which constitutes a part of the systematic literature review introduced in Section 5.3. Initially, the method used to create a descriptive model of the suggested inference process system with its functional components is identified. It shall be noted that the functional components themselves are not analyzed in detail in the course of this step. This aspect is subject of the component-oriented analysis provided later in this chapter. Subsequently, the degree of coherence offered by the descriptive model is evaluated. Within this context, coherence refers to its consistency, clarity and informative content. The evaluation is done by qualitatively determining how well the process system's intricacies are conveyed in a self-contained manner without the need for detailing textual explanation. Also, modeling elements contributing to the observer's confusion are noted.

Shortcomings of Descriptive Models

All publications identified as relevant suggest valuable scientific contributions to the field of concept drift adaptation. An overview of these can be taken from Section 4.2.2. However, an analysis of the respectively provided descriptive models reveals several shortcomings. These are outlined for selected representatives in the following. The descriptive models associated to these representatives are depicted in Figure 5.4.

The inference process system proposed by Casado et al. (2023) targets classification problems and relies on the use of machine learning ensembles. The authors furthermore aim to consider the intricacies of federated-learning scenarios in their design.
The analysis reveals that the descriptive model (cf. sub-figure (a) of Figure 5.4) is based on a well-defined flow chart with a procedural character (Trat et al. 2025). It displays a series of highly specific actions and decision switches consistently represented as rectangles and diamonds, respectively. One questions that could be raised addresses the role of the free-form element "data stream". In the text body it is described as supplying data as input for preprocessing and subsequent estimator model training as part of a concept drift adaptation measure. Logically it should also supply data for the estimator inference, which is not shown. Also, it is unclear whether samples from the data stream or measures computed from estimator predictions serve as input for concept drift detection. Ambiguities with respect to the type of detector input are also a matter of other descriptive models. In one instance, the descriptive model supports the assumption that the detector directly receives the estimator output but actually receives the error computed from the output (Angbera and Chan 2024). In other instances, the detector input is either unclear, which results from the descriptive model critically lacking details (Li and Zhao 2024, Wilson et al. 2023) or from the selected level of process abstraction (Suryawanshi et al. 2024).
Another question that could be posed with respect to the inference process system of Casado et al. (2023) concerns practical responsibilities. Who is tasked with executing the described process and when? No modeling elements explicitly consider the personnel-related aspect. Regarding the timing, concept drift adaptation is a continuous process, whereas this descriptive model exhibits no loops. The logical assumption therefore is that this descriptive model represents the internal activity flow of the process system and is traversed or instantiated each time a data batch is somehow provided to it.

Hu (2022) suggests an inference process system involving an active-learning mechanism triggered by concept drift detections. This mechanism involves requesting labels from a human expert and buffering these in a sliding reservoir for delayed labels "SRADL". The label demand is economized by applying a semi-supervised learning approach. Concept drift occurrences are identified by employing an ensemble-like structure of detectors. Evaluating the proposed system on various datasets is reported to perform well, even in the case of low and delayed label

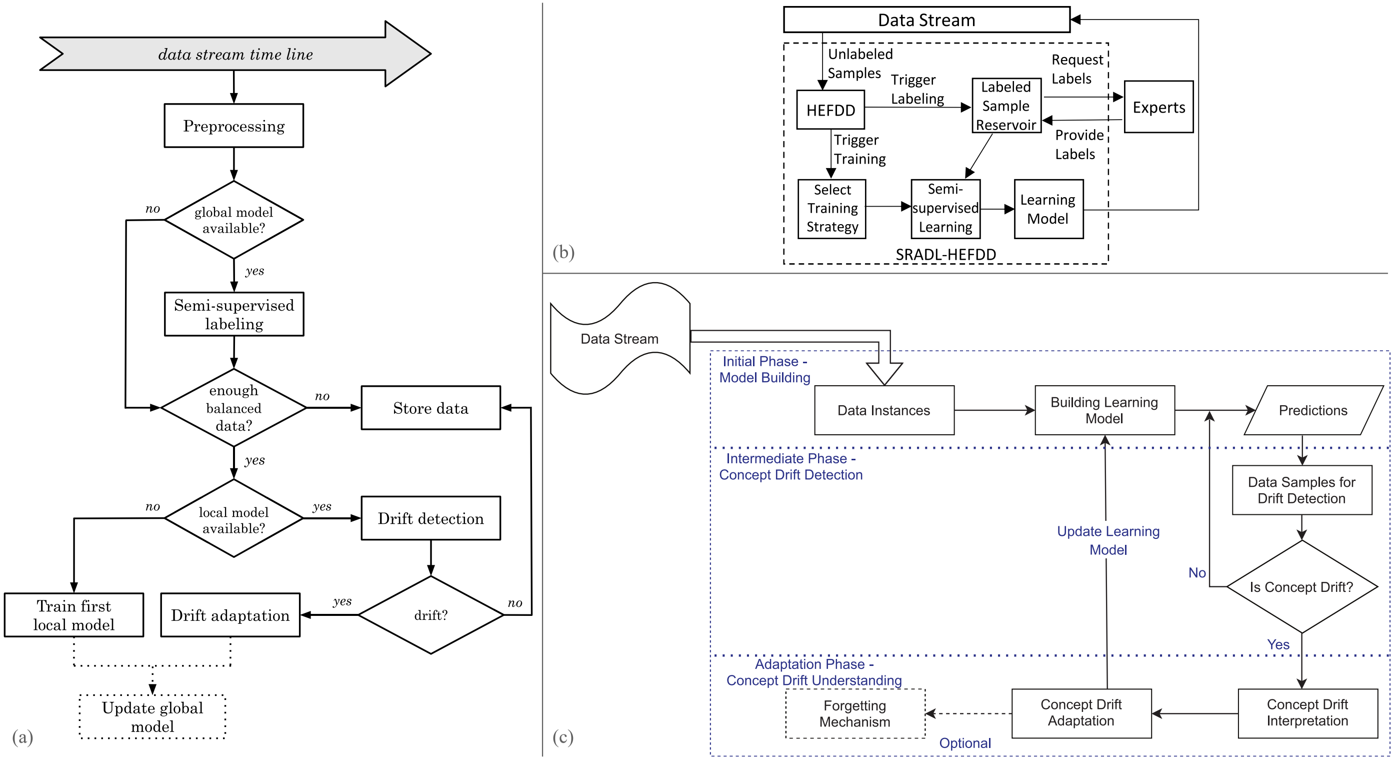


Figure 5.4: Existing descriptive models of inference process systems from the literature; compilation based on Trat et al. (2025). Each of the process systems employs concept drift adaptation to increase the robustness of contained machine learning estimators. (a) a flow chart to describe adaptation processes (Casado et al. 2023), (b) free-form modeling of elements for active as well as semi-supervised-learning processes (Hu 2022), (c) separating adaptation processes into distinct phases of a free-form model (Agrahari and Singh 2022). Several aspects in examples, such as the choice of labeling terms, shape and representation consistency, reduce the modeling coherence. This leaves space for ambiguity and can cause confusion for the observer.

availability.
An analysis of the associated descriptive free-form model (cf. sub-figure (b) of Figure 5.4) reveals the following aspects reducing its clarity. Based on the annotations, it is unclear whether the arrows and rectangles represent objects, such as data, or actions, such as the requesting and providing of objects. Some arrows are not annotated at all. In the part of the model containing the arrow "Unlabeled Samples" that enters the concept-drift-detector-representing rectangle labeled "HEFDD", arrows seem to represent objects or results and rectangles actions. However, in other parts, the opposite holds. For instance, the rectangle "Labeled Sample Reservoir" represents an object. It is connected with the arrows representing the actions "Request" and "Provide Labels". Other publications share the same notation inconsistency regarding both the employed rectangle and arrow elements (Din et al. 2024) or only rectangle elements (Ding et al. 2021, Dong et al. 2022, Angbera and Chan 2024, Li and Zhao 2024).

Agrahari and Singh (2022) survey the state of science on concept drift adaptation with an emphasis on concept drift detectors. They outline associated research trends as well as gaps and suggest a taxonomic characterization of their different algorithmic approaches.
They also propose a descriptive free-form model representing an abstract inference process system (cf. sub-figure (c) of Figure 5.4). It accumulates various concept-drift-adaptation-related aspects and allocates these to three distinct *phases*. This phase-based design can be interpreted as indicating a linear process. This interpretation is backed up by most but not all rectangles that contain descriptions of actions and are connected by typical process-flow-indicating arrows. Other rectangles contain descriptions of data objects. This inconsistency leaves space for interpretation: For instance, are data samples *created, prepared, collected, reduced* or similar (Trat et al. 2025)? Apart from that, not all activities are described inside of rectangles, such as "Update Learning Model". The "Initial Phase" is explicitly dedicated to estimator model building. However, it also contains the inference process represented as parallelogram labeled "Predictions". This contradicts the textual explanation stating that building and inference occur at different times as well as on different data. Furthermore, logically speaking, inference should be conducted continuously, on each incoming data batch, and not only initially. This raises the question whether the phase separation applies to all actions enclosed in associated areas or merely to some of them. Concept drift detection is described to occur in an "Intermediate Phase", which itself is followed by an "Adaptation Phase". However, especially given that process phases are usually traversed linearly, the question could be posed whether the authors rather mean to describe a continuously running process system with activity groupings. In that case, the use of the term *phase* is debatable. Missing a "Final Phase" is another associated aspect when not considering the unexplained branch flow into the "Forgetting Mechanism" rectangle (Trat et al. 2025), which denotes a highly relevant aspect of maintaining various stream concepts. Additionally, the latter is modeled as a dead-end as it only features an ingoing arrow but no outgoing one.

With regard to the rectangle notation, this descriptive model exhibits the same inconsistencies as the ones outlined for the model of Hu (2022). On several occasions, rectangles and parallelograms denote objects. On others, they denote actions. Also, it shares the inconsistencies regarding concept drift detector input with the model of Casado et al. (2023). Based on the explaining text, raw data samples are utilized as input. The model, however, does not display an arrow from any data source to the diamond shape indicating the concept drift detection process. Therefore, the origin of the samples for concept drift detection as well as the time they are gathered remains open. Based on further aspects outlined in the text, the estimator predictions or a performance-related measure computed from these would also serve as feasible detector input. The arrow-based link between both shapes in the model even hints towards that notion, which contributes for ambiguous interpretation.

Summary of the Analysis

The preceding subsection provides a qualitative analysis of publications identified as relevant. They propose several concept drift adaptation approaches with the common goal of increasing the robustness of inference process systems in practical problem scenarios. Among the different methods employed to create descriptive models of such systems, free-form flow charts are predominant. Some publications augment this method with additional custom procedural modeling elements. Others create descriptive models without following any standardized method.

The vast majority of the analyzed descriptive models does not sufficiently enclose the complexity of various inference process system designs. This is concluded either based on a lack of formality, precise notation or consistency regarding modeling elements. Various system aspects require detailing textual explanations for an adequate degree of comprehensibility and, on multiple occasions, they even stand in conflict with the descriptive models. As a result, this leaves considerable space for ambiguities which has the potential to severely limit the coherence of the models. Together with the previously outlined limitations of data science process, these insights into state of science approaches provide an answer to **RQ I-2**.

5.5 A Reference Architecture for Robust Inference Process Systems

The previous section outlined, among other aspects, the limitations of state-of-the-art methods for descriptively modeling robust inference process systems in a coherent way. This section aims at overcoming these limitations while retaining the model-centric perspective established

throughout the surrounding chapter. It provides concrete support for system implementation with a distinct focus on sustaining the robustness of machine learning estimators by means of concept drift adaptation.
As solution approach, the form of the reference architecture is selected. This type of model is employed to abstractly describe complex technical, e.g. software, architectures while simultaneously retaining technology independence and reusability (Reidt 2019, pp. 21–25). A reference architecture bundles practical expert knowledge as well as technical detail and aims at straightforwardly facilitating concrete system implementations (Reidt et al. 2018).

Driven by **RQ II**, this section presents the iterative process designed to create a reference architecture for robust inference process systems as well as the artifact resulting from it. As novel approach within this context, the modeling method Subject Orientation is utilized. The artifact, i.e. the suggested reference architecture[7] is described in detail with all of its components in this section and is displayed in Figure 5.6 and figures throughout Subsection 5.5.2. It facilitates understanding, describing, designing, implementing and maintaining robust inference process systems. Also, it is intended to be specialized for a wide range of practical use cases with an inherently high degree of coherence.
The suggested reference architecture is validated in a two-fold fashion and concrete guidelines for its practical application are presented. In this regard, it can complement widely employed data science process models by introducing a concrete and structured approach to robustness. Simultaneously, the research gap regarding exactly where and how concept drift adaptation approaches can be embedded in this practical system context is addressed. This way, the suggested reference architecture shall serve as a standardized framework accessible for a wide range of practitioners responsible for inference process systems in industrial companies of various sizes. It shall also be remarked that this scientific contribution is presented in a work (Trat et al. 2025) separately published by the author of this thesis.

5.5.1 Design Process

A carefully designed process is followed to inductively produce the reference architecture for robust inference process systems proposed in this chapter. The process is of an iterative nature and condenses different sources of knowledge. The modeling method Subject Orientation, applied via the language Parallel Activity Specification Schema, constitutes its core. An introduction to this method can be taken from Section 3.4.

[7] An editable version of the full reference architecture for application in practice is openly available at `https://github.com/m-martin-j/CDA-systems-ref-arch`.

Process Steps

The process consists of the following steps, which are additionally displayed in Figure 5.5[8]:

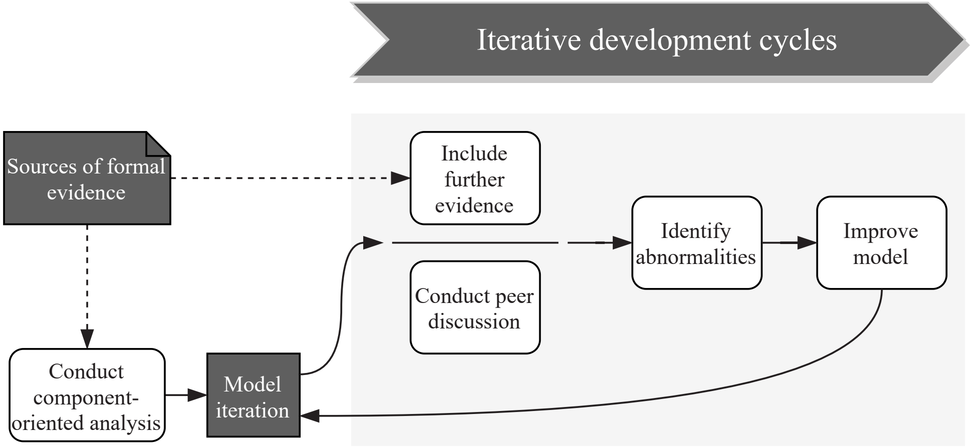


Figure 5.5: The iterative design process employing the modeling method Subject Orientation to produce a formal reference architecture for robust inference process systems. It begins with an initial model draft that can be arbitrarily further improved by using additional formal evidence or conducting discussions with peers.

- First, the process is initiated with a draft model. It is the result of applying Subject Orientation on a set of a priori available sources of formal evidence.
- Subsequently, the process can be continued with arbitrarily many iterative cycles of further improving modification. Within each cycle, further formal evidence can be considered or discussions with peers can be conducted. This way, abnormalities, such as errors and imprecise modeling choices, can successively be reduced.
- At the end of each cycle, additionally obtained knowledge is included in the Subject-Oriented model.

[8] One could be struck by a certain irony of the fact that Subject Orientation was not practiced for producing this figure. The author of this thesis, however, believes that a flowchart-based model suffices to capture the complexity of the iterative process producing a formal reference architecture. He bases this on its relative simplicity when compared with a robust inference process system.

Process Execution

For creating the initial draft of the reference architecture for robust inference process systems, two sources of systematically gathered formal evidence are considered:

- **Insights from a component-oriented analysis of recently published work on concept drift adaptation approaches as robustness-increasing component of practically implementable instances of inference process systems**: This way, specifics also of other functional components besides concept-drift-adaptation-related ones are assembled and practically relevant boundary conditions, i.e. varying foci and problem contexts, are considered in the reference architecture design. The works are gathered within the course of an exhaustive systematic literature review, which is introduced in Section 5.3. They are overviewed in Section 4.2.2 and qualitatively analyzed with regard to modeling coherence as well as the respectively utilized approach to describe them in Section 5.4.2.
- **Significant scientific literature on concept drift adaptation fundamentals**: Including this source of evidence aims at introducing substantiated knowledge that heavily influenced actors of the entire scientific field into the development of the reference architecture. More specifically, the works of Gama et al. (2014), Khamassi et al. (2018), Iwashita and Papa (2019), Gemaque et al. (2020) and Lima et al. (2022) are considered.

The synthesis of these two sources of formal evidence constitutes a substantial body of knowledge from literature. Therefore, exclusively peer discussions were conducted during a total of six subsequent improvement iterations. Four of them were conducted by the authors of Trat et al. (2025). After these, two additional iterations were initiated within the context of the 58th Hawaii International Conference on System Sciences 2025. They took as input the questions as well as valuable feedback obtained during the double-blind peer review and from the audience of the subsequent presentation of the associated paper.

Limitations

The choice of Subject Orientation as modeling method is motivated by previous successes of its application to describe rather complex concepts while still resulting in coherent accessible models. It is also motivated by the fact that it enables system designers to well include human roles operating the modeled target systems (Elstermann et al. 2021, Kurtz et al. 2022). Still, this choice can be classified as an arbitrary one. Within this thesis, no other method is employed to formulate an alternative reference architecture that could be compared with respect to modeling coherence.

Also, the proposed reference architecture is based on and accumulates aspects from existing concept drift adaptation approaches. As outlined above, they are extracted from recent works on the associated research field. This way, a possible bias of the reference architecture towards exclusively recent concept-drift-adaptation-related considerations cannot be excluded. This limitation, however, is alleviated by also integrating knowledge from earlier influential works that significantly impacted the design of the reference architecture.

The research questions **RQ II-1** and **RQ II-2** shall therefore be raised and addressed within the course of the remaining section. **RQ II-1** primarily concerns the feasibility of this method for the task at hand. **RQ II-2** builds upon the former and is designed to work out the resulting reference architecture's benefits for practical use.

5.5.2 The Proposed Reference Architecture

Following the previously described process produces a reference architecture for robust inference process systems. The considered formal evidence comprises publications suggesting various systems employing supervised and semi-supervised concept drift adaptation approaches. This is also reflected in the primary application focus of this model.

Overview

The SID of this reference architecture is presented in Figure 5.6. It comprises all relevant subjects, i.e. functional system components, as well as their interaction during different stages of a system's operation. Two tightly conjoined sub-systems are considered by this SID: One primarily responsible for machine-learning-based inference (top) and one for concept-drift-adaptation-based maintenance processes (bottom).

Each subject, as well as the workflows and interactions it is involved in, is formally described by its associated SBD. Interface-type subjects form an exception. Being modeled exclusively regarding their interaction with other subjects, they do not feature SBDs. They are marked with the letter "I" and a red token in Figure 5.6.

Inference Sub-System

The inference sub-system comprises the subjects streaming data source, inference unit and consuming entity. Furthermore, among other roles, the data scientist is responsible for designing

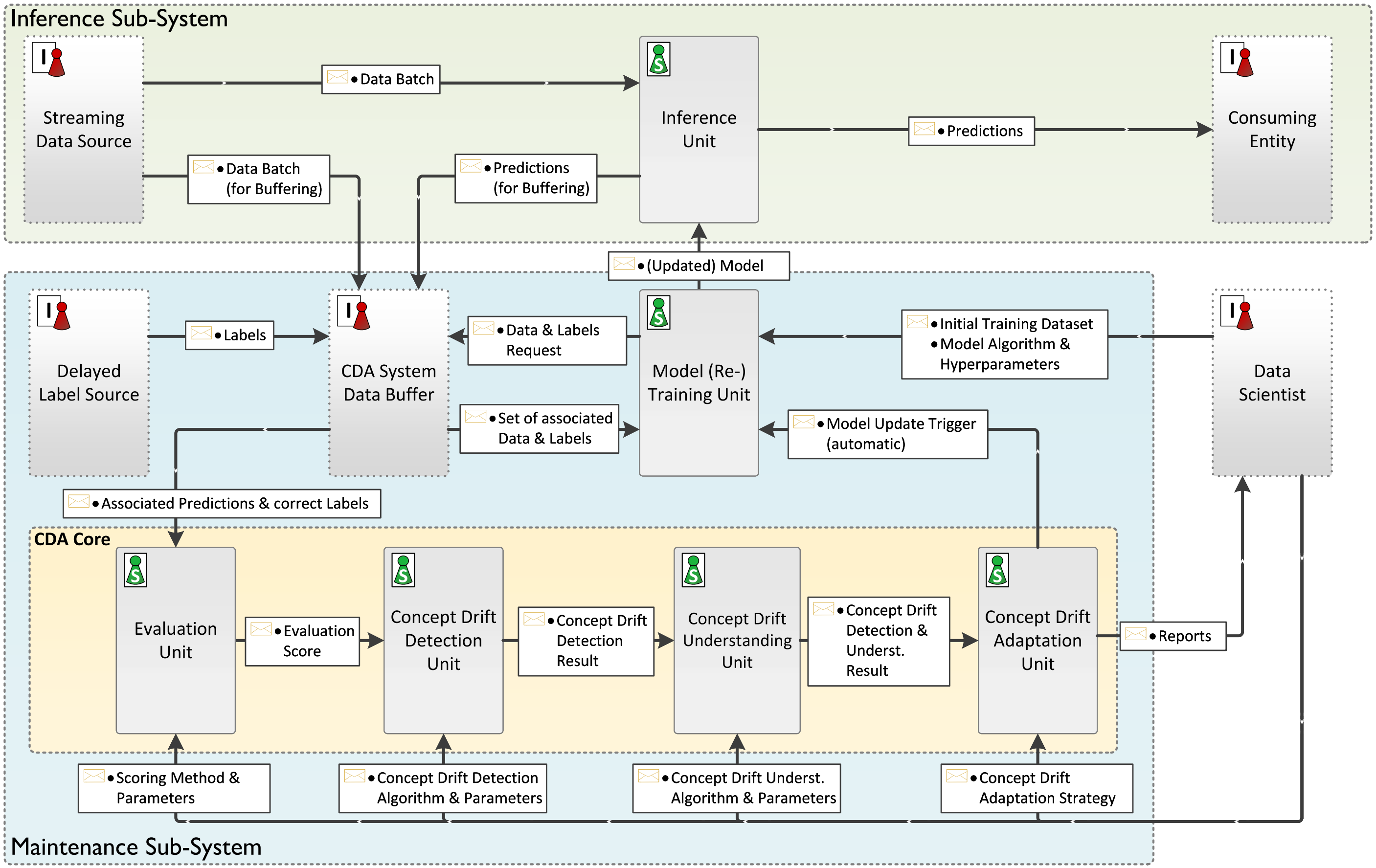

Inference Sub-System
Streaming Data Source
•Data Batch
Inference Unit
•Predictions
Consuming Entity
•Data Batch (for Buffering)
•Predictions (for Buffering)
•(Updated) Model
Delayed Label Source
•Labels
CDA System Data Buffer
•Data & Labels Request
Model (Re-) Training Unit
•Initial Training Dataset
•Model Algorithm & Hyperparameters
Data Scientist
•Set of associated Data & Labels
•Model Update Trigger (automatic)
•Associated Predictions & correct Labels
CDA Core
Evaluation Unit
•Evaluation Score
Concept Drift Detection Unit
•Concept Drift Detection Result
Concept Drift Understanding Unit
•Concept Drift Detection & Underst. Result
Concept Drift Adaptation Unit
•Reports
•Scoring Method & Parameters
•Concept Drift Detection Algorithm & Parameters
•Concept Drift Underst. Algorithm & Parameters
•Concept Drift Adaptation Strategy
Maintenance Sub-System

Figure 5.6: The subject interaction diagram (SID) of the reference architecture for robust inference process systems (Trat et al. 2025). In this context, such a system is one that features concept drift adaptation (CDA) approaches. Alongside a set of subject behavior diagrams (SBDs) presented in figures throughout Section 5.5.2, it is the result of an iterative design process. Both the SID and its associated set of SBDs serve as template that can be specialized for a concrete use case and that supports the implementation of robust inference process systems.

and operating inference process systems. This role is also represented as a subject. These aforementioned subjects are elaborated on in the following.

Streaming data source. An inference process system is utilized to process data. Different sources of such, often streams, can be tapped into. The SBD of a streaming data source needs to be sufficiently abstract to represent a wide variety of different data sources, such as sensors or other systems. As a result, the behavior definition would be trivial and basically just feature the advent and forwarding of data. The streaming data source subject is therefore modeled as interface subject.

Inference unit. The technical core of any inference process system is the inference unit. It pools compute resources and routes input data from their sources, here generalized by the interface subject streaming data source, through preprocessing pipelines to the machine learning estimator holding the current model version. Afterwards, it passes the computed estimations on to a consuming entity and to a data buffer in the maintenance sub-system. Its SBD is depicted in Figure 5.7.

Consuming entity. The Consuming entity represents arbitrary downstream applications. Such utilize the outputs of machine learning estimators for specific tasks, such as decision support, reporting or others. For similar reasons as the streaming data source, this subject is modeled as interface subject.

Data scientist. Data scientists are among the people operating inference process systems. As depicted in Figure 5.6, this involves parametrizing various system units and receiving as well as analyzing reports on concept understanding and adaptation measures. Such can e.g. contain the timing of concept drifts and associated treatment details. Arbitrary other practitioner roles, such as machine learning or data engineers can be represented as subjects within the reference architecture. This subject is modeled as interface subject as the inference-process-system-related responsibilities and actions of a data scientist can vary greatly across use cases.

Maintenance Sub-System

The inference sub-system is vulnerable to concept drift. It makes the existence of a maintenance sub-system necessary to sustain the performance of a machine learning estimator's output. Therefore, the reference architecture also features the subjects concept drift detection, understanding and adaptation unit, as well as the concept drift adaptation system data buffer, delayed label source, model (re-)training unit and evaluation unit. Details on these subjects are provided in the following.

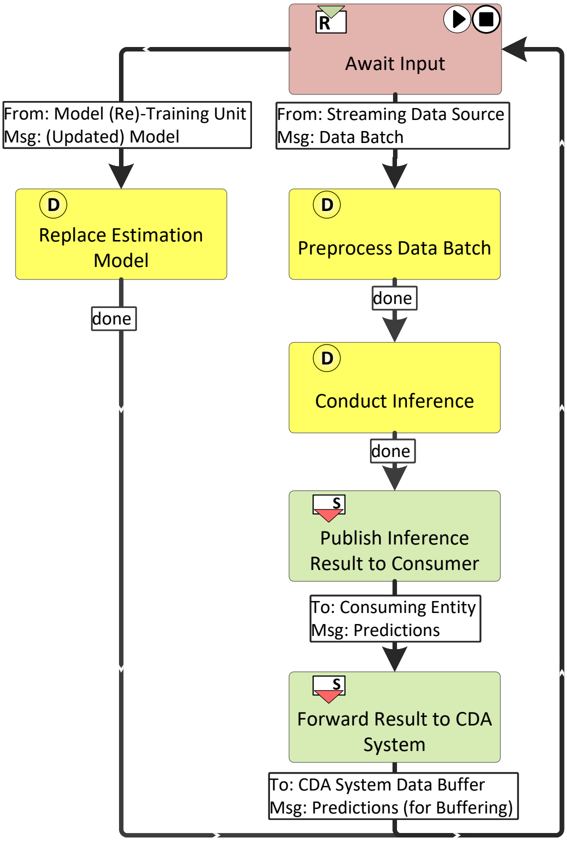

R
Await Input
From: Model (Re)-Training Unit
Msg: (Updated) Model
From: Streaming Data Source
Msg: Data Batch
D
Replace Estimation Model
D
Preprocess Data Batch
done
done
D
Conduct Inference
done
S
Publish Inference Result to Consumer
To: Consuming Entity
Msg: Predictions
S
Forward Result to CDA System
To: CDA System Data Buffer
Msg: Predictions (for Buffering)

Figure 5.7: The subject behavior diagram (SBD) of an inference process system's inference unit (Trat et al. 2025). This system component serves machine learning estimators and processes input data.

Concept drift detection unit. This unit employs concept drift detectors. Given the reference architecture's focus on semi and fully supervised concept drift adaptation approaches, performance-related scores computed from the machine learning estimator's predictions are processed by this unit. Its outputs are forwarded to analyzing and orchestrating subjects of the maintenance subsystem. Figure 5.8 provides its SBD.

Concept drift understanding unit. Concept drift understanding is a matter of active research. In its essence, this research field focuses on identifying root causes of concept drifts and to provide further explanatory insights (Pinto et al. 2019, Xiang et al. 2023). Within the context of robust inference process systems, an associated system component can be considered. As visualized in Figure 5.9, these insights can be exploited for further concept-drift-treating actions.

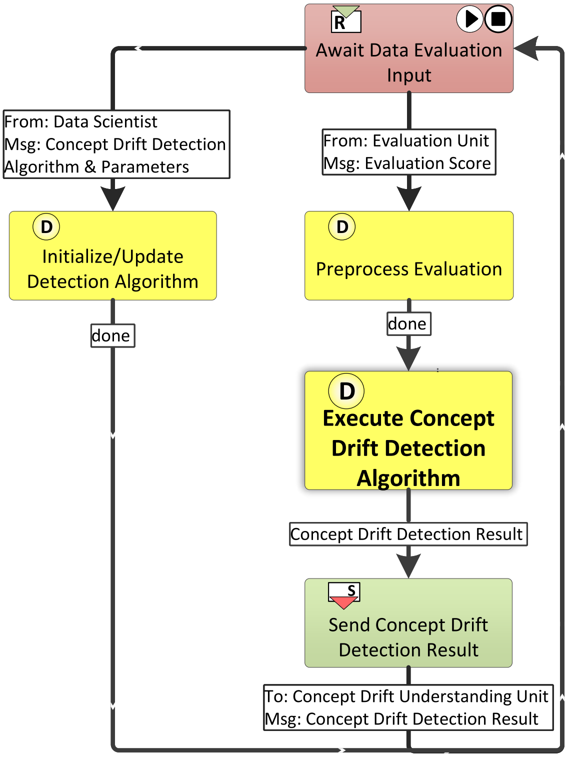

R
Await Data Evaluation Input
From: Data Scientist
Msg: Concept Drift Detection Algorithm & Parameters
From: Evaluation Unit
Msg: Evaluation Score
D
Initialize/Update Detection Algorithm
D
Preprocess Evaluation
done
done
D
Execute Concept Drift Detection Algorithm
Concept Drift Detection Result
S
Send Concept Drift Detection Result
To: Concept Drift Understanding Unit
Msg: Concept Drift Detection Result

Figure 5.8: The subject behavior diagram (SBD) of an inference process system's concept drift detection unit (Trat et al. 2025). This system component processes concept drift evidence and provides other system components with associated insights.

Concept drift adaptation unit. This unit is the entity receiving evidence from the concept drift detection and the understanding unit. It acts as an entity orchestrating associated responses. Its SBD is presented in Figure 5.10. The figure visualizes the invocation of this unit in different cases: Initially or occasionally, parametrization is conducted by the data scientist. In a regularly recurring fashion, outputs from other subjects of the maintenance sub-system is processed. Subsequently, this unit can trigger updates of the machine learning estimator's model.

Concept drift adaptation system data buffer. This entity accepts and provides data for various other system entities. As visualized in Figure 5.6, it is hence one of the most central subjects maintaining interactions with several others. It is a vital entity of robust inference process systems in practice, as it takes into consideration the delay of labels, i.e. ground truth, which is the case often observed in real-world scenarios. Being a buffer, this entity maintains a memory of sufficient size to store data. More precisely, this data comprises corresponding input data, estimation and label tuples. As soon as these tuples are processed and no longer required, they can be dropped

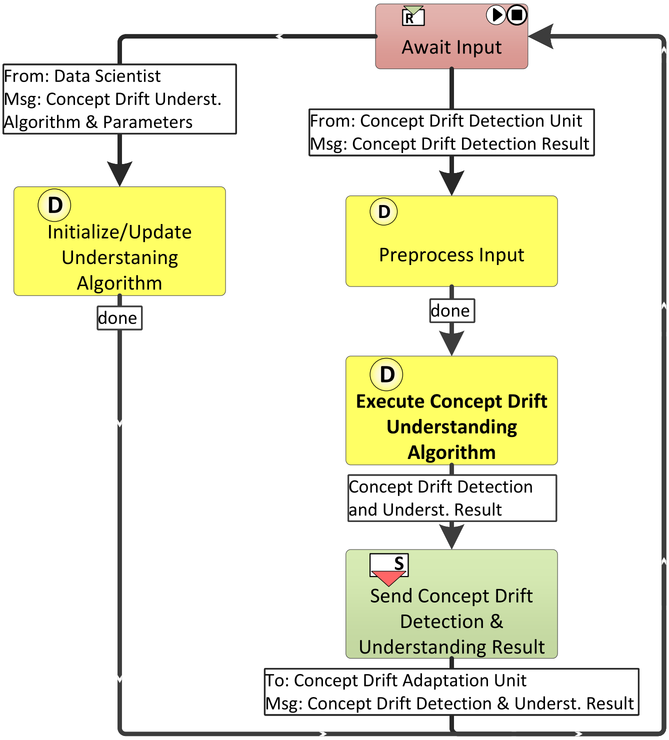

Await Input
From: Data Scientist
Msg: Concept Drift Underst. Algorithm & Parameters
From: Concept Drift Detection Unit
Msg: Concept Drift Detection Result
D
Initialize/Update Understaning Algorithm
done
D
Preprocess Input
done
D
Execute Concept Drift Understanding Algorithm
Concept Drift Detection and Underst. Result
S
Send Concept Drift Detection & Understanding Result
To: Concept Drift Adaptation Unit
Msg: Concept Drift Detection & Underst. Result

Figure 5.9: The subject behavior diagram (SBD) of an inference process system's concept drift understanding unit (Trat et al. 2025). Its inclusion in an inference process system can enable more guided concept-drift-adaptation-related actions as well as the provision of explanatory information.

to free up memory. As the Subject-Oriented behavior definition of a buffer would be trivial, i.e. merely accepting and providing data when requested, it is modeled as an interface subject.

Delayed label source. As indicated above, the delay of labels is often observed in real-world scenarios. The amount of the delay can vary greatly. In this regard, the case of entirely absent labels, i.e. such that will never be received, can be represented as the limiting case of infinite delay (Gomes et al. 2019). In the reference architecture, this abstract entity is represented by the subject delayed label source. For similar reasons as the streaming data source, this subject is modeled as interface subject.

Evaluation unit. This unit's primary function is to compute performance-related scores. Figure 5.11 presents its SBD. It is therefore provided with corresponding estimation and label pairs. Specific system implementations might do so in a batched fashion as soon as labels themselves also arrive in batches. The resulting scores are then sent to the concept drift detection unit. As

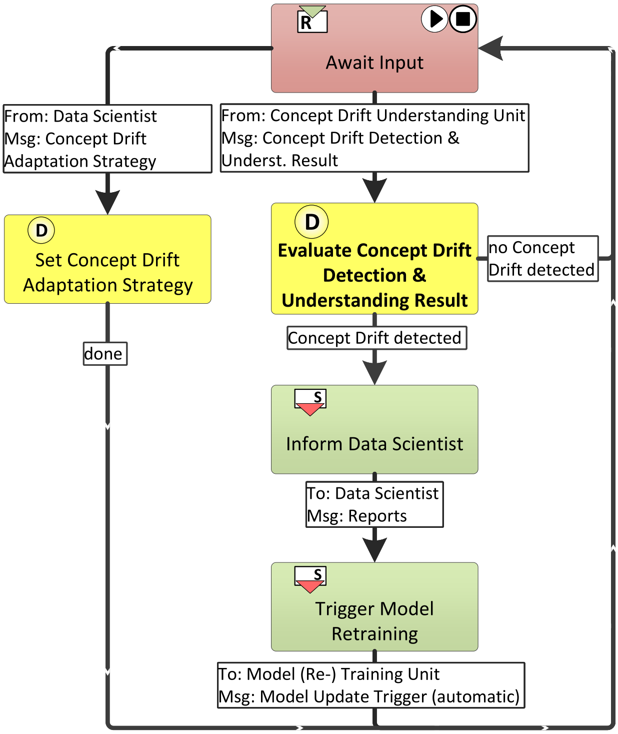


Figure 5.10: The subject behavior diagram (SBD) of an inference process system's concept drift adaptation unit (Trat et al. 2025). This subject represents a system component that evaluates concept-drift-related evidence and orchestrates associated robustness-sustaining actions.

outlined earlier, this type of input is required for supervised concept drift detection and adaptation.

Model (re-)training unit. The model (re-)training unit is, besides the concept drift adaptation system data buffer, represented by one of the most central and interconnected subjects. Its SBD is visualized in Figure 5.12. Initially, it is parametrized and optionally provided with historical data by responsible personnel such as the data scientist. It pools compute resources for estimator model updates and supplies the inference unit with the resulting sequential model versions, i.e. continuously trained updates of a model. For these training activities, it can request a range of labeled data samples from the data buffer.

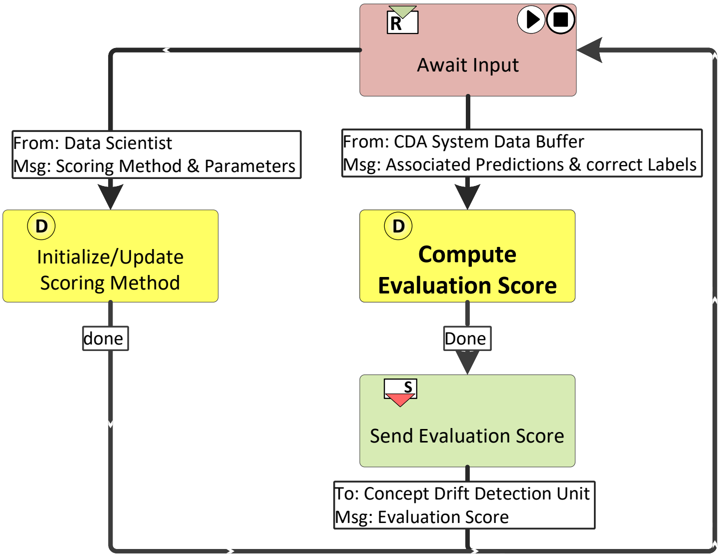

R
Await Input
From: Data Scientist
Msg: Scoring Method & Parameters
From: CDA System Data Buffer
Msg: Associated Predictions & correct Labels
D
Initialize/Update Scoring Method
D
Compute Evaluation Score
done
Done
S
Send Evaluation Score
To: Concept Drift Detection Unit
Msg: Evaluation Score

Figure 5.11: The subject behavior diagram (SBD) of an inference process system's evaluation unit (Trat et al. 2025). It computes performance scores based on corresponding estimation and label pairs and forwards the results to the concept drift detection unit.

5.5.3 Design Validation

To provide evidence to the previously raised research questions **RQ II-1** and **RQ II-2**, the proposed reference architecture is subject to validation. This is done continuously during the design process described earlier in this section and based on its final form. In this regard, the reference architecture can be considered final merely within the context of this dissertation. Fundamentally, the design process enables the execution of potentially infinitely many iteration cycles and associated incremental versions of the reference architecture.

Validation Approach

The Parallel Activity Specification Schema can be coupled with a syntactical validation algorithm that is introduced in Section 3.4. During the design process, this algorithm is applied at least once in each iterative cycle. Especially for extended iterative cycles, however, it can be applied more often. In such cases, for the proposed reference architecture, it is done every time an at least rudimentarily functional version of a newly added subject is available. As a result, errors as well as incoherent design choices can be identified early. This, in turn, can significantly increase the modeling quality. Analogously, the syntactical validation algorithm is executed based on the final version of the reference architecture.

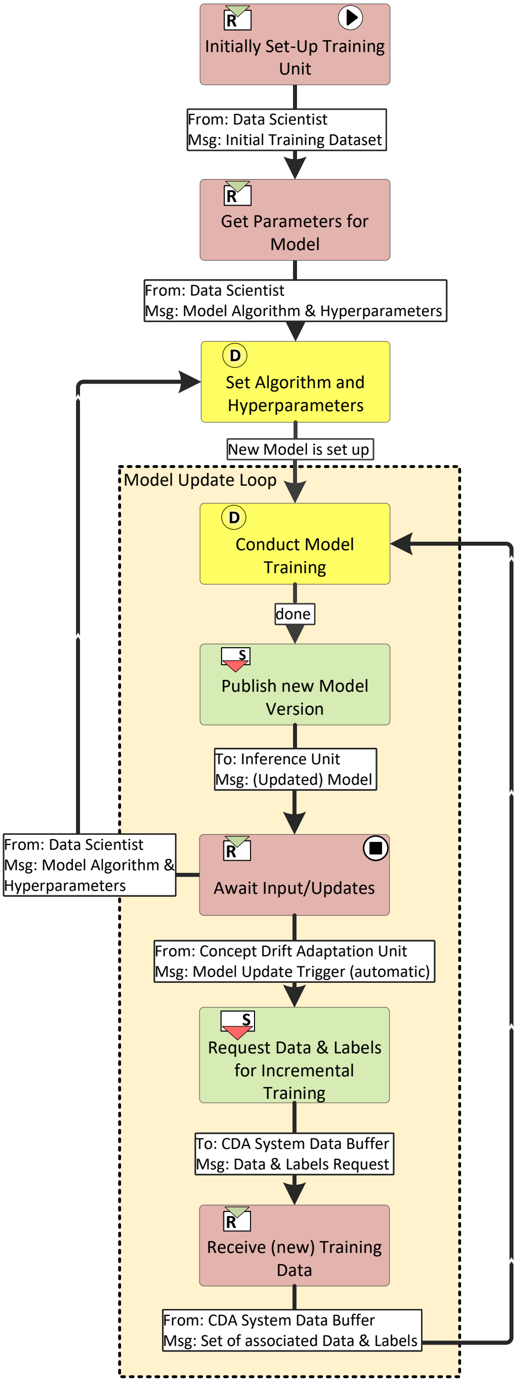

R
Initially Set-Up Training Unit
From: Data Scientist
Msg: Initial Training Dataset
R
Get Parameters for Model
From: Data Scientist
Msg: Model Algorithm & Hyperparameters
D
Set Algorithm and Hyperparameters
New Model is set up
Model Update Loop
D
Conduct Model Training
done
S
Publish new Model Version
To: Inference Unit
Msg: (Updated) Model
From: Data Scientist
Msg: Model Algorithm & Hyperparameters
R
Await Input/Updates
From: Concept Drift Adaptation Unit
Msg: Model Update Trigger (automatic)
S
Request Data & Labels for Incremental Training
To: CDA System Data Buffer
Msg: Data & Labels Request
R
Receive (new) Training Data
From: CDA System Data Buffer
Msg: Set of associated Data & Labels

Figure 5.12: The subject behavior diagram (SBD) of an inference process system's model (re-)training unit (Trat et al. 2025). Estimator models are retrained and versioned here.

Additionally, qualitative insights are gathered together with practical experts. This refers to findings emerging during jointly designing the reference architecture as well as when jointly examining its final version.

Validation Results

The results of this validation approach as well as further aspects that are notable within this context are discussed in the following.

Syntactical soundness. Applying the syntactical validation for the final version of the proposed reference architecture reveals no errors. It therefore is syntactically sound and able to model all behavior as well as interactions between all of its subjects.

Rigorous modeling framework with high capacity for capturing complexity. The qualitative validation reveals that strict Subject Orientation guidelines prove to be beneficial for sound system design. Using this method has the potential to guide the process of designing robust inference process systems. It forces one to specifically consider system-involved subjects and the exchange of necessary messages between them without having to refrain from employing clearly defined start and termination criteria of classical process descriptions (Trat et al. 2025). For instance, this facilitates the reference architecture to represent the cyclic character of many concept-drift-adaptation-related processes well. As illustrated by Figure 5.10, even complex interactions between different subjects are modeled while still allowing for precise but comprehensible cyclic and linear process sub-flows. Subject Orientation also rigorously requires designers to carefully annotate connectors, given that there is no option to just draw e.g. an unlabeled transition error. In relation to the results of the analysis of existing system descriptions outlined in Section 5.4.2, this would help avoid aspects contributing to confusion.
Additionally, Subject Orientation can unveil and eliminate ambiguities early, which realizes a high level of system modeling consistency. In large part, this can be attributed to the duality of this method: Modularization concerns are handled in the SID and process flow aspects are described in the different SBDs. This separation of concerns has the potential to promote focused design. In contrast, classical modeling paradigms typically require to include all aspects into one diagram, which can quickly lead to visual overload (Trat et al. 2025).

The two previously outlined insights demonstrate that Subject Orientation is fundamentally suitable for modeling robust inference process systems. Therefore, **RQ II-1** is answered affirmatively.

Comprehension and discussion aid. It can be argued that the proposed reference architecture has the potential to facilitate various stakeholders to comprehend complex intricacies of robust inference process systems. This manifests as the property to act as a visually accessible basis

enabling stakeholders to engage in highly informed and fruitful discussion about system-related aspects, such as implementation changes of critical system components. To illustrate this with an example, modeling a concept drift adaptation core (cf. CDA core in Figure 5.6) initially comprising three subjects is the result of such a discussion between the authors of Trat et al. (2025). It is a conscious choice jointly agreed upon based on practical knowledge about the tasks of a data scientist for ensuring estimator robustness. Interestingly, the discussion itself was sparked by the strict guidelines Subject Orientation imposes on the designers. Aspects associated to this specific finding were already further expanded upon above. Furthermore, the input of reviewers of the 58th Hawaii International Conference on System Sciences 2025 critically contributed in including a subject representing concept-drift-understanding-related approaches in the concept drift adaptation core. This expansion of the designer pool thus integrated further practical knowledge in the design process.
Similarly, this property can be made available throughout any design and usage stage of a robust inference process system. During validation, it was also observed to hold for stakeholders with different backgrounds as it fosters diverse questions, e.g. regarding comprehension, completeness and nomenclature. This, again, has the potential to improve the modeling quality of subsequent iteration cycles (Trat et al. 2025).

System implementation support. The proposed reference also has the potential to support implementation efforts. Especially the use of subjects to represent system components as well as their complex interactions enables the derivation of software architectures or code generation. This is indicated by the fact that this approach already encompasses the modularization aspect. This can ultimately help to accelerate the development of robust inference process systems as well as to foster component reusability across different system implementations.

Human-machine interfaces. Additional benefits with regard to human-machine interfaces can be expected from the reference architecture. The state of science of the field of concept drift adaptation critically lacks clear consideration of such (Trat et al. 2025). Being designed using Subject Orientation, the reference architecture facilitates to include the human as an integral role involved in system design and operation (Elstermann 2020, pp. 277–278). This is demonstrated by integrating the action field of data scientists. In specialized robust inference process systems, differentiated roles of the data scientists or further human actors, such as data or machine learning engineers, can self-evidently be integrated as well.

Based on these above-outlined insights, it can be argued that the suggested reference architecture indeed has wide range of practical advantages. Among others, it facilitates deep comprehension of system intricacies, associated discussion, system implementation, as well as including the human as an important factor. Carefully applied, it does so during and subsequent to the system design stage. Therefore, **RQ II-2** is answered affirmatively.

5.5.4 Practical Integration

In the following, concrete guidelines on how to apply the proposed reference architecture for further practical use cases are presented. In large parts, the foundation for these stem from practical insights gathered during the industry research project furnFUSION (Sönke and Trat 2025). This project involved activities to design robust inference process systems within the context of researching a data ecosystem for the industrial retail sector.

As strategic integration context, the product development principle front-loading is employed. This principle heavily relies on considering various forms of expertise from e.g. future users or domain experts early on in development activities. It is known to have various positive impacts on the development. Apart from changes being relatively cost-efficient in early development stages, it can also contribute to shorter development cycles and increased innovation capacity (Ovtcharova et al. 2015, pp. 118–119), (Thomke and Fujimoto 2000).
The implementation of front-loading in industrial practice is based on three requirement pillars (Ovtcharova 05.05.2025), as depicted in Figure 5.13:

1. Platform: Implementing front-loading requires a digital mock-up structure that permits the functional integration of already an incomplete subset of the product's components.
2. Development flexibility: Product development processes are required to be sufficiently parallelizable as well as adaptable to enable the early integration of expert knowledge.
3. Knowledge: Involved personnel resources are required to possess appropriate technological skills.

These pillars are transferred into the inference process system development context. In this regard, the reference architecture serves as platform, specifically derived from its template-based form for each use case. Details on this as well as the associated process and organizational context complementing data science process models are outlined in the following subsections. This way, an answer to **RQ II-3** is provided.

Template-based Application

The suggested reference architecture is an abstract template that is intended to be specialized as visually represented in Figure 5.14. This way, it can support the design process as well as implementation of robust inference process systems for a variety of use cases with their specific requirements. In line with the platform aspect of front-loading, it simultaneously serves as a digital mock-up of a validatable inference process system model. Parallelized development tracks of system components can converge in it and allow for an iterative system implementation

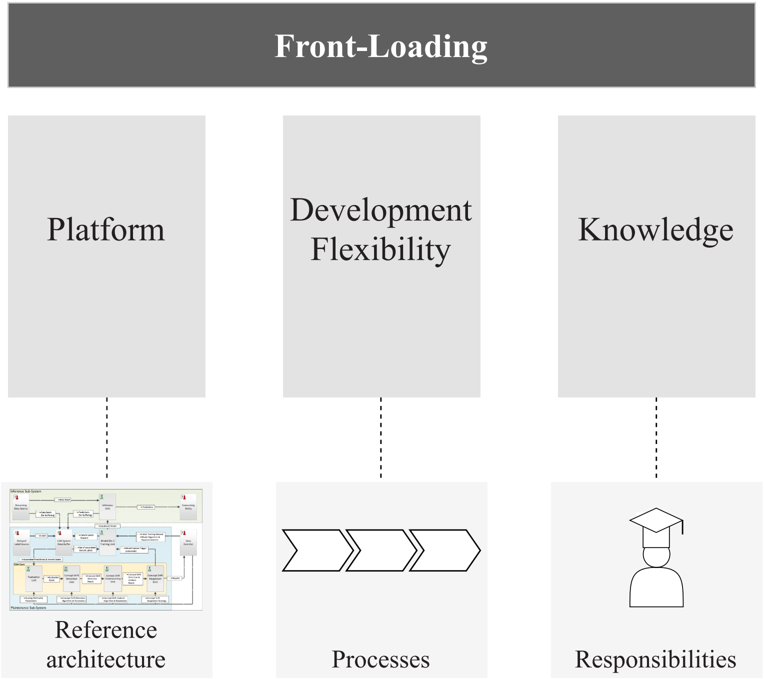


Figure 5.13: Integrating the proposed reference architecture via the product development principle front-loading. This principle's potential for increasing the efficiency of product development by making available expert knowledge early on is leveraged to guide the practical application of the reference architecture. For successful implementation, the three pillars of front-loading represent the requirements that need to be fulfilled in the industrial practical context.

and continuous testing. Also, it can help various stakeholders understand the intricacies of the specialized system as well as its requirements when integrating it within existing system landscapes.

In its form described in Section 5.5.2, the reference architecture covers use cases that require fully and semi-supervised explicit concept drift adaptation. Several presented subjects with their respective behavior definition are relevant across different kinds of concept drift adaptation approaches and can be reused. Optionally, they can be modified to fit use-case-specific constraints and design needs. For instance, in the case of unsupervised explicit concept drift adaptation approaches, the concept drift detection unit would directly receive input samples from a streaming data source (Gemaque et al. 2020) and not from an evaluation unit to probe for concept drift occurrences. Also, the often required presence of multiple subjects sharing the same role can straightforwardly be modeled as multi-subjects using Subject Orientation. This way, e.g. multiple streaming data sources or consuming entities could be represented.

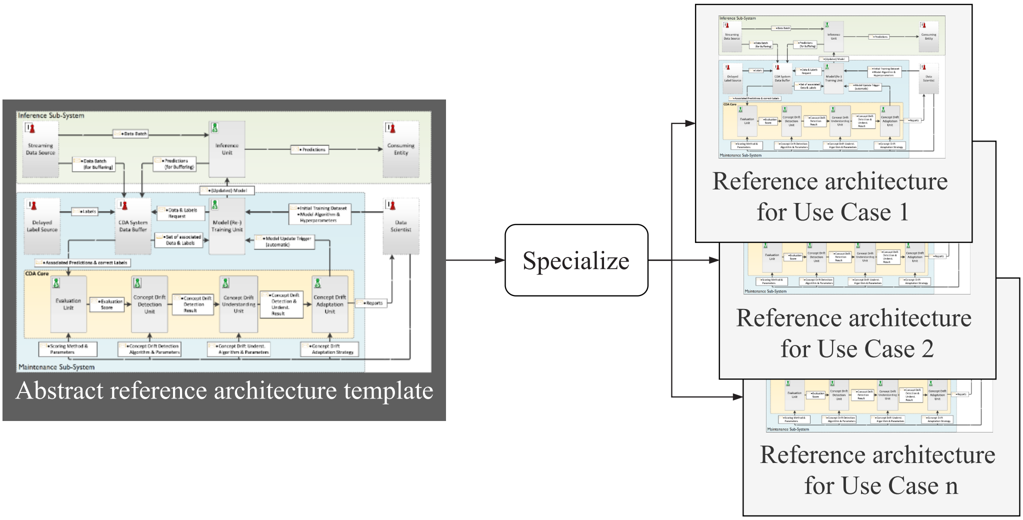

Inference Sub-System
Streaming Data Source
Data Batch
Data Batch (for buffering)
Predictions (for buffering)
Inference Unit
Predictions
Consuming Entity
(Updated) Model
Delayed Label Source
Labels
CDA System Data Buffer
Data & Labels Request
Model (Re-) Training Unit
Initial Training Dataset
Model Algorithm & Hyperparameters
Data Scientist
Set of associated Data & Labels
Model Update Trigger (automatic)
Associated Predictions & correct Labels
CDA Core
Evaluation Unit
Evaluation Score
Concept Drift Detection Unit
Concept Drift Detection Result
Concept Drift Understanding Unit
Concept Drift Detection & Underst. Result
Concept Drift Adaptation Unit
Reports
Scoring Method & Parameters
Concept Drift Detection Algorithm & Parameters
Concept Drift Underst. Algorithm & Parameters
Concept Drift Adaptation Strategy
Maintenance Sub-System
Abstract reference architecture template
Specialize
Reference architecture for Use Case 1
Reference architecture for Use Case 2
Reference architecture for Use Case n

Figure 5.14: Template-based practical application of the proposed reference architecture. By analyzing boundary requirements of a use case, the reference architecture can be specialized to describe an appropriate robust inference process system.

Process Context

As concluded in Section 5.4, data science process models hardly provide concrete architectural guidance for the implementation of inference process systems. This is even less pronounced in models that remotely take machine learning robustness into careful consideration. Given their process-centric orientation, however, one might consider this deficiency a reasonable one. Nevertheless, considering a model-centric perspective, in the form of the proposed reference architecture, during system development is shown to be beneficial above in this section. This benefit, furthermore, distinctly and deliberately includes operation-phase-related activities, which are only partially covered by data science process models. Namely, among others, performance degradation and maintenance-related ones are formally represented in the proposed reference architecture.

To illustrate, one might again consider the model CRISP-ML(Q) (cf. Figure 3.11). Integrating a model-centric perspective should not be initiated as late as deployment efforts of a satisfactorily trained estimator model is being tended to. Instead, this can already be initiated in parallel when "Model Engineering" starts, i.e. early on in the associated phase "Model Development". This is particularly evident in the fact that the estimator model choice has a pronounced impact on the choice of a suitable concept drift detection algorithm (Lima et al. 2022). This proposed parallelism of data science and robust system design concerns is further illustrated in Figure 5.15. In similar vein, the model PAISE can be integrated with a use-case-specialized reference architecture early on. In its case, this can already be initiated during early development cycles of the "AI component". One might conclude that adapting these processes as described above is possible, which is in line with the second pillar of front-loading implementation – development flexibility. Paired with continuous testing, it harbors the potential for reducing development effort and errors (Ovtcharova 05.05.2025).

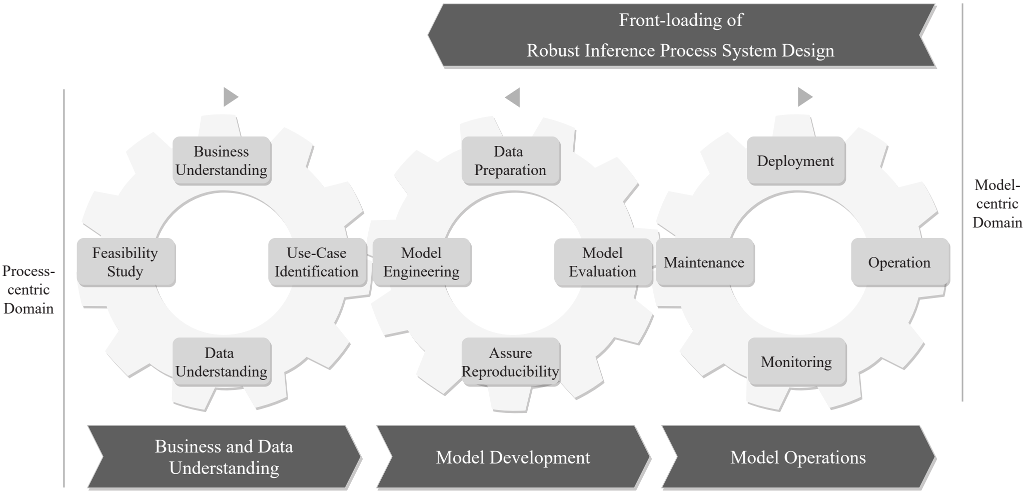


Figure 5.15: The Cross-Industry Standard Process Model for the Development of Machine Learning Applications with Quality Assurance Methodology (CRISP-ML(Q)) integrated with the model-centric perspective of the proposed reference architecture for robust inference process systems. Employing the latter is suggested to be front-loaded, i.e. to start already during model development, represented by the horizontal position of diagram elements. This way, system-design-related concerns are partially parallelized to two of the three major phases of CRISP-ML(Q), which are denoted at the bottom. Figure based on Studer et al. (2021).

Organizational Context

Among the data science process models compared in Section 5.4 (cf. Table 5.2), PAISE (cf. Figure 1.1) is the only representative mentioning personnel responsibilities. However, this is done in an abstract fashion primarily focused on design phases, while largely omitting the operation phase. For the latter, PAISE describes the hand-over of artifacts to merely an unspecified organizational unit or, in the case of a public client, an authority (Hasterok et al. 2021). There is an inherent paradox in this, with the operation phase potentially being the most laborious phase as it contains a plethora of pitfalls with respect to system maintenance (Sculley et al. 2015).

Apart from operating, designing inference process systems that scale in practical scenarios is also still a notoriously difficult challenge. Both require the involvement of different roles with different expert skills. Among others, these include monitoring engineers, data scientists, machine learning operations specialists, machine learning and data engineers, project managers, as well as also domain experts. Figure 5.16 attempts a clustering of these as well as associated roles based on to their respective responsibility domain within the context of inference process systems. Given that many of these more production-oriented disciplines are not yet part of data science education programmes, there is a critical shortage of sufficiently skilled human resources (Kreuzberger et al. 2023, Anaconda Inc. 2021). For these reasons, it can be argued that the explicit assignment of responsibilities is crucial for these phases. Trivially, in such a complex environment, clear responsibility definitions can reduce the potential for errors due to responsibility gaps. However, they also have the potential to, among other aspects, enhance accountability and deep technical understanding, improve operational efficiency, and reduce risks associated to e.g. human oversight. Given its efficacy in software-related domains, the responsibility assignments are often integrated within agile structures (Amershi et al. 2019).

The proposed reference architecture serves as a formal model that is shown to have the capacity to represent human roles in the context of robust inference process systems. Consequently, it can be employed to directly leverage this capacity to represent assigned responsibilities and required skills as posited by the knowledge aspect of front-loading. Simultaneously, it clearly defines the associated human-machine interfaces and interactions.

5.6 Summary

Typically, the development of machine learning estimators commences within a highly experimental setting. Once a satisfactorily generalizing estimator model is available, it is eventually deployed into the designated productive environment to serve the purpose it was initially conceptualized for. In the industrial practice, this is increasingly successful. However, it merely constitutes a first step towards its robust productive application. The software systems designed to operate such estimators in the long term widely lack measures to preserve their robustness against concept drift. To prevent catastrophic failure, their continuous monitoring and maintenance is crucial.

This chapter initially formalizes robustness as an estimator's continuously high estimation performance and capability to rapidly recover from concept drift occurrences. It also formally defines the infrastructural and software-based contexts operating estimators as inference process systems. Motivated by the idea to further the design of such systems, a systematic literature review is conducted. Initially considering 278 scientific publications, a data basis of modeling approaches for formally and visually describing inference process systems containing robustness-enhancing measures is gathered. A qualitative analysis of this data basis is then conducted to identify their

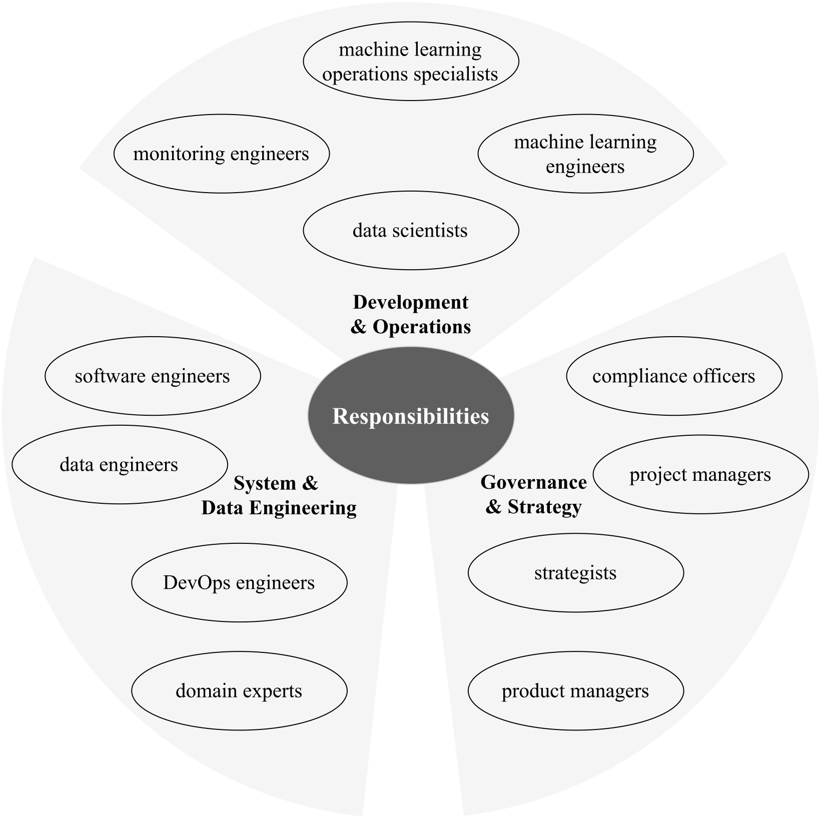


Figure 5.16: Layers of inference-process-system-related responsibilities is organizations. The layer of development & operations revolves around design and maintenance of systems as well as their machine-learning-estimator-based cores. System & data engineering and governance & strategy refer to responsibilities associated with enabling technical system foundations and technical or personnel-related decision-making, respectively.

shortcomings: A widespread lack of consistency, clarity and informative content that critically limits these descriptive models' capability to capture the high degree of complexity of concept drift adaptation approaches.

Furthermore, by reviewing the state of the art of data science process models, this chapter finds that these lack emphasis on process phases that are most critical for ensuring the robustness of inference process systems. Incorporating robustness-enhancing measures during their design is also hardly supported by such process models. Together, these insights provide answers sought by **RQ I**.

Next, this chapter's research question **RQ II** draws the reader's attention to a suggested model-centric solution to the afore-described research gaps and shortcomings that specifically supports robust inference process system design: A reference architecture created through a carefully defined Subject-Oriented approach, utilizing the results of a component-oriented analysis of the data basis from the above-mentioned systematic literature review. The resulting reference architecture is provided in Figure 5.6 and further figures throughout Section 5.5.2 detailing its components. Additionally, concrete guidelines for its front-loaded integration in practical industrial contexts, well complementing the process-centric nature of data science process models, are elaborated on.

By formalizing robustness and inference process systems, this chapter provides a well-founded groundwork used throughout this thesis to conceptualize and evaluate further aspects. The proposed reference architecture provides the framework to formulate a stream-applicable method in Chapter 7 in a careful way, which is due to the rigorous modeling framework of Subject Orientation. The stream-applicable method is then employed in Chapter 8 for a range of real-world industry scenarios.

The research presented in this chapter shall further enable the scientific community to derive and describe specialized architectures for robust inference process systems for a wide range of industrial use cases. It shall also encourage a transition from conventional free-form flowcharts. The underlying rationale for this is that, despite being popularly applied, such are largely inappropriate to sufficiently capture the complexity of concept drift adaptation approaches. The open source publication of the reference architecture is intended to facilitate associated activities.

6 Concept Drift Detection Ensembles

This chapter explores concept drift detection ensembles (CDDEs) as one of the central aspects of this thesis, which are formed by combining multiple concept drift detectors. Doing so unlocks a set of potentials by, among other aspects, leveraging their diverse strengths. Especially given the wide range of different concept drift types occurring in real-world applications, ensembling enables the combination of detectors with distinct concept drift type specializations. This way, the aforementioned range can potentially be better covered. As a result, this can help increase the concept drift detection performance, which in turn supports effective concept drift adaptation.

Within the broader context of machine learning, ensembling introduces a set of already quite well understood potential benefits, as outlined in Section 3.1.2. Figure 6.1 additionally visualizes its underlying motivation. However, it is important to make a strict distinction between machine learning estimator and concept drift detector ensembles. One of the main reasons for this is that the nature of the respective member algorithms is profoundly different. Also, it is noteworthy that, similarly to machine learning estimator ensembles (Polikar 2012, p. 1), CDDEs have been under-researched for a long time despite the fundamentals being accessible. This thesis aims at addressing this gap.

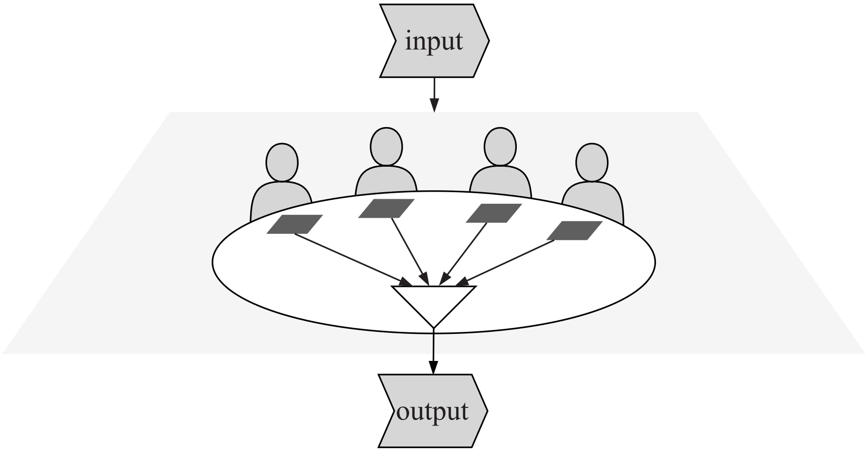


Figure 6.1: Leveraging the potential of diverse expertise. Machine learning ensembles are comparable to committees of experts. Diverse statements from individual entities can thus be aggregated into a more reliable single one.

The research questions addressed in this chapter are re-stated in Section 6.1. To provide a solid foundation for the research on CDDEs, a review on the state of science on the associated field is conducted. This is done by following a systematic strategy probing 220 publications. Further details are outlined in Section 6.2. From this review, several insights into the research on CDDEs regarding the temporal progression, notable contributors as well as research trends are extracted. The results of an analysis of these aspects is provided in Section 6.3. Subsequently, this chapter places a distinct emphasis on describing key characteristics and principles of designing CDDEs in Section 6.4. The associated findings are gained from the qualitative analysis of the publications gathered within the review and contextualized with existing machine learning knowledge. Finally, the result is condensed into a structured design blueprint for CDDEs.

This chapter's Sections 6.3 and 6.4 partly contain verbatim passages from a work previously published by the author of this thesis (Trat and Ovtcharova 2023). These passages are not individually cited when the author is responsible for the intellectual content and research insights. Contributions from co-authors as well as figures and tables are explicitly cited.

6.1 Research Questions

The scientific approach of this chapter is governed by the following research questions:

RQ III How extensively are concept drift detection ensembles studied?

RQ III-1 How can concept drift detection ensembles be formally characterized?

RQ III-2 How did the research field progress and in which maturity stage is it?

RQ III-3 What are the most notable contributions on the field?

In other words, **RQ III** is designed to work out the current state of the research on CDDEs. This question can therefore be further operationalized by formulating several sub-questions: While **RQ III-1** demands a definition attempt, **RQ III-2** and **RQ III-3** pursue the continuous development of the research field and contributors as well as pivotal findings, respectively. Associated answers are provided throughout Section 6.3.

RQ IV How can one design concept drift detection ensembles?

RQ IV-1 Which aspects for concept drift detection ensembles design exist?

RQ IV-2 How can design considerations increase diversity within concept drift detection ensembles?

RQ IV-3 Which interdependencies exist between different design aspects?

RQ IV explores best practices for designing CDDEs themselves, which is approached by the following sub-questions: **RQ IV-1** demands the exploration as well as enumeration of feasible CDDE components and configurations. These are scrutinized regarding their capability to introduce ensemble diversity and possible interdependencies by **RQ IV-2** and **RQ IV-3**, respectively. Section 6.4 offers evidence to respond to these questions.

6.2 Systematic Literature Review

This section outlines how the three-step systematic literature review process introduced in Section 2.1 is applied to obtain insights into research on CDDEs. It involves gathering potentially relevant publications, filtering out irrelevant ones and analyzing the remaining ones based on the pre-defined research questions. The flowchart in Figure 6.2 specifies the abstract one in Figure 2.1 for application of the process in this instance.

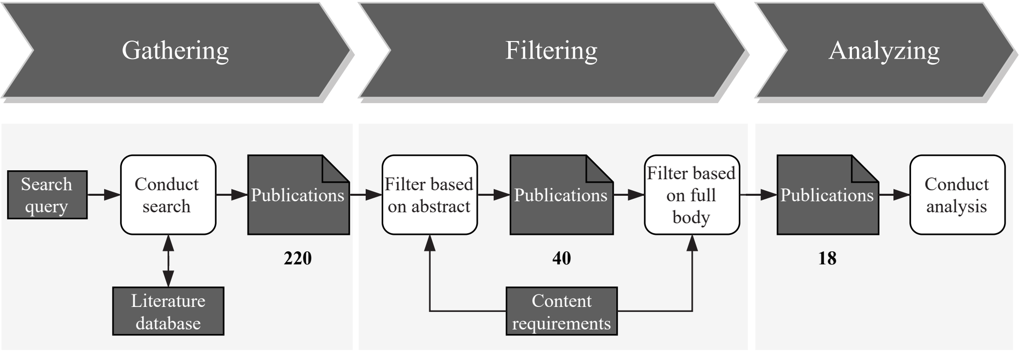


Figure 6.2: Flowchart representation of the systematic literature review applied to obtain insights into research on concept drift detection ensembles. It involves gathering, filtering and analyzing publications. Being a specification of Figure 2.1, details of filtering-related sub-steps are named in the respective flowchart elements and quantities of retrieved, filtered and analyzed publications are provided as bold-formatted numbers.

Gathering

The initial step of gathering publications involves using the curated abstract and citation database Scopus[9]. It indexes a large set of databases belonging to publishers such as ACM, Springer and IEEE. This step was finalized on January 10th 2023, hence considers literature published until that date.

The query string employed for searching via Scopus is designed to retrieve potentially relevant publications based on their title, abstract and keywords. It widens the search space by allowing to omit the word *concept* when referring to *concept drift*. Furthermore, it requires common detection-related (e.g. *identification*) and ensemble-related terms (e.g. *boosting*).
A query designed this way returns 790 publications. However, based on a preliminary screening, the relative number of relevant publications among the returned ones is found to be low. To improve the fitness of the query and to reduce the number of returned publications to an assessable volume, it is further developed. With this being the first review of CDDEs, it shall be avoided to limit the publication time horizon to facilitate a comprehensive analysis of the research field's historical evolution. For this reason, word proximity requirements are employed. These require the detection-related and ensemble-related terms to appear in the same sentence with the term *drift*. This is implemented by following the recommendation of Scopus to set the distance between two query terms to 15 words (Elsevier 2023). The resulting final query string is shown in Table 6.1. Performing the search according to this strategy produces 220 results.

Table 6.1: Details on gathering publications on concept drift detection ensembles. A specifically designed search query is utilized with the meta research database Scopus.

Literature database	Scopus

String	`TITLE-ABS-KEY(drift W/15 (detect* OR identif*))`
	`AND TITLE-ABS-KEY((detect* OR identif*)`
	`W/15 (ensemble OR boosting OR bagging OR stacking))`

Conditions	none

[9] Available at `https://www.scopus.com/search/form.uri`, accessed: 14.02.2024.

Filtering

The gathered publications undergo a careful two-step screening. A publication is considered relevant, if it fulfills the following content requirements:

- It suggests a specific CDDE approach,
- *or* analyzes the use of one or more such.
- It provides a clear and detailed documentation of the CDDE design.
- It underwent a sound peer-reviewing process.
- It is published in English language.

If it does not meet these content requirements, it is filtered out.

Firstly, the 220 publications returned from the search are filtered based on their abstract. This is complicated by the circumstance that concept drift detection and purely adaptation-related terms (e.g. *handling* or *reacting*) are often used interchangeably in retrieved publications. As concept drift detection pools methods for explicitly identifying concept drift occurrences and concept drift adaptation such for updating predictive models as a reaction to concept drifts, they are, however, two terms representing crucially different aspects (Gama et al. 2014). After abstract-based filtering, the number of potentially relevant publications is reduced to 40.
Secondly, the full body of these 40 publications is then read. Applying the content requirements reduces the number of publications to 18. This remaining set of relevant publications is subject to the analysis. A comprehensive list of these 18 publications is provided in Section 6.4.

As mentioned above, a word proximity limitation is applied. It cannot be precluded that this limitation creates a latent set of unidentified but potentially relevant publications. However, as none of the related-work sections of retrieved publications reveals a previously unidentified one, the search method is not modified.

Analysis

The 18 publications are then subject to a historical and bibliometric as well as a qualitative literature analysis. Within this context, the temporal progression of CDDE research is described and trends are identified. Measured based on bibliometric information, notable contributions as well as their respective authors are highlighted. For the qualitative content analysis, a set of content attributes are defined and populated for each publication. Table A.2 in the appendix of this thesis presents these attributes.

6.3 Evolution of Concept Drift Detection Ensembles

This section presents the results of the conducted literature analysis. Initially, the obtained knowledge is condensed into an attempt to formally define CDDEs as well as taxonomize them. While presenting the 18 gathered publications, it outlines the temporal progression of research and characterizes it based on various aspects. Subsequently, patterns and sources of notable contributions as well as emerging research trends are qualitatively discussed. Also, the research groups and individuals who have made significant contributions to the field are introduced.

6.3.1 A Definition of Concept Drift Detection Ensembles

The existing literature does not provide a complete definition of CDDEs. For this reason and to directly respond to research question **RQ III-1**, an attempt to formally characterize them shall be made. For this purpose, gained insights into retrieved publications on CDDE conception and application as well as core principles on machine learning estimator ensembles (cf. Section 3.1.2) are compiled, resulting in the following definition:

> **Definition 6** (Concept drift detection ensemble)**.** A concept drift detection ensemble is an algorithmic ensemble construct E^{det} combining n^{det} base concept drift detectors D_i as members differing in algorithm f^{det} or parametrization θ, with $i \in \{1...n^{det}\}$. As members, the construct can alternatively combine sub-ensembles E_i^{det} with the same basic properties.
> The construct employs a distribution strategy S_{dist}^{det} expressed as a function $S_{dist}^{det}(\theta_{dist}^{det})$ and an aggregation strategy S_{agg}^{det} expressed as a function $S_{agg}^{det}(\theta_{agg}^{det})$, with θ_{dist}^{det} and θ_{agg}^{det} respectively denoting their parametrization. They define the allocation of concept drift evidence input among the D_i and the consolidation of their individual alert outputs into an ensemble-wide output, respectively.

On the one hand, Definition 6 outlines the major components of CDDEs, i.e. detectors and strategies for processing data input and alert output. Regarding detectors D_i, the definition employs the term *base* to refer to them being singular non-ensemble algorithmic models. At the same time, this property is extended to also account for a hierarchical ensemble structure with sub-ensembles E_i^{det} as members of E^{det}. On the other hand, this definition differentiates CDDEs from machine learning estimator ensembles by not considering machine learning models as members. It also does not explicitly require the use of performance-optimizing measures, which are, among other aspects, subject of Section 6.4. These measures contain but are not limited to member selection and variety as well as continuous adaptation strategies. An abstract schematic representation of a CDDE is additionally provided in Figure 6.3.

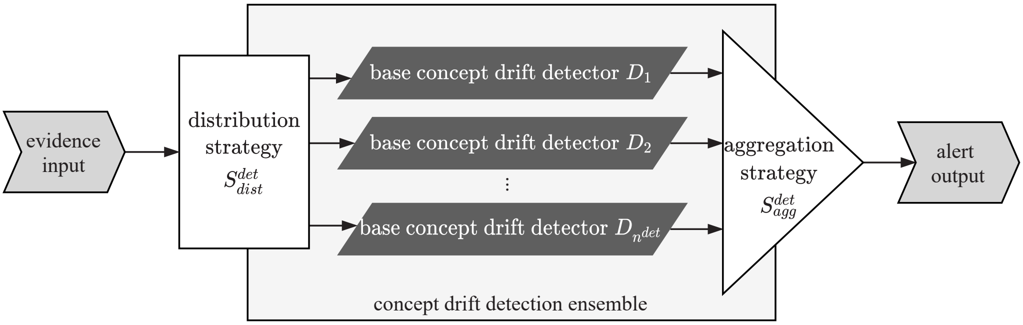

evidence input
distribution strategy S_{dist}^{det}
base concept drift detector D_1
base concept drift detector D_2
⋮
base concept drift detector $D_{n^{det}}$
aggregation strategy S_{agg}^{det}
alert output
concept drift detection ensemble

Figure 6.3: An abstract schematic visualization of a concept drift detection ensemble. It combines multiple base concept drift detectors D_i as members and allocates data to them based on distribution strategy S_{dist}^{det}. The alert outputs of members are consolidated into an ensemble-wide output based on an aggregation strategy S_{agg}^{det}.

6.3.2 A Taxonomy for Concept Drift Detection Ensembles

The literature on the broader field of concept drift detection already suggests associated taxonomies. They are e.g. based on the requirements of detectors regarding the supervisedness context of the data stream (Khamassi et al. 2018) or the way they process unlabeled stream data directly (Gemaque et al. 2020). However, their suitability to taxonomize CDDEs is limited as they would too strictly focus on individual member characteristics and disregard a wide range of ensemble characteristics. As it significantly impacted the evolution of CDDEs with regard to the temporal research progression as well as the emergence of research trends, one of their central differentiation criteria needs to be introduced initially. This additionally addresses **RQ III-1** as a taxonomy supports the formalization of CDDEs.

Based on the analysis conducted for this thesis, it can be argued that a significant impact on CDDE design arises from the member algorithm variety, i.e. the variety of employed base concept drift detector algorithms. While this section is centered around historical research intricacies, the associated design implications of this aspect are detailed in Section 6.4. Nevertheless, this variety aspect plays a major role within the context of the research progression as it considers the characteristics of entire member sets. Therefore, the 18 publications identified as relevant during the literature review are grouped for analysis based on whether they consider heterogeneous or homogeneous CDDE architectures. Heterogeneous ones combine multiple different base concept drift detector algorithms f_i^{det}, each parametrizable by a specific set of parameters θ_i. This is visualized in Figure 6.4 by featuring differently colored rectangles denoted as f_i^{det} in the detection algorithm column. In contrast, homogeneous architectures combine multiple instances of one specific base concept drift detector algorithm, visualized by their common equally colored rectangle in the detection algorithm column. Their variety is induced by having different parametrizations represented by differently colored rectangles denoted as θ_i in the parametrization column. Note that Figure 6.4 omits the parametrization column for the heterogeneous CDDE, as parameters are intrinsically bound to the respectively selected algorithm while not being a primary variety-inducing aspect. Table 6.2 lists all relevant publications while simultaneously applying this grouping criterion.

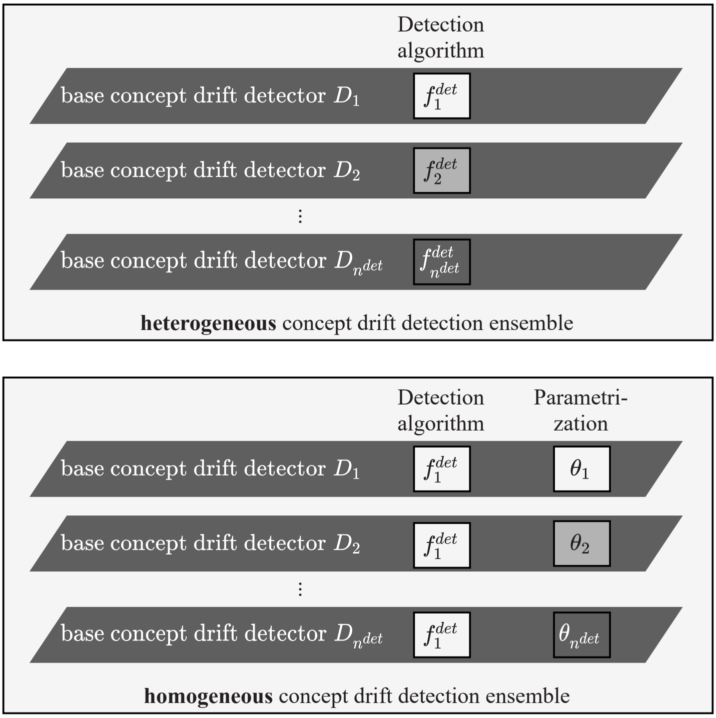

Detection algorithm
base concept drift detector D_1
f_1^{det}
base concept drift detector D_2
f_2^{det}
⋮
base concept drift detector $D_{n^{det}}$
$f_{n^{det}}^{det}$
heterogeneous concept drift detection ensemble
Detection algorithm
Parametri-zation
base concept drift detector D_1
f_1^{det}
θ_1
base concept drift detector D_2
f_1^{det}
θ_2
⋮
base concept drift detector $D_{n^{det}}$
f_1^{det}
$\theta_{n^{det}}$
homogeneous concept drift detection ensemble

Figure 6.4: Concept drift detection ensemble member algorithm variety. While heterogeneous concept drift detection ensembles vary in terms of the algorithm f_i^{det} of their detector members D_i, homogeneous ones do not. However, homogeneous ones would vary in terms of each member algorithm's parametrization θ_i.

Table 6.2: Overview of concept drift detection ensemble publications identified as relevant during the structured literature review of this thesis (based on Trat and Ovtcharova (2023)). The respectively suggested concept drift detection ensemble approach is grouped based on the algorithm variety of the base concept drift detectors it combines.

Member algorithm variety	Publications
heterogeneous ensembles	Sobolewski and Woźniak (2013)
	Maciel et al. (2015)
	Toumi et al. (2020)
	Du et al. (2015)
	Hu et al. (2018)
	Hu and Kantardzic (2022)
	Woźniak et al. (2016a)
	Zhang et al. (2020)
	Perez et al. (2020)
	Xu and Klabjan (2021)
homogeneous ensembles	Woźniak et al. (2016b)
	Bu et al. (2016)
	Pesaranghader et al. (2018)
	Lapinski et al. (2018)
	Korycki and Krawczyk (2019)
	Komorniczak et al. (2022)
	Okawa and Kobayashi (2021)
	Nguyen et al. (2022)

6.3.3 Temporal Progression

As gathered from the structured literature review, a set of 18 publications constitute the body of research on CDDEs. The following paragraphs provide insights into the temporal progression of the field.

Bibliometrics

Initially, the research field was established by the publication of Sobolewski and Woźniak (2013) on heterogeneous CDDEs. Until now, as can be taken from Table 6.2, 10 publications continued this research branch by focusing on this architecture type. Starting in 2016, homogeneous architectures are focused on by 8 publications. Regardless of architecture type, the number of CDDE publications ranges between zero and three per year during the analyzed time horizon. Figure 6.5 provides a visualization of the cumulated number of publications as well as citation insights. The bibliometric analysis reveals that contributors seldomly referenced previous CDDE works in their own. This fact is represented in the figure by the cumulated number of first-time references, i.e. the number that is increased each time an existing publication is initially referred to in a later publication. One can notice that at the end of the analyzed time horizon, only 9 publications in total are mentioned in other ones. This observation does not necessarily point towards a detrimental circumstance. Conceivable explanations considering the given maturity stage are, for instance, the exploratory character of a research that only begins to discover CDDEs and the lack of condensed and structured knowledge on the field in its entirety. Based on these findings, one can argue that the research can be characterized as being in an early stage, which is an aspect targeted by **RQ III-2**.

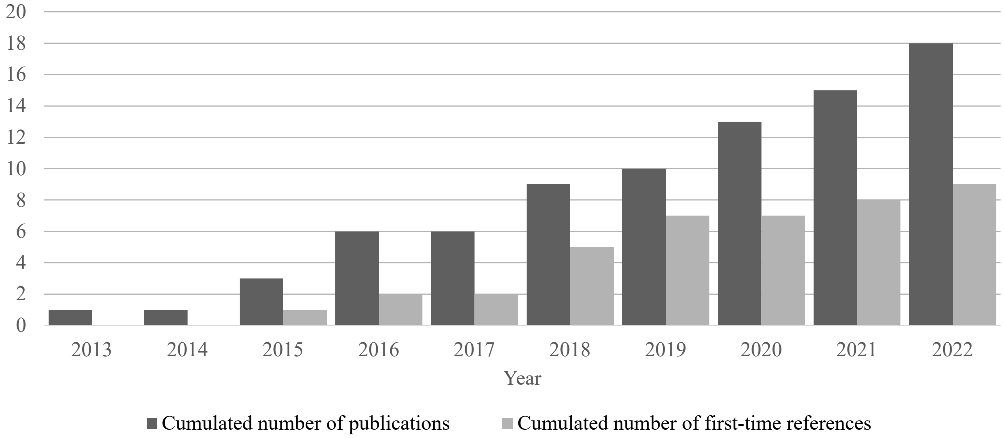


Figure 6.5: History of publications on concept drift detection ensembles. Until the time of writing, the number of publications grew to 18. The cumulated number of first-time references reveals that only 9 publications in total are mentioned in other ones so far.

Major Qualitative Characteristics

The relevant publications on CDDEs are predominantly driven by the motivation to employ this algorithmic ensemble construct as means to improve concept drift detection performance. This is comprehensible given the straightforward diversity potential introduced by ensemble methods. Only few publications consider the potential of CDDEs to improve adaptation performance (Lapinski et al. 2018, Toumi et al. 2020). Several other publications still apply simple concept drift adaptation approaches but do not delve into specifics (Sobolewski and Woźniak 2013, Maciel et al. 2015, Du et al. 2015, Pesaranghader et al. 2018, Korycki and Krawczyk 2019, Zhang et al. 2020, Perez et al. 2020, Xu and Klabjan 2021). Instead, it is utilized as an indirect measure of detection performance in the absence of concept drift labels obstructing the direct computation of detection-specific evaluation scores. This circumstance remains static over the analyzed time horizon.

Not surprisingly, the prevailing research gap of machine learning regression extends towards the common applications of CDDEs. Strikingly however, the body of research exclusively considers classification problems, with no involvement of any regression problem scenario. Also, evaluation using synthetic or often-employed public benchmark datasets is prevalent in relation to using data from real-world case studies (Toumi et al. 2020). This raises concerns regarding the applicability of these methods to potentially more complex and dynamic environments. Nevertheless, the supervisedness of the data stream scenarios, i.e. supervised, semi-supervised or unsupervised, is well balanced among the publications throughout the years.

In terms of insights into the fitness of CDDEs, their at least highly positive evaluation remains consistent across the analyzed time horizon. Some publications even report their exceptionally superior performance over singular concept drift detectors.

In conclusion, the field is still in its early stages of maturity, with significant research gaps persisting. This concerns areas such as the quantitative exploration of the space of feasible design options of CDDEs, employing them in real-world scenarios, their evaluation as well as a more profound generalized understanding of them. Further details can be obtained from the publication of Trat and Ovtcharova (2023). These qualitative insights also address **RQ III-2** by outlining characteristics of CDDE research throughout the years as well as its current stage.

6.3.4 Notable Contributions and Research Trends

The majority of publications on CDDEs exclusively consider related research on individual base concept drift detectors as scientific groundwork for their approach. As outlined before, previous works on CDDEs is rarely explicitly referred to. One might argue that future CDDE research could benefit from empirically benchmarking existing approaches. This could help to more sustainably progress the field with increasingly more promising insights being reported. Nevertheless, one can observe how certain publications do have an impact on subsequent ones and establish trends. Several instances of distinctly observable trends are described in the following, supported by Figure 6.6. In this figure, the 18 relevant publications, represented as rounded rectangles, are vertically arranged according to their publication year. Identified trends are illustrated by directed edges between publications. Publications not connected to other ones can be interpreted as predominantly introducing new ideas to progress the field of CDDEs. Based on these findings highlighting influential contributions to the research on CDDEs, **RQ III-3** is addressed.

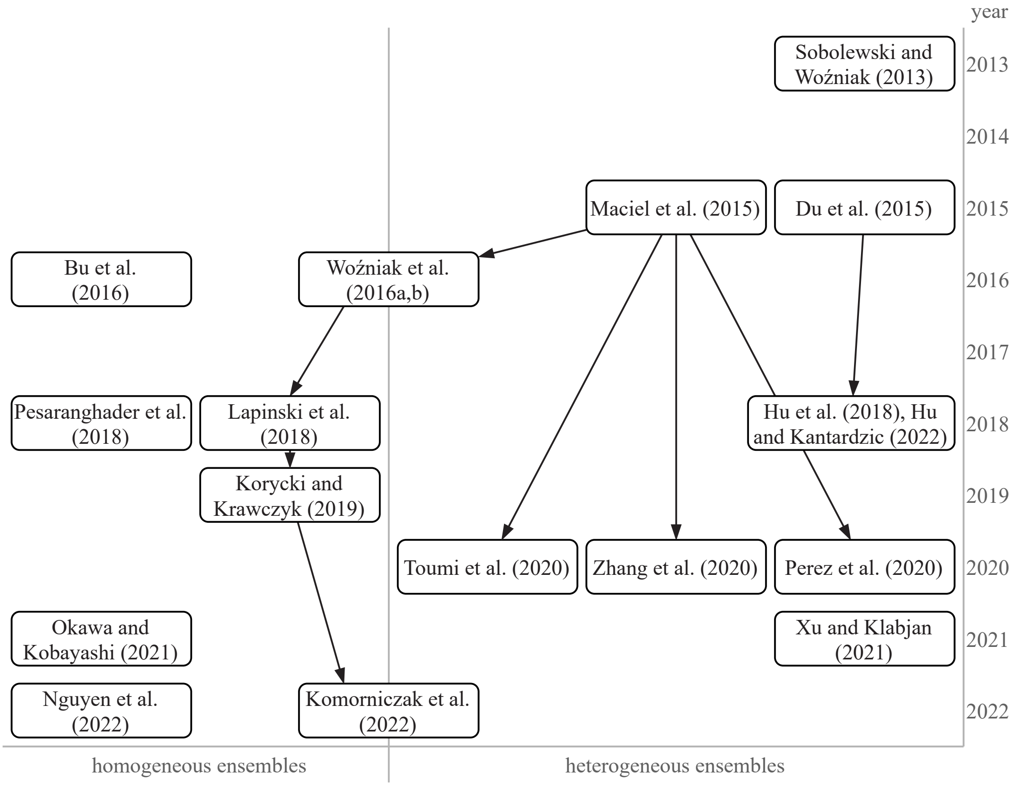


Figure 6.6: Temporal progression of research on concept drift detection ensembles by architecture type (based on Trat and Ovtcharova (2023)). Publications are represented by rounded rectangles containing the respective reference information. Their vertical position indicates the publication year, which is resolved on the vertical axis. A focus on homogeneous or heterogeneous ensemble architectures or both of each work is represented by its horizontal alignment. Directed edges represent distinctly observable research trends that are continued across multiple publications.

The onset of homogeneous concept drift detection. Woźniak et al. (2016a) focus on heterogeneous CDDE architectures. In a subsequent work (Woźniak et al. 2016b), they include the idea of homogeneous ones. Both publications are pooled in one rectangle as their contributions share a common core. While Bu et al. (2016) addresses the same idea, members of the extended research group around Woźniak et al. (2016b)[10] continued this approach: Homogeneous architectures

[10] `https://kssk.pwr.edu.pl/`, accessed: 14.02.2024

are the target of further research by Lapinski et al. (2018) and Komorniczak et al. (2022), and one might argue that it was an avenue strategically pursued by a research group. As the work of Komorniczak et al. (2022) contains a combination of both heterogeneous and homogeneous architecture aspects it is oriented on the border between homogeneous and heterogeneous CDDE research in Figure 6.6. Their contribution, however, emphasizes the former and is therefore listed accordingly in Table 6.2.

Concept drift detection by voting. In their highly influential work, Maciel et al. (2015) suggest combining different base concept drift detectors and aggregate their detections based on a voting mechanism. This is later applied by Toumi et al. (2020) and extended by Zhang et al. (2020) and Perez et al. (2020) by additionally weighing as well as aggregating concept drift alert levels. Also, Maciel et al. (2015) inspire Zhang et al. (2020) to also relax the timing constraints of individual base concept drift detector's alerts. This is done by considering multiple alerts falling into a specific temporal range. All alerts falling into this range are then accumulated to jointly trigger one ensemble-wide concept drift alert. Without this, all alerts would be required to be emitted simultaneously.

Base concept drift detector diversity. Du et al. (2015) originally motivates to promote diversity in CDDEs. Individual specializations and strengths of different base concept drift detectors shall be considered and redundancies among members shall be avoided. The works of Hu et al. (2018), Hu and Kantardzic (2022) continue this idea in their approach. Additionally, they include more different concept drift types to be covered.

Integrating distributed local concept drift detection. Lapinski et al. (2018) even pick up on ideas on further ideas of Woźniak et al. (2016b) than outlined above. This additionally reinforces the assumption of CDDE research being a strategic aspect of the surrounding research group. They also employ a homogeneous CDDE architecture and begin to consider local concept drift detection on the feature level. Subsequently, Korycki and Krawczyk (2019) and Komorniczak et al. (2022) further develop this approach by combining multiple features to subspaces. These are then providing the input data for associated concept drift detectors.

6.4 Principles for the Design of Concept Drift Detection Ensembles

A review of the literature on CDDEs reveals a notable research gap: The lack of a comprehensive reflection on design principles for CDDEs. None of the 18 publications identified as relevant from the structured literature review addresses this gap. However, these publications offer valuable contributions that inform the approach employed by this thesis to address this gap. The following insights therefore build on the results of the qualitative analysis of these publications, from which aspects for the design of CDDEs are extracted.
This section discusses these aspects as a state of the art of CDDE design. Simultaneously, it further augments the state of the art by general ensemble-related knowledge contributed by Krawczyk et al. (2017), Witten et al. (2011) and Polikar (2012). As result, the space of only partially explored CDDE design aspects is extended to include more feasible ones. Furthermore, by discussing as well as condensing the gathered insights, this section aims at harnessing these design aspects as principles for future CDDE design. A central artifact visually pooling all these insights is provided in Figure 6.12. It abstractly illustrates the schematic depiction of a CDDE within the context of its application in data stream environments.
Based on the findings provided in this section as well as this artifact, research question **RQ IV-1** is addressed. It shall also be remarked that this research is the primary matter of a work (Trat and Ovtcharova 2023) separately published by the author of this thesis.

Moreover, as found during the analysis, the member algorithm variety, i.e. choosing between heterogeneous and homogeneous architectures, has considerable impact on other aspects. Instances of this finding are additionally highlighted at several points throughout this section. Each of these points therefore contributes to responding to **RQ IV-3**.

Table 6.3 characterizes each of the 18 publications identified as relevant during the literature review regarding their choice of major design aspects. This way, it augments Table 6.2. The following subsections are structured based on these major design aspects. Apart from theoretically discussing these aspects, the subsections provide practical implications on design implementations.

6.4.1 Base Concept Drift Detectors

A CDDE is composed of multiple base concept drift detectors D_i that could alternatively be applied individually instead of as members of an ensemble. The exact set of detectors in an

Table 6.3: Qualitative characterization of concept drift detection ensemble publications with respect to major design aspects (based on Trat and Ovtcharova (2023)): The heterogeneity of supervised (sup) and unsupervised (unsup) base concept drift (CD) detectors, the type of employed aggregation being majority vote (MV), threshold vote (TV), early-find-early-report vote (EFER) or value-based (VB); and proposed optimization measures.

Authors	Base CD detectors	Aggregation	Ensemble Optimization
		heterogeneous ensembles	
Sobolewski and Woźniak (2013)	unsup	MV	none
Maciel et al. (2015)	sup	TV	none
Toumi et al. (2020)	sup	TV	none
Du et al. (2015)	sup	EFER	base CD detector selection
Hu et al. (2018)	unsup	MV, EFER	base CD detector selection
Hu and Kantardzic (2022)	unsup	MV, EFER	base CD detector selection
Woźniak et al. (2016a)	sup	TV	none
Zhang et al. (2020)	sup	TV	stacking
Perez et al. (2020)	sup	MV, EFER	none
Xu and Klabjan (2021)	semi-sup	MV, EFER	none
		homogeneous ensembles	
Woźniak et al. (2016b)	sup	TV	none
Bu et al. (2016)	unsup	MV, VB	bagging
Pesaranghader et al. (2018)	sup	EFER	data representation
Lapinski et al. (2018)	sup	TV	none
Korycki and Krawczyk (2019)	unsup	MV, EFER	guided subspace selection
Komorniczak et al. (2022)	unsup	TV	random subspaces, ensemble optimization
Okawa and Kobayashi (2021)	unsup	EFER	base CD detector optimization
Nguyen et al. (2022)	sup	MV	weighting

ensemble and the parametrization θ_i of their respective detection algorithms f_i^{det} can have a crucial impact on its diversity. This design aspect is schematically visualized in Figure 6.7.

Selection

Generally, members of a CDDE can be selected from a reservoir of arbitrary candidate concept drift detectors D. Given the continuously increasing research attention to the surrounding field, this reservoir grows steadily. They can e.g. use different detection mechanisms and be supervised or unsupervised. A set of representatives of D are exemplarily introduced in Section 3.2.4. The selection can be done manually by the user, in a guided manner based on various selection criteria or at random.

Manual selection. Optimally, a user manually selecting candidates for the D_i has knowledge of their characteristics, such as monitored metrics or fitness for detecting specific drift types. 17 out of 18 screened publications choose this option. One might argue that, by mostly including theoretical remarks on the selected D, they accumulate associated knowledge. The only deviation is done by Du et al. (2015) as outlined in the following paragraph.

Guided selection. Selecting in a guided manner, however, can be very costly as multiple often crucially different D need to be compared regarding their fitness for specific detection goals. The only screened work following this approach uses diversity and detection-performance-related scores (Du et al. 2015). Using such can lead to finding combinations of D that are apt to detect different concept drift types.

Random selection. Trivially but thus far unexplored, D can also be selected at random which requires neither knowledge nor the definition of selection criteria.

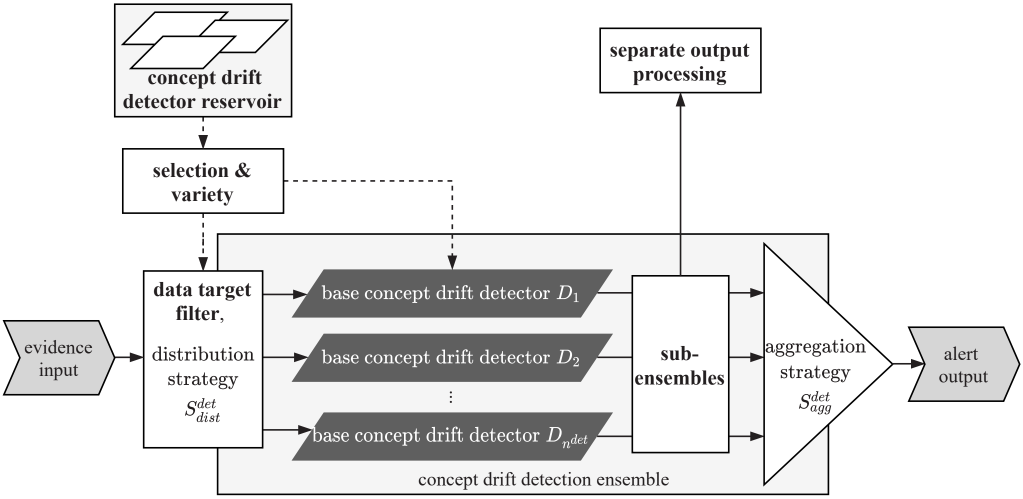

concept drift detector reservoir
separate output processing
selection & variety
evidence input
data target filter,
distribution strategy S^{det}_{dist}
base concept drift detector D_1
base concept drift detector D_2
⋮
base concept drift detector $D_{n^{det}}$
sub-ensembles
aggregation strategy S^{det}_{agg}
alert output
concept drift detection ensemble

Figure 6.7: A schematic visualization of base-concept-drift-detector-related design aspects of concept drift detection ensembles. These involve the selection of members D_i as well as measures to increase their variety. Also, concept drift detection ensembles can be considered as consisting of sub-ensembles. Various accordingly denoted rectangles represent this set of design aspects. Solid connecting lines describe data flows, whereas dashed ones represent asynchronous flows of other types of data or information.

Variety

Variety among the members D_i of a CDDE can be induced by varying member algorithms f^{det}, data targets or parameters θ.

Member algorithm variety. Variety can be implemented by selecting D with different algorithms f^{det} as ensemble members. For purely heterogeneous CDDEs, it constitutes a trivial source of variety. This architectural type of CDDEs is outlined in Section 6.3 and visualized in Figure 6.4. It is also worth noting that this option can be interpreted as straightforward measure to achieve concept drift type coverage as different f^{det} can exhibit fundamentally different detection specializations. This can be interpreted as an inherent advantage of heterogeneous CDDEs over homogeneous ones. Regarding detection specializations, arranging for redundancy, i.e. using D with different f^{det} exhibiting the same concept drift type specialization, has the potential to reduce false positive rates (Hu and Kantardzic 2022). Simultaneously, as multiple D with different f^{det} need to be selected for heterogeneous CDDEs, the selection process can be more costly than for selecting only D sharing a common f^{det} for homogeneous CDDEs.

Data target variety. Alternatively, variety can be implemented by different data targets. This is a part of a CDDE's distribution strategy S^{det}_{dist}. Such can e.g. be defined via employing local D focusing on different data features. One-on-one local monitoring of each of a stream's features can be computationally expensive for high-dimensional data and miss out on concept drift spanning multiple features (Lapinski et al. 2018). Creating separately monitored multi-feature subspaces randomly (Komorniczak et al. 2022) or in a guided manner based on feature-diversity and importance-related measures (Korycki and Krawczyk 2019) is therefore preferred and found to positively impact detection performance.
Combining local and global concept drift detection is an unexplored yet feasible approach. Another way for defining different data targets is by employing different reference data sets with each representing concepts that can be assumed stable (Bu et al. 2016). Alternatively, different lengths of windows containing current or reference data could be employed (Pesaranghader et al. 2018) or the D_i could be assigned to different classes and associated data points (Okawa and Kobayashi 2021).

Parametrization variety. Varying the θ of the D_i can result in crucially different detection behavior. This is the primary aspect of homogeneous CDDEs and is also outlined in Section 6.3 and visualized in Figure 6.4. Such can be set manually (Woźniak et al. 2016b, Bu et al. 2016) or randomly, which is not explored so far. Progressing from initial values of θ towards such leading to better performance can be done using Bayesian optimization (Nguyen et al. 2022).

It should be noted that a combination of multiple of above-outlined options is possible. As explained above, heterogeneous CDDEs already inherently realize variety if members are chosen properly. In contrast, one might argue that choosing options inducing data target or parametrization variety, or both, is compulsory when designing homogeneous CDDEs to sufficiently ensure variety. This needs to be underscored given that variety-inducing design decisions are the most suitable options to increase ensemble diversity. This applies for the selection of members for an ensemble as well. Research question **RQ IV-2** can therefore be answered by referring to the provided insights into these design aspects.

Sub-Ensembles

A CDDE E^{det} can be composed of multiple sub-ensembles E_i^{det} each containing a set of members D_i. On a side note, such a setting requires an aggregation strategy S_{agg}^{det} that enables the consolidation of alert output across multiple ensemble levels, which is further explained in Section 6.4.2. The potential benefits of considering sub-ensembles include specialization, separate output processing and nesting.

Specialization. Forming sub-ensembles enables a CDDE to develop distinct specializations: For instance, each sub-ensemble could combine D_i of a single type, e.g. supervised or unsupervised D (Xu and Klabjan 2021). Alternatively, each sub-ensemble could focus on a specific concept drift type (Hu and Kantardzic 2022). Also, they could be dedicated to a separate part or target of a specific variety-inducing option. For instance, all D_i of one sub-ensemble target one feature subspace while those of another sub-ensemble target another feature subspace (Korycki and Krawczyk 2019).

Separate output processing. As a secondary benefit, detections of sub-ensembles can be separately exploited before global aggregation is done. The rationale behind this is that the specialization of sub-ensembles that alert concept drift detections, while others do not, can implicitly contain information. Among other things, this enables the user to leverage the information of concept drift-type-specific or otherwise-guided adaptation strategies and to improve concept drift understanding. Also, if feasible for the employed machine learning model, adaptation can be limited to specific parts of the model identified to be affected by concept drift to economize computation capacities. None of this is done in existing publications on CDDEs (Hu et al. 2018).

Nesting. Although not described in the literature of CDDEs yet, sub-ensembles can also be nested. This has the potential to increase the granularity of separate concerns they are associated to, or to implement even more detection specializations: For instance, based on the idea of monitoring differently sized windows by one of the D_i each (Pesaranghader et al. 2018), different sub-ensembles could be employed to implement different temporal monitoring ranges and to potentially increase per-window diversity. Other than that, recurring concept drifts could be differentiated by sub-ensembles. Once a new recurrent concept drift pattern can be identified, a new sub-ensemble could be instantiated. It could then be specialized on detecting associated concept drifts marking its re-occurrence.

6.4.2 Aggregation

The detections of multiple base concept drift detectors D_i need to be aggregated to define an ensemble-level detection state. The aggregation strategy S_{agg}^{det} of a CDDE bundles associated measures. This is implemented by different aggregation rules, which constitute another crucial aspect of CDDE design. If sub-ensembles are employed, multi-level aggregation is required. Figure 6.8 provides a schematic visualization of this design aspect.

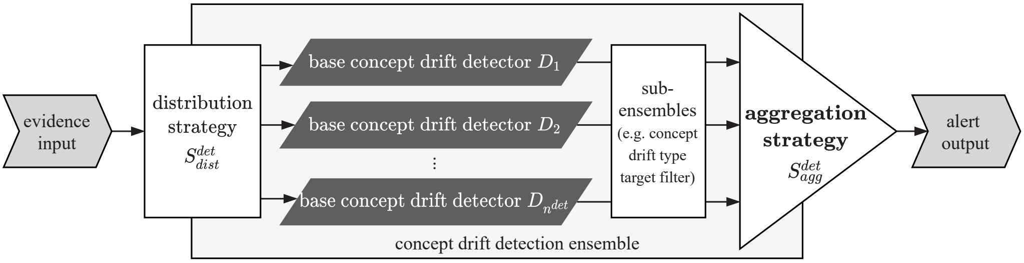

evidence input
distribution strategy S_{dist}^{det}
base concept drift detector D_1
base concept drift detector D_2
⋮
base concept drift detector $D_{n^{det}}$
sub-ensembles (e.g. concept drift type target filter)
aggregation strategy S_{agg}^{det}
alert output
concept drift detection ensemble

Figure 6.8: A schematic visualization of aggregation-related design aspects of concept drift detection ensembles. These involve the consolidation of alert outputs represented as triangle denoted as aggregation strategy S_{agg}^{det}. It is possible across various nesting levels as concept drift detection ensembles can contain sub-ensembles. Solid connecting lines describe data flows.

Aggregation Rules

Feasible options to aggregate the output of multiple D_i are vote and value-based aggregation. While the former is a highly popular choice, the latter is rarely applied.

Vote-based aggregation. All existing publications apply the former option by using either TV, MV or voting based on the early-find-early-report (EFER) rule at least once in their proposed design. As the TV rule defines a detection if the number of votes amounts to at least the threshold's value, MV and EFER are merely special cases. The MV rule corresponds to TV with a threshold value corresponding to half of n^{det}, i. e. the number of D_i, incremented by one for an even and to the rounded-up number of half for an odd number of D_i. TV with a threshold value of 1 corresponds to the EFER rule. Per definition, in contrast to certain TV settings, applying MV cannot lead to ties. When applying TV, a tie could technically be solved via randomization (Sobolewski and Woźniak 2013). However, tie-producing thresholds should

be avoided to increase the interpretability of results.
Greatly different D_i, which is a common setting for heterogeneous CDDEs, might detect concept drifts asynchronously due to their different test algorithms. Therefore, it can be argued that the only vote-based strategy accounting for this is the EFER rule. Other plainly applied options require temporal detection synchronicity (Du et al. 2015). However, temporally buffering votes and considering those for any vote-based aggregation rule is also a feasible solution for this (Maciel et al. 2015).
Vote-aggregating all D_i outputs with uniform weights, which is done in 16 of 18 publications, can be interpreted as an implicit assumption of uniform importance among members. Deviating from this can be achieved via soft voting that defines the individual D_i's importance via weights, which is observed to work well. Initial values can be set manually (Nguyen et al. 2022), randomly or in a guided manner (Zhang et al. 2020). Other further-engineered types of vote-based approaches can also be evaluated in this context. For this, the reader is referred to e.g. the work of Rahman et al. (2002).

Value-based aggregation. Value-based aggregation takes the raw numeric output of D_i and computes a measure that is subsequently evaluated by a concept drift detection algorithm f^{det}. Merely one work chooses this option and computes the mean of all D_i outputs (Bu et al. 2016). For homogeneous CDDEs, the most straightforward but not sole way is to omit the D_i's algorithm f^{det} on member level. It would then instead be applied once on ensemble level to evaluate a computed measure (Bu et al. 2016). Applying this type of aggregation for heterogeneous CDDEs would technically also be possible. It would, however, require specific engineering for evaluating a measure computed from potentially greatly different D_i.
Analogously, the importance of the individual D_i outputs can be defined by weighing, which is also unexplored thus far. Regarding timing, only the aggregation of temporally synchronous detections is tried for this option in the literature (Bu et al. 2016). Additionally aggregating over a time range, e.g. also involving damping dynamics, would be conceivable to better account for temporally dispersed detections.

Multi-Level Aggregation

As indicated above, sub-ensembles require a separate aggregation step. It needs to follow an aggregation rule on each nesting level to define a CDDE's global state. Applied rules can vary, with e.g. MV on sub-ensemble level and EFER on the global level (Hu et al. 2018, Korycki and Krawczyk 2019, Xu and Klabjan 2021), or stay the same across levels.

Despite being unexplored, value-based aggregation could also be done across ensemble levels, analogously requiring specific engineering for heterogeneous CDDEs. Closely related to that is the possibility to separately aggregate for each concept drift alert level. This is done by 7 (Maciel et al. 2015, Du et al. 2015, Woźniak et al. 2016b, Bu et al. 2016, Toumi et al. 2020, Perez et al. 2020, Xu and Klabjan 2021) of 18 publications by counting warning and change-level alerts separately. The remaining 11 publications ignore warning-level alerts. Additionally, one could allow for summation across different levels. This could be done by e.g. considering warning and change-level alerts when computing the ensemble-wide warning state definition. While a change state could not be established yet, the requirements for a warning state might be met (Maciel et al. 2015).

6.4.3 Ensemble Optimization

Over time, the data stream, e.g. emerging or disappearing classes or features, and concept drift patterns can change (Gama et al. 2014). This requires dynamic adaptation of CDDEs and design-specific considerations for optimizing the hyper-parameters of concept drift detectors. This design aspect is schematically visualized in Figure 6.9.

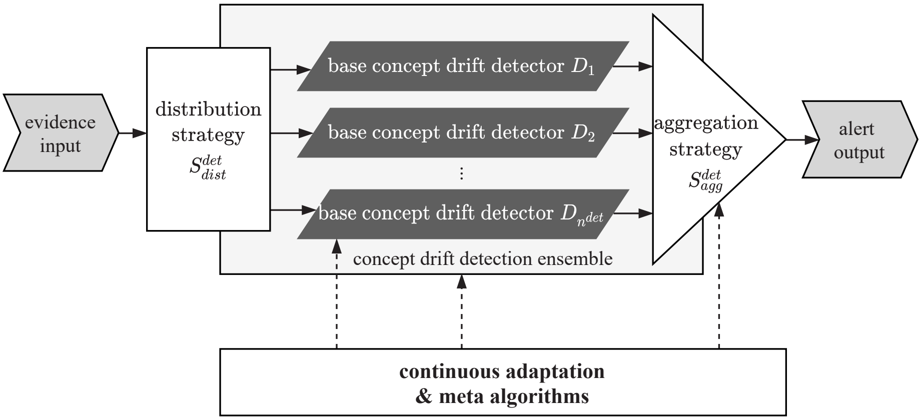

evidence input
distribution strategy S^{det}_{dist}
base concept drift detector D_1
base concept drift detector D_2
⋮
base concept drift detector $D_{n_{det}}$
concept drift detection ensemble
aggregation strategy S^{det}_{agg}
alert output
continuous adaptation & meta algorithms

Figure 6.9: A schematic visualization of optimization-related design aspects of concept drift detection ensembles. These involve the continuous adaptation of various design aspects as well as the application of ensemble meta algorithms. It is represented as rectangle at bottom of the figure. Solid connecting lines describe data flows, whereas dashed ones represent asynchronous flows of other types of data or information.

Continuous Adaptation

Continuously adapting CDDEs can be done either on the level of individual base concept drift detectors D_i or on ensemble level. Approaches utilizing concept drift labels for this purpose are conceivable.

Optimizing base concept drift detectors. The optimization of D_i refers to the search for parametrizations that lead to an improvement in detection accuracy or reduction of detection delay in a structured way. This is not standard practice in the literature and critically lacks a formalized methodology. Instead, the conducted qualitative analysis finds that standard parameters are often kept. Among the analyzed works, only Bu et al. (2016), Okawa and Kobayashi (2021) and Nguyen et al. (2022) do suggest variations of fitting the D_i to data, which can remotely be interpreted as an optimization. However, this fitting is required by the design of the highly specific concept drift detectors they apply. These approaches are not easily transferable to other concept drift detectors, which limits their generalizability.

This inspired the author of this thesis to define and supervise an associated project, which was undertaken by undergraduate student Voß (2025) in his bachelor's thesis. He found that Bayesian optimization is a feasible approach to identify parameter configurations that can critically improve the detection performance of different D_i. However, he also discovered that unsuitable configurations can lead to excessive detections, with increasingly many false positive ones. Consequently, it is imperative to carefully define appropriate parameter search spaces as well as conduct systematic experimentation on various relevant datasets.

Another way to optimize D_i is to address their weakness of being prone to detection cascades. In this context, cascades describe the repeated signalling of concept drift detections potentially caused by previous detections. It can be observed for several detection algorithms f^{det}. The author of this thesis conducted further research on this issue and suggests that a cooldown mechanism provides a remedy. This mechanism allocates time for detectors to readapt their internal scores to a new concept and is visualized in Figure 6.10. It displays a signalled concept drift detection as solid vertical line at t_0 and the suppression of all subsequent ones falling into the cooldown range of 3 temporal units, depicted as dashed lines inside a hatched area. Downstream, this is observed to result in less false-positive detections and thus concept drift alerts of improved quality (Trat et al. 2023).

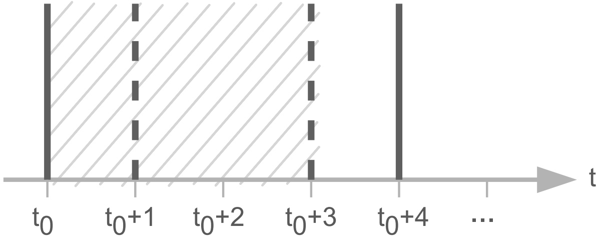


Figure 6.10: Concept drift detector optimization via a cooldown mechanism. The impact of a cooldown length of 3 temporal units on the detection behavior is represented as hatched area. While the first detection at t_0 is signalled, depicted as solid vertical line, further ones falling into the cooldown range are suppressed, depicted as dashed vertical lines. Detections falling after this range are again signalled (Trat et al. 2023).

Optimizing concept drift detection ensembles. Tuning various CDDE-specific parameters is also possible and can even become vital if concept drift characteristics change critically, e.g. affecting different subspaces. Conversely, one might argue that the benefit of optimizing CDDEs or D_i is further increased if concepts recur and associated concept drifts share recognizable properties. Examples of tunable parameters are, among others, data-target-related measures like subspace sizes (Korycki and Krawczyk 2019), number n^{det} of D_i or sensitivity thresholds (Maciel et al. 2015, Komorniczak et al. 2022). Generally, optimization can be done pre-deployment on suitable data as well as continuously post-deployment once concept drift labels become available (Hu et al. 2020, Zhang et al. 2020). This way, instead of remaining static (Nguyen et al. 2022), D_i-assigned weights could be targets of dynamic adaption (Zhang et al. 2020). Other than that, the performance of D_i that exhibit decreasing detection accuracy or increasing delays can be dynamically improved by degrading their weights (Zhang et al. 2020). Not yet evaluated but also a promising research avenue would be to dynamically modify CDDEs by discarding or substituting D_i (Krawczyk et al. 2017).

Leveraging concept drift labels. The term label typically refers to the true value or symbol predicted by a machine learning estimator. However, within the context of concept drift handling, another type of ground truth can additionally be considered: Concept drift labels can be described as information on the location or at least the quantity of concept drifts in a specific range within a dataset or stream. They can be available at different levels, which also determines feasible concept drift detector evaluation types. This is outlined in the following and additionally visualized in Figure 6.11. If concept drift labels are available, the performance of detectors or detector ensembles can be evaluated in an explicit way, which enables their supervised optimization. However, such labels are often hard to determine in real-world scenarios (Krawczyk et al. 2017), especially without detailed expert knowledge. Nevertheless, already one-sided knowledge can help: If one can confidently make the assumption that certain data ranges do not contain concept drifts, i.e. they are exclusively assigned the no-concept-drift label, it can be exploited. This can be done by tuning a concept drift detector's parameters until a satisfyingly low false positive rate is achieved (de Barros and Santos 2019).

Without any expert knowledge, one option to gather concept drift labels is to use a detector that is confidently assumed to exhibit high detection performance on the given data. By considering this detector as a gold-standard, its detections could then serve as labels (Cerqueira et al. 2022). In case only few data with concept drift labels is available, yet unexplored is then the possibility to resample it, e.g. based on the difficulty to detect it. This way, during detector optimization, labeled concept drift data could be presented multiple times to a detector; the more difficult the detection is, the more often the data is presented.

An alternative to obtain real concept drift labels and to conduct supervised detector optimization is to artificially create the labels (Komorniczak et al. 2022). There is a set of methods to synthesize concept drift data in a deterministic way. Among others, Hyperplane (Hulten et al. 2001), STAGGER (Schlimmer and Granger 1986) and the Streaming Ensemble Algorithm (Street and Kim 2001) are examples of such. However, they do not emphasize synthesizing data in a highly realistic fashion but rather data that contains imputed artifacts, which put detectors to the test. Still, especially highly realistic approaches to synthesize concept drift are critically under-researched in the literature. This led the author of this thesis to define a project evaluating the fitness of Hidden Markov Models to probabilistically generate concept drift data at a high level of reality. Undergraduate student Lange (2023) therefore applied these models to initially internalize statistical properties of stream data, which allows one to generate more data with similar properties. Then, by manipulating model parameters, he finds that realistic concept drift data can successfully be generated while offering a high degree of customizability.

Alternatively, concept drift detectors can also be evaluated in the absence of concept drift labels. By measuring the change in performance of estimators that undergo adaptation as a consequence of detections, the detector itself can be evaluated in an implicit way. This way, it can be indirectly optimized with the estimator performance as target criterion.

Independently of the available knowledge, several comments need to be made. Regarding the compilation of data for optimization, it is important to mention that a single data point might not be enough to represent one concept drift sample. In certain cases, a wide range of data might be required for detecting the one concept drift with an extended dynamic profile it contains. Also, even for the detection of abrupt concept drifts, sufficiently large batches might be necessary to store depending on involved concept drift detectors. This is due to characteristics of their data processing method as well as their detection delay.

Meta Algorithms

Within the context of estimator ensembles, the meta algorithms bagging, boosting and stacking or their continuous-learning analogues for data stream scenarios (Krawczyk et al. 2017) are popularly employed approaches. They are introduced in detail in Section 3.1.2. They target the process of integrating and parametrizing suitable and diverse members while simultaneously decreasing estimation variance, specifically for regressors (Witten et al. 2011, pp. 353–354) (Polikar 2012, pp. 2–4). Within the context of CDDEs, the set of design aspects outlined in the previous sections also, among other things, aims at increasing the ensemble diversity. Ensemble meta algorithms are an additional way to foster exactly that. During the qualitative analysis, it becomes apparent that only few publications apply such, as outlined in the following.

For CDDEs, bagging and boosting could be applied pre-deployment. To also incrementally improve existing CDDEs, it can occasionally be applied in a continuous-learning fashion during stream processing once enough concept drift labels are gathered. This requires appropriate data

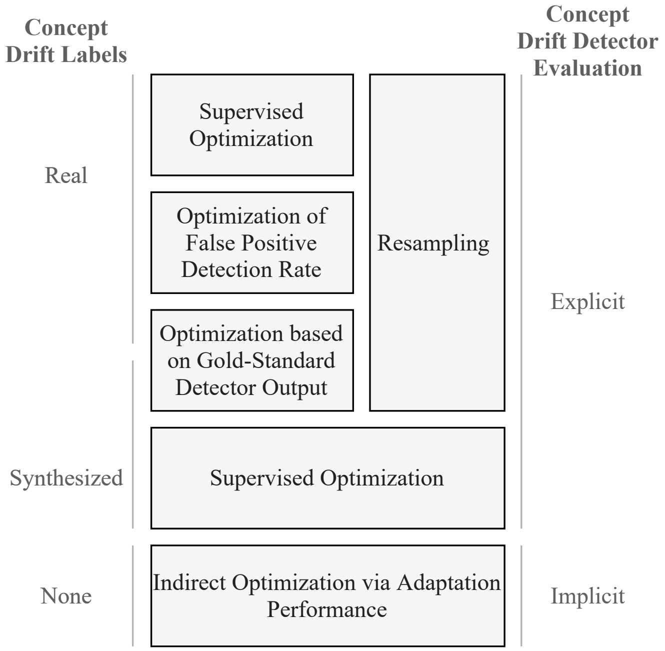

Concept
Drift Labels
Concept
Drift Detector
Evaluation
Real
Synthesized
None
Explicit
Implicit
Supervised Optimization
Optimization of False Positive Detection Rate
Optimization based on Gold-Standard Detector Output
Resampling
Supervised Optimization
Indirect Optimization via Adaptation Performance

Figure 6.11: Concept drift detection optimization methods with respect to the availability of concept drift labels and the evaluation type. The methods are represented as rectangles. The associated availability of concept drift labels as well as evaluation approach is indicated by the rectangles' vertical positions with respect to the axes on both sides. The labels axis specifies the level at which concept drift labels are available. The evaluation axis specifies how detectors can be evaluated during optimization, namely explicitly based on concept drift labels or implicitly based on the performance of an estimator subject to concept drift adaptation.

sampling strategies (Krawczyk et al. 2017), which is invariably a basic requirement of these algorithms. Despite being often applied for homogeneous machine learning ensembles, both meta algorithms could be especially suitable for heterogeneous CDDEs. The reason for this is that sets of highly different ensemble members enhance the diversity-inducing capabilities of these meta algorithms (Witten et al. 2011). A strategy similar to boosting is evaluated for this type of CDDEs by Du et al. (2015). For homogeneous CDDEs, one bagging-based approach is proposed thus far by Bu et al. (2016).

As mentioned before, heterogeneous CDDEs are observed to work well with vote-based aggregation. As finding suitable weights is not trivial, stacking could provide a methodically sound solution, which however requires many concept drift labels. The only existing work implicitly employing stacking uses an estimation model similar to a perceptron (Zhang et al. 2020). All other publications apply aggregation in a deterministic way, i.e. it remains unchanged during stream processing. For the rather rare scenarios where concept drift labels become successively available at reasonable costs, stacking could be explored further.

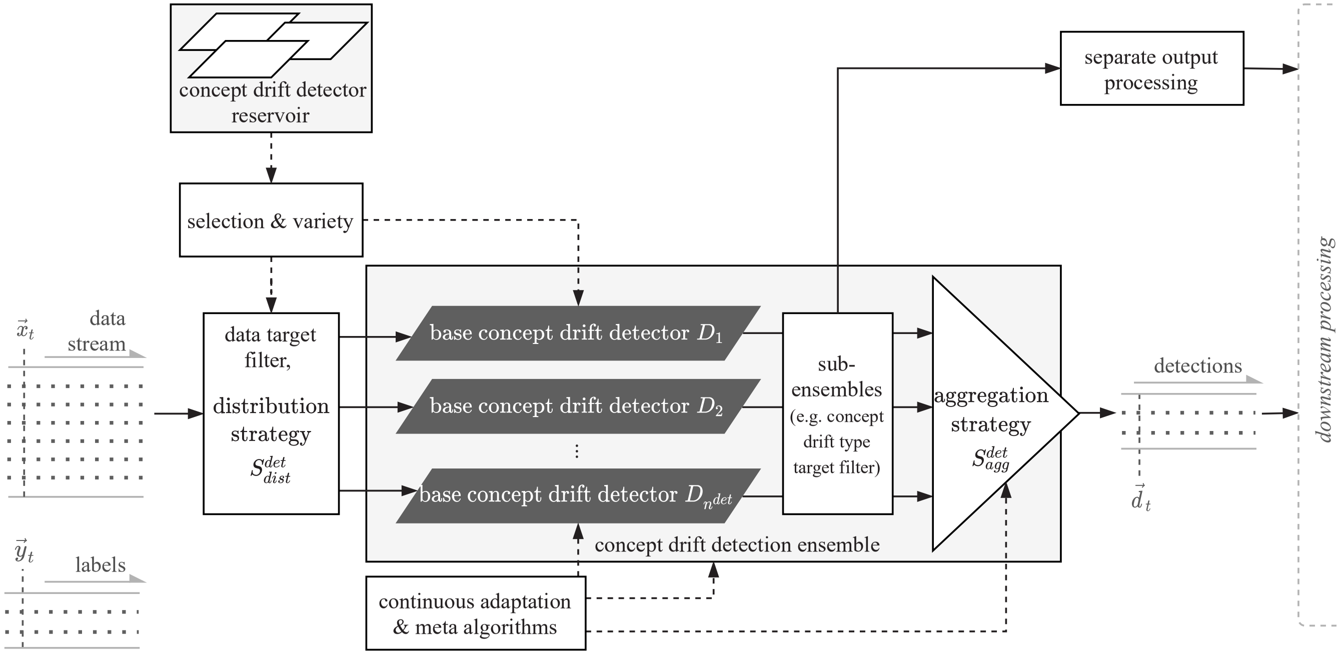


Figure 6.12: Schematic depiction of design aspects and components of concept drift detection ensembles based on Trat and Ovtcharova (2023). Solid connecting lines describe the multi-dimensional data stream and data flows derived from processing it, whereas dashed ones represent asynchronous flows of other types of data or information. Detections can serve as input for downstream applications, such as estimator adaptation or monitoring systems.

While some challenges persist, a plethora of research works pursue the goal of making labels available for machine learning estimation problems (Fredriksson et al. 2021). Estimation labels can also become available or can be obtained at certain costs in certain environments. In contrast, one might argue that the difficulty of obtaining concept drift labels is much greater due to the necessity of very profound knowledge to comprehend concept drift intricacies in data. This might help explaining the rare application of ensemble meta algorithms. Therefore, similar to the still growing trend in machine learning (van Engelen and Hoos 2020), semi-supervised approaches with respect to concept drift labels could be subject of further research.

6.5 Summary

This section introduces the under-researched approach of ensembling concept drift detectors. Doing so results in a structure that is consequently referred to as concept drift detection ensemble (CDDE). This structure has the potential to foster diversity by combining the output obtained from multiple different or differently parametrized concept drift detectors.

Initially, relevant research for this chapter is gathered by conducting a systematic literature review of 220 publications. As targeted by the research question **RQ III**, the existing state as well as the progression of science on CDDEs is determined via a historical and bibliometric as well as a qualitative analysis of the gathered research. This unveils details on the historical evolution of CDDEs, which is described by outlining nascent research trends as well as notable contributions. This chapter then also contributes to the field by providing an attempt of formally defining as well as taxonomizing CDDEs. Additionally, persisting research gaps are identified.
Furthermore, a comprehensive set of important CDDE design aspects is accumulated and their interplay is discussed, as pursued by **RQ IV**. The associated insights are condensed within a blueprint supporting the sound and careful design of CDDEs. It is provided as a schematic depiction (cf. Figure 6.12) referencing a detailed catalog of design aspects discussing their feasible configurations, augmented by knowledge on machine learning ensembles.

Within this thesis, research on CDDEs is vital as it formally introduces these structures as powerful solution to detect concept drifts. They have the potential to improve explicit concept drift adaptation by providing profound information on the existence and timing of concept drifts, which in turn enables more targeted reactions. Given the finding of this chapter that the majority of existing CDDE works focus on improving the concept drift detection performance, this thesis pursues a novel avenue: It puts emphasis on improving the concept drift adaptation performance, which is arguably more critical for increasing the robustness of machine learning models (Woźniak 09.10.2023). As a result, this chapter lays the groundwork for designing versatile and diverse

CDDEs for use in practice. Therefore, based on the framework provided in the form of the reference architecture proposed in Chapter 5, the gained design insights are leveraged to conceptualize a core component of the data-stream-applicable method in Chapter 7: A composition of a CDDE that is used to continuously monitor the robustness of a machine learning regressor. Also, a set of projects overseen by the author of this thesis going beyond its scope are referenced. Among other aspects, they investigate methods for concept drift detector optimization and leveraging labels.

In a broader scientific context, the circumstance that CDDEs are under-researched is surprising, considering the high performance potential and versatility of machine learning ensemble methods. Also, the first publications on CDDEs provide a very positive assessment of their performance. The findings and contributions presented in this chapter are therefore intended to stimulate further research activities.

7 Concept Drift Adaptation for Robust Productive Regressors

"The people who design our world usually never take a biology class."
— Janine Benyus, founder of the Biomimicry Institute[11]

In the course of this chapter, the various research strands of this thesis converge and are transformed into practically applicable artifacts. One of them constitutes the core contribution of this thesis: The stream-applicable method for concept drift adaptation in regression scenarios (SAM-CDAR). This method is designed to sustain the robustness of productive regressors deployed into data streams of industrial real-world problems. The biological reference advocated by the above-provided quotation is strived for here: The method SAM-CDAR takes as template processes from nature, more specifically from evolutionary biology, as well as makes extensive use of machine learning ensemble methods for the design of its components.

Section 7.1 provides an overview of the research questions that guide the scientific approach of this chapter. Novel methods for detecting and adapting to concept drift, specifically considering properties of regression problems are proposed in Section 7.2 and 7.3, respectively. These are then the building blocks for the composite method SAM-CDAR that leverages their combined strengths. Section 7.4 details its technical intricacies.
In parts, Section 7.3 contains verbatim passages from a work previously published by the author of this thesis (Trat et al. 2024). These passages are not individually cited when the author is responsible for the intellectual content and research insights. Contributions from co-authors as well as figures and tables are explicitly cited.

[11] The term biomimicry, coined by Benyus, is composed of the Greek words *bios* and *mimesis* meaning life and imitation. She argues that a wide variety of engineering disciplines bear the potential to greatly benefit from nature as a template for designing solutions. In a way, nature thus may serve as a model, measure and mentor (cf. Benyus 2002).

7.1 Research Questions

The guiding scientific structure of this chapter is designed to answer research question **RQ V**. It is operationalized by formulating several sub-questions that build on the insights gathered in previous chapters and attempt to synthesize the respective novel aspects:

RQ V How can one sustain the robustness of practically applied machine learning regressors against concept drift?

- **RQ V-1** How can one optimize the performance of concept drift detection in industrial regression problems?
- **RQ V-2** How can one discard outdated concepts and maintain (imminently) valid ones in industrial regression problems?
- **RQ V-3** How can one design a stream-applicable method integrating the matters addressed by the preceding sub-questions?

RQ V refers to sustaining robustness against concept drift in the sense of, on the one hand, detecting it and, on the other hand, adapting to it. As adaptation targets, it specifies estimators as regressors for all sub-questions. Regarding detecting, **RQ V-1** explores a set of approaches that can be leveraged to identify concept drifts in regression scenarios. Associated answers are provided in Section 7.2.
Necessarily constituting the second step of the procedural order, **RQ V-2** demands for a solution meaningfully making use of concept drift detections to continuously conduct adaptation of the regressors representing effective data stream concepts. One such solution is discussed in Section 7.3.

RQ V-3 finally explores the methodical integration of insights acquired within the context of the preceding sub-questions. More specifically, it inquires about the orchestration as well as interaction of artifacts derived from these insights. Section 7.4 responds to this sub-question in the form of a compound method.

7.2 A Method Leveraging Concept Drift Detection Ensembles

Especially within the context of regression problems, there are hardly any solutions for identifying concept drift occurrences. However, such problems are widespread in industrial real-world scenarios. For this reason, this section proposes a method that provides concept drift detection approaches for such kinds of problems. Also, being based on theoretical insights into and design guidelines for CDDEs acquired and developed earlier in this thesis, the method leverages the performance potentials of machine learning ensembling. In the following, the method's rationale as well as its technical details are outlined. Additionally, several remarks on its practical application are provided.

7.2.1 Rationale and Motivation

An analysis of the state of science on concept drift detection for regression problems, as overviewed in Section 4.1.1, reveals this area as being heavily under-researched. By proposing method E^{det}, a range of novelties are contributed to this area. This is achieved by E^{det} leveraging the concept drift evidence inherent in timeseries of regression error scores as well as in classification-like representations obtained by transforming the original regression problems.

Due to their great performance potential, as outlined in Section 4.1.2, E^{det} is based on the conceptual and algorithmic foundations of CDDEs, which directly addresses research question **RQ V-1**. It is designed by applying the CDDE blueprint introduced in Chapter 6. An abstract visual representation of E^{det} is provided in Figure 7.1. It is conditionally agnostic with respect to the selection of concept drift detectors employed as base members, depending on their expected input data type, and versatile with respect to the distribution of concept drift evidence to them. Also, E^{det} allows fine-grained control of its global detection emission behavior by means of an aggregation strategy.
This way, guided by **RQ V-1**, E^{det} permits parametrization to satisfy the requirements of various real-world industry scenarios situated within the trade-off continuum between not overlooking any concept drifts and favoring exclusively high-confidence detections. The former is achieved by leveraging sets of concept drift detectors covering a wide range of concept drift types and magnitudes and paying great attention to individual detections. The latter, in contrast, is achieved by algorithmically demanding sufficient detections in close temporal proximity via aggregation.

7.2.2 Design

To produce the CDDE-based method E^{det}, the design principles introduced throughout Section 6.4 and visualized in Figure 6.12 are applied. The process of doing so as well as required formalizations are detailed in the following.

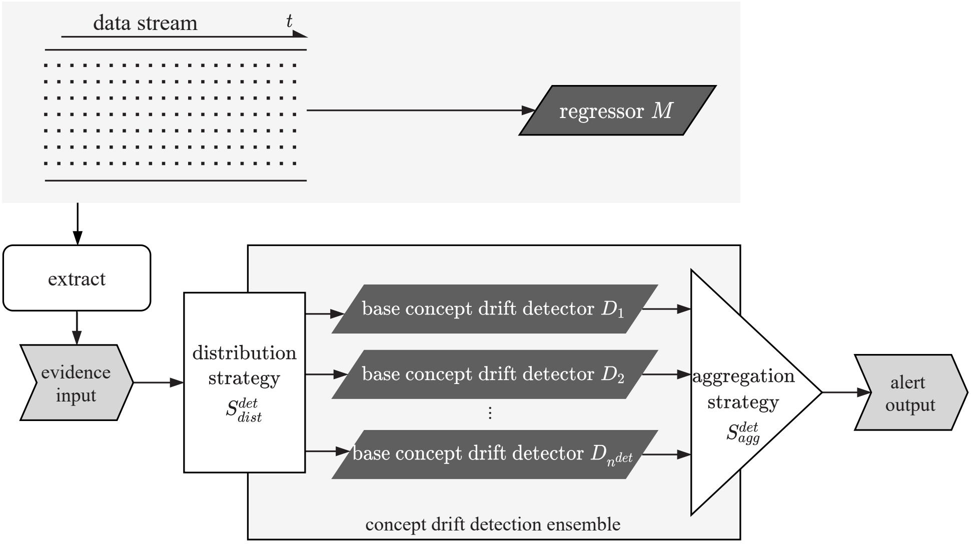

data stream
t
regressor M
extract
evidence input
distribution strategy S_{dist}^{det}
base concept drift detector D_1
base concept drift detector D_2
base concept drift detector $D_{n^{det}}$
aggregation strategy S_{agg}^{det}
alert output
concept drift detection ensemble

Figure 7.1: The suggested method, designed based on the concept drift detection ensemble blueprint introduced in Chapter 6. It is employed to monitor for and alert concept drift occurrences in the data streams of regression problems.

Data Target Variety

A critical design aspect of E^{det} is the careful selection of its input data streams. Being a core benefit of CDDEs, it exploits diverse sources and provides them as input for its member concept drift detectors D_i via a distribution strategy S_{dist}^{det} as visualized in Figure 7.2. Primarily, a diverse set of mean-error scores, which quantify the performance of regressors, is exploited. While technically various other scores are possible, the following ones are employed:

- mean absolute error (MAE)
- mean squared error (MSE)
- root mean squared error (RMSE)
- mean absolute percentage error (MAPE)

The computation of theses error scores is executed on successive batches of actual target values y_u and respectively regressor-estimated values $\hat{y}_u$, with u marking temporal context, and yields one value each. Further introduction into their technical details and formalizations can be found in Section 3.1.3.

The benefit of such a diverse set of error scores lies in the differences of its elements. One might

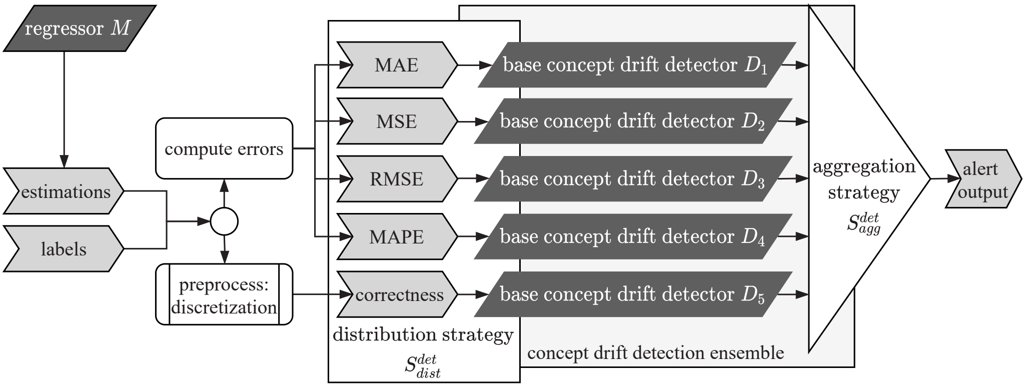

regressor M
estimations
labels
compute errors
preprocess: discretization
MAE
MSE
RMSE
MAPE
correctness
distribution strategy
S^{det}_{dist}
base concept drift detector D_1
base concept drift detector D_2
base concept drift detector D_3
base concept drift detector D_4
base concept drift detector D_5
concept drift detection ensemble
aggregation strategy
S^{det}_{agg}
alert output

Figure 7.2: Method E^{det}'s data inputs. A diverse set of sources is distributed to its members D_i via a distribution strategy S^{det}_{dist}. Apart from exploiting several regression error scores, it also exploits an input computed from a classification-based representation of the regression problem. This requires further preprocessing, which is presented in more detail in Figure 7.3.

hypothesize that different concept drift types and magnitudes manifest as different patterns and degrees of clarity in these scores. If this holds true, these properties could be exploited for the purpose of concept drift detection by monitoring the different dynamic patterns of these scores. This is implemented by distributing the score streams over E^{det}'s members. For instance, the MAE exhibits a focus on absolute error score value magnitudes heavily reflecting properties of an arbitrary regressor's input data. The MSE greatly accentuates large errors over smaller ones due to the way the non-linear squaring operation has a much larger impact on large numbers. Given that the RMSE is computed as the square root of the MSE, cf. (3.3), it works similarly as and shares properties of the MAE. In contrast, the MAPE describes a notion of error relative to the actual target values y_u estimated by a regressor. It is therefore a scale-independent error score (Hyndman and Koehler 2006) and establishes dependencies on numerical properties of the latter. These, among other aspects, translate into an accentuation and an attenuation of the same absolute regression errors observed for smaller and larger target values, respectively.

Secondarily, a classification-based representation of the regression problem, formulated via discretization, is exploited. It requires several preprocessing steps, which are outlined as follows and additionally presented in Figure 7.3. Discretization is implemented by breaking up the full range of the actual target values y_u, sorted by increasing value, into a user-defined number z of bins. Applying a quantile-based strategy, the bins' lower and upper bounds are defined to have all hold the same number of y_u, respectively.

For each data batch containing b data points $\vec{x}_u$, a regressor estimates values $\hat{y}_u$ that are then assigned to the bins based on their previously defined bounds. A correctly assigned $\hat{y}_u$, i.e. one that ends up in the same bin as its associated y_u, is represented by a boolean value of $true$.

An analogously incorrectly assigned one is represented by a value of $false$. A user-defined threshold τ^{det}_{class}, with $0 \leq \tau^{det}_{class} \leq 1$, is then applied onto the resulting b boolean values. The classification-specialized concept drift detector receives the following input:

a. A data point representing a correct classification, if at least a fraction of τ^{det}_{class} $true$ values are gathered for a data batch.

b. A data point representing an incorrect classification, if less than a fraction of τ^{det}_{class} $true$ values are gathered for a data batch.

Finally, the bin bounds are updated if new y_u are received.

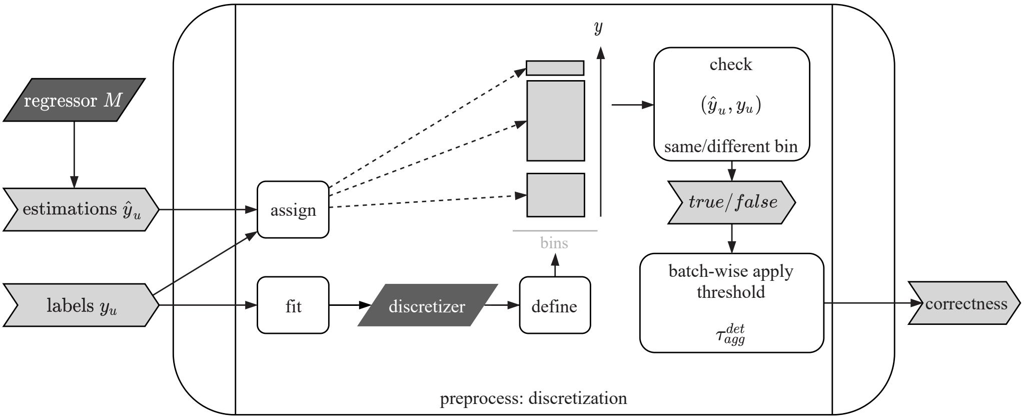


Figure 7.3: Preprocessing of E^{det}'s input yielding a classification-based representation of a regression problem. It includes discretizing actual target values y_u to form z contiguous bins as well as the assignment of associated estimated values $\hat{y}_u$ to these bins. A data-batch-wise threshold is applied to the relative number of correct assignments, i.e. both values being assigned to the same bin, to derive a correctness value. This constitutes a step included in the input data distribution approach depicted in Figure 7.2.

Base Concept Drift Detector Selection and Variety

The concept drift detectors constituting the members of E^{det} are elements of a set $\mathbf{D} = \{D_1, D_2, ...D_{n^{det}}\}$. As remarked above, each D_i receives a different error stream or a binary signal as input with

$$D_i := f^{det}_i(\theta_i) \quad \text{with } i \in \{1 \dots n^{det}\}, \text{ detection algorithm } f^{det} \text{ and parametrization } \theta. \quad (7.1)$$

Both f^{det} and θ are varied across the D_i. This way, the potentials of heterogeneous and homogeneous CDDE architectures, respectively, are exploited. The D_i taking error scores as input are characterized by featuring the same f_i^{det} with varied θ_i. Therefore, they may be considered a sub-ensemble of E^{det} with a homogeneous architecture. While any concept drift detector able to process continuous-valued input can be employed as f_i^{det} of these D_i, popular choices are Adaptive Windowing (ADWIN) as well as the Page-Hinkley test (PH). Technical details of these detectors are provided in Section 3.2.4.

With respect to the input computed from the classification-based representation of a regression problem, within this context, a detector able to process correctness-related binary inputs is employed. The popular choice for this f_i^{det} is the Drift Detection Method (DDM), which is also introduced in Section 3.2.4. When considering both the afore-described homogeneous sub-ensemble and this binary input detector, E^{det} can be characterized as featuring a heterogeneous architecture. A visualization of this ensemble architecture topology is provided in Figure 7.4.

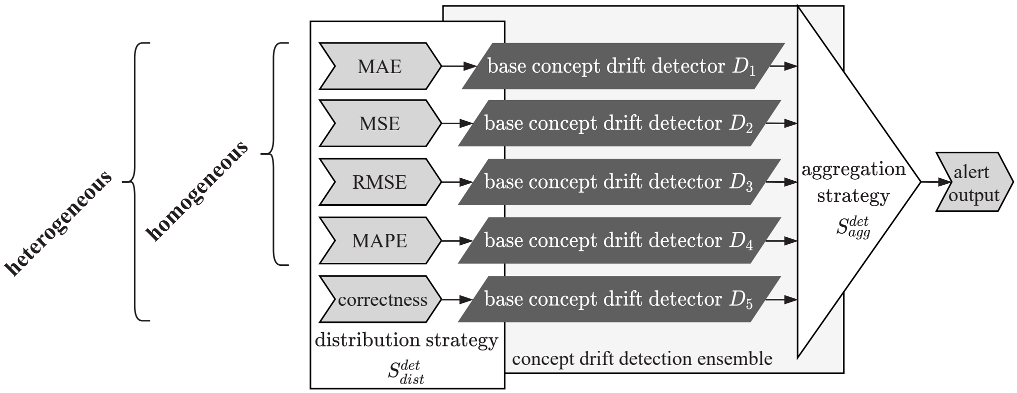

heterogeneous
homogeneous
MAE
MSE
RMSE
MAPE
correctness
base concept drift detector D_1
base concept drift detector D_2
base concept drift detector D_3
base concept drift detector D_4
base concept drift detector D_5
aggregation strategy S_{agg}^{det}
alert output
distribution strategy S_{dist}^{det}
concept drift detection ensemble

Figure 7.4: Method E^{det}'s ensemble architecture topology. Several regression-error-processing concept drift detectors form a homogeneous sub-ensemble. When considering this sub-ensemble and the correctness-related binary-input-processing detector together, E^{det} can be characterized as a heterogeneous CDDE.

The ultimate choice of the f_i^{det} and θ_i cannot be defined in a static fashion. It needs to be based on knowledge of as well as specifically tailored to the respective application problem. Optimally, expert insights are exploited for it.

When compared to the state of science of concept drift detection for regression problems introduced in Section 4.1.1, the previously outlined variety-inducing data selection approach providing input for a selection of concept drift detectors constitutes a distinct novelty. It makes available multiple sources of concept drift evidence and enables the user to choose from a broad set of base concept drift detectors D_i to process this evidence: Such able to process continuous-valued as well as binary input. Therefore, this approach fosters ensemble diversity and thereby addresses **RQ V-1**.

Aggregation

As elaborated on in the previous subsections, E^{det} receives various sources of concept drift evidence as input and distributes these to its members D_i. After processing it, each D_i outputs an alert, which, depending on f_i^{det}, indicates the absence or a certain magnitude of detected concept drift (cf. Section 3.2.4). The set of all detections gathered for a certain evidence input is then subject to an aggregation strategy S_{agg}^{det}. It is individually applied for each alert magnitude, i.e. in case of a discretization into warnings and changes, both are aggregated separately. This is done via threshold vote (TV) based on a user-parametrized threshold τ_{agg}^{det}, with $\tau_{agg}^{det} \in \mathbb{N} \setminus \{0\}$ and $\tau_{agg}^{det} \leq n^{det}$, determining the minimally required number of D_i emitting an alert of the respective magnitude that causes an ensemble-global detection. This approach constitutes a generalization of the combination rules introduced by Woźniak et al. (2016b) (cf. Section 6.4.2).

Instead of requiring all alerts to occur simultaneously, they are aggregated over a certain time horizon m^{det} as initially introduced by Maciel et al. (2015). The value of m^{det}, with $m^{det} \in \mathbb{N} \setminus \{0\}$, is also parametrized by the user and expressed as the number of successive data batches forming the basis concept drift evidence is extracted from to produce detector inputs. This temporal aggregation of detections is visualized in Figure 7.5. During m^{det}, the actual number of member detections, which is evaluated against τ_{agg}^{det}, is calculated as the sum of members emitting at least one detection. Thus, multiple detections of each single member are still counted only once. After each ensemble-global detection, the start of m^{det} is reset. This ensures that a detection that contributed to an ensemble-global detection cannot contribute in a subsequent one.

In summary, the user can parametrize E^{det}'s aggregation strategy S_{agg}^{det} to optimize the processing of alerts obtained from different concept drift detectors. This enables her to consider the concept-drift-related conditions of various application scenarios and therefore addresses **RQ V-1**.
For the sake of sound notation, the aggregation strategy can be formalized as a function $S_{agg}^{det}(\theta_{agg}^{det})$ with

$$\theta_{agg}^{det} = \{\tau_{agg}^{det}, m^{det}\}. \tag{7.2}$$

Further Design Remarks

Efficient input discretization. As elaborated on above, implementing a classification-based representation of the regression problem as input data for E^{det} requires discretization of the actual target values y_u. While the maximum and minimum of the full value range could straightforwardly be updated over the course of the data stream, updating the bin bounds in a meaningful way requires frequent processing of the individual values of y_u. Storing all incoming y_u in an ever-growing set would well serve the computation of bin bounds but theoretically require infinite memory

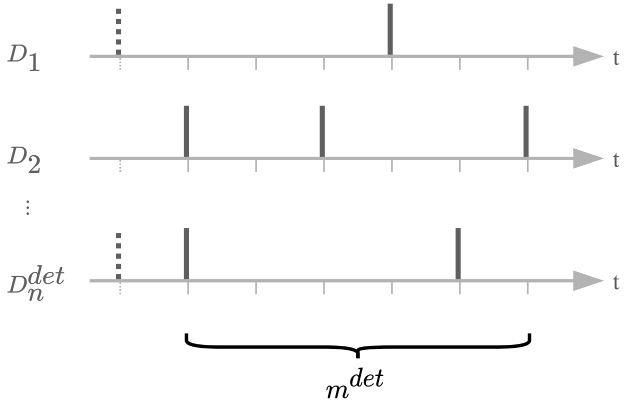


Figure 7.5: Temporal aggregation of member concept drift detections. To be considered for aggregation, they need to fall in a time horizon of maximum length m^{det}. These detections are represented as solid vertical lines. Detections observed further back in the past are not considered for aggregation, which is represented as dashed vertical lines.

resources.

To memory-efficiently realize a set that is representative of the value distribution of the y_u, reservoir sampling with an exponential decay is applied. At the time of deploying E^{det} into the data stream, the reservoir is initialized with the y_u of a previously gathered historical dataset. Then, continuously received y_u are added to the reservoir. With the growing size of the reservoir, based on probabilistic sampling from an exponential decay function, it becomes increasingly likely that older y_u are dropped on newer ones being inserted. The user controls the dynamic of the decay via a bias parameter and the maximum size of the reservoir. The implementation of this sampling approach, as employed for E^{det}, guarantees insertion of new y_u, which constitutes a deviation from the original algorithm by Aggarwal (2006) that only probabilistically allows it.

Modifying the MAPE. What needs to be addressed is that the MAPE can suffer from numerical instabilities. Specifically, cf. (3.4), target values numerically close to 0 lead to the fraction converging towards infinite values (Hyndman and Koehler 2006). In order to ensure its robust usage, E^{det} consistently receives MAPE inputs treated by introducing ϵ, which corresponds to the constant small positive real value of a computer's error resulting from rounding floating values[12]:

$$\text{MAPE}^{robust}(y_u, \hat{y}_u) = \frac{1}{b}\sum_u |\frac{y_u - \hat{y}_u}{\max(|y_u|, \epsilon)}| \tag{7.3}$$

[12] This treatment is e.g. included in the popularly employed machine learning library Scikit-Learn. Its user guide is available at `https://scikit-learn.org/stable/modules/model_evaluation.html`, accessed: 18.06.2025.

To further stabilize the MAPE's computation, it can additionally be modified by adding a real-valued positive offset o to the value of ϵ or to y_u and $\hat{y}_u$, as formalized in (7.4) and (7.5), respectively. The user is given control over selecting a value for o as its optimum can vary across different data scenarios.

$$\mathrm{MAPE}^{mod^1}(y_u, \hat{y}_u) = \frac{1}{b}\sum_u |\frac{y_u - \hat{y}_u}{\max(|y_u|, \epsilon + o)}| \tag{7.4}$$

$$\mathrm{MAPE}^{mod^2}(y_u, \hat{y}_u) = \frac{1}{b}\sum_u |\frac{(y_u + o) - (\hat{y}_u + o)}{\max(|y_u|, \epsilon)}| \tag{7.5}$$

Full parametrization. To complete the formalization of the proposed method E^{det}, its parametrization is defined as

$$\theta^{E^{det}} = \{\mathbf{D}, S^{det}_{dist}, S^{det}_{agg}\}. \tag{7.6}$$

7.2.3 Practical Integration

In the following paragraphs, a range of considerations for the effective practical application of the proposed method E^{det} in industrial data stream scenarios are provided.

Parametrization. Scenario-aware tuning of the method's parametrization $\theta^{E^{det}}$, cf. (7.6), is crucial. The member set $\mathbf{D}$ may possibly be one of the most vital aspects in that regard. Without prior knowledge on potentially occurring concept drift types in a data stream, assembling a $\mathbf{D}$ covering a wide range of such may be a feasible approach. In contrast, with such knowledge being available, $\mathbf{D}$ should contain D_i featuring detection algorithms f^{det}_i that are fit to detect the respectively expected concept drift types. This variant also provides an answer to **RQ V-1**. Additionally, one might even argue that several D_i featuring the same f^{det} with suitably selected different parametrizations could be employed redundantly to improve their joint detection coverage.

Also, the threshold τ^{det}_{agg} for ensemble-global detections must be set with careful consideration of its pronounced interdependence with the aggregation time horizon m^{det}. For instance, in application scenarios that demand the number of false positive detections to be kept at a minimum, higher values need to be selected for the former. Conversely, in such characterized by high costs attributed to overlooking concept drifts while false positives are relatively uncritical, lower values can be selected. In all cases, selecting high values for m^{det} promotes the emission of detections as the D_i are allocated more time to accumulate the threshold-exceeding number of detections, while low values impede it.

Transferability. One of the core method design elements proposed in this section, namely the processing of one or multiple regression error scores, addresses an under-researched aspect of the field of concept drift detection and adaptation for regression problems (cf. Section 4.1.1). This design element is neither exclusively tied to E^{det} nor to CDDEs in general. On the contrary, it can be employed in a standalone fashion. Within this context, there is also a lack of studies that comparatively examine the utilization of different instances of such error scores and explore the associated intricacies as well as opportunities.

Input data preprocessing. As described in this section, E^{det} employs data preprocessing that involves computations on entire data batches consistently producing exactly one value passed to each member. For regression error scores, this is achieved via averaging and for correctness-related binary inputs via a threshold-based operation. In the former case, a point instead of batch-wise processing of concept drift evidence is conceivable. This, however, might suffer from the potentially high variance of observed error values. In the latter case, a similar remark can be made, however, without the variance-related drawback. Therefore, as an variant of E^{det}, a mixed form of batch and point-wise processing could be experimentally evaluated. It might bring forth a feasible approach for application scenarios where members processing correctness-related inputs are observed to underperform.

Outlook. There are multiple feasible approaches to transform a regression into a classification problem. The proposed method employs a binning-based discretization approach as introduced above in this section. As the literature on concept drift detection for regression problems (cf. Section 4.1.1) hardly covers this aspect, studies comparing various approaches with their respective impact on detection performance would be desirable. More insights could augment the capability of priorly classification-problem-related concept drift detectors to be also utilized for regression problems.

Also, the usage of E^{det}'s concept drift detections for a diverse range of purposes could be evaluated. As illustrated later in Section 7.4, this thesis leverages such for the prominent purpose of concept drift adaptation. However, utilizing detections for separate output processing (cf. Section 6.4), for instance via identifying suitable sub-ensembles of E^{det} that establish certain specializations on specific concept drift types (cf. Hu et al. 2018), as well as concept drift understanding, i.e. identifying root causes of concept drifts and providing further explanatory insights (Pinto et al. 2019, Xiang et al. 2023), are conceivable.

7.3 A Nature-Inspired Concept Drift Adaptation Method

Industrial practitioners are increasingly faced with the task of transferring machine learning estimators that exhibit high performance on static datasets to productive streaming use. Given the requirement that such need to be robust against concept drift as well as the limited choice of and experience with such featuring purely continuous learning algorithms (cf. Section 3.1.1), it is a non-trivial task. Therefore, to support accomplishing it, a well defined concept drift adaptation method as well as guidelines to purposefully employ it are proposed in this section. This method aims at sustaining the performance of arbitrary machine learning regressors in ensembles and, in its core, enables internalizing and maintaining new data stream concepts while discarding outdated ones. Guided by research question **RQ V-2**, it is specifically designed for industrial real-world regression problems. The research producing this method is the subject of a work (Trat et al. 2024) separately published by the author of this thesis.

7.3.1 Rationale and Motivation

Given the state of science pointed out in Section 4.2.1, the proposed method is novel as it combines performance and experience-based mechanisms for ensemble development. Additionally, it is model-agnostic with respect to regressor model selection as well as learning algorithm type, i.e. is not limited to continuous learning approaches but supports the stream use of batch-based ones (cf. Section 3.1.1). With batch-based learning approaches being far more utilized in industrial practice (Gupta et al. 2023), the method is very broadly applicable. It is an explicit concept drift adaptation approach as members are instantiated and added to the ensemble on change detections. Once a maximum ensemble size, defined by the user, is reached, members are rotated based on mechanisms designed to imitate naturally occurring processes described by evolutionary biology. This way, this method directly addresses the associated research gap outlined in Section 4.3. Driven by the motivation to more reliably and efficiently sustain the robustness of regressors than state-of-science methods, this one is designed specifically for the boundary conditions of real-world industrial data streams with delayed label availability. Also, it features mechanisms that economize the utilization of these labels. Figure 7.6 schematically displays this method on a high level.

The earliest iteration of the proposed method was inspired by the stable-reactive duality encoded in the method Paired Learners by Bach and Maloof (2008). At its core, this method considers an ensemble of a stable and a reactive learner. While the stable learner is continuously trained after its instantiation, the reactive one is exclusively trained on recent data from a stream. The former

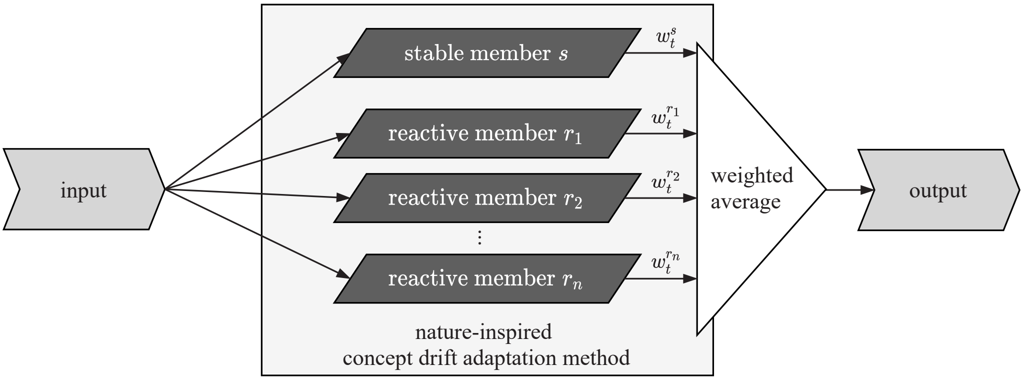


Figure 7.6: The proposed nature-inspired concept drift adaptation method. It aims at sustaining the robustness of arbitrary machine learning regressors, which are members of an ensemble. The various concepts represented by these members can be maintained over time, i.e. either preserved and dynamically varied in importance or dropped.

is employed to define the ensemble-global predictions. On incorrect classifications by the stable learner, while the reactive one is correct with respect to the same data points, the stable is reset (Bach and Maloof 2008).

Several experiment-observation-based design iterations as well as aspects derived from **RQ V-2** led to the method's improvement. One of these aspects, derived from the research goal of discarding and maintaining data stream concepts, is to optimally balance the stability and plasticity of arbitrary user-selected regressors. To leverage ensembles constitutes an immensely promising approach to achieve this goal and thus contributes to answering **RQ V-2**. On the one hand, stability needs to be established by maintaining concepts for as long as they are valid, which is determined via member performance quantification. On the other hand, plasticity needs to be leveraged to support concept drift adaptation. Therefore, new concepts are internalized by scaling up or rotating members, while outdated ones are discarded. Transitions between concepts are smoothed by dynamic experience-based member weighing. Further aspects derived from **RQ V-2** are highlighted throughout the following subsections.

7.3.2 Design

In the following, the proposed estimator ensemble method E^{est} is described in detail[13]. This entails the relevant formulae and algorithms for its dynamic structure and development as well as

[13] The Python implementation is openly available for application in practice at `https://github.com/m-martin-j/NICDAM`.

the underlying rationale borrowed from evolutionary biology for apt mechanism design, which are additionally visualized in Figure 7.7.

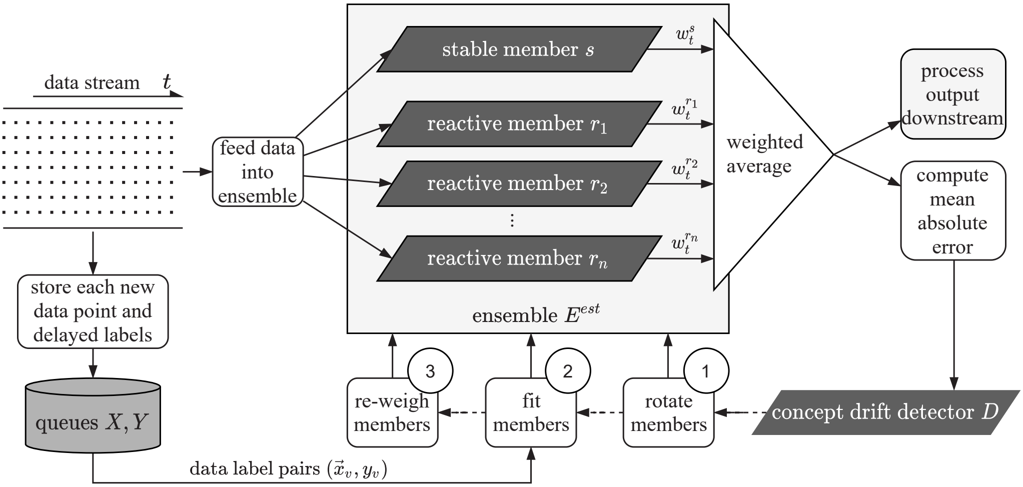


Figure 7.7: Core mechanisms of the proposed nature-inspired concept drift adaptation method. Being an explicit one, ensemble maintenance occurs based on a concept drift detector's output via member rotation. In this figure, the method employs a supervised concept drift detector, which could be replaced with an unsupervised one. Training and reweighing is resourcefully conducted when labels become available, with algorithms being inspired by evolutionary biology.

Ensemble Structure and Member Development

As represented by lines 1 to 5 in Algorithm 1, before deployment, the ensemble E^{est} only contains a stable member s with dynamic weight $w_t^s = 1.0$ initially. It is trained on an arbitrarily sized historical dataset gathered also before deployment. After deployment (cf. line 6 onwards), E^{est} can instantiate up to n_{max}^{est} reactive members $r_i \in \mathbf{R}$ with $|\mathbf{R}| <= n_{max}^{est}$ and $i \in \{1 ... n_{max}^{est}\}$. Each r_i is assigned a dynamic weight $w_t^{r_i}$. For inference, the ensemble-global estimation output is computed as the weighted sum of the outputs of its members.

All members of E^{est} consistently employ the same arbitrary batch or continuous-training-type regressor model f^{est} initially selected by the user. An arbitrary construct capable of detecting concept drifts, e.g. a detector D, is parametrized to achieve timely detections of change-level concept drifts. As input, it receives either the mean absolute error (MAE) of E^{est}'s ensemble-global predictions in a supervised approach or, in an unsupervised one, it might receive raw or preprocessed input data. After deployment, E^{est} constantly makes predictions $\hat{\vec{y}}_t$ of target regression value vectors $\vec{y}_t$, also referred to as label vectors, for batches $\mathbf{X}_t$, with batch size

$b = |\hat{\vec{y}}_t| = |\vec{y}| = |\mathbf{X}_t|$ and $b \geq 1$ at any time $t >= 0$ (cf. line 7). Predicted values $\hat{y}_u$, label values y_u and data points $\vec{x}_u$ are the respective elements of the aforementioned vectors and sets, i.e. $\hat{y}_u \in \hat{\vec{y}}_t$, $y_u \in \vec{y}_t$ and $\vec{x}_u \in \mathbf{X}_t$, and have u as individual temporal reference, with $u \leq t$.

At certain points in time t, one or more previously unseen label values y_v might become available, making it possible to conduct supervised member development of E^{est}. The following holds:

$$t = v + \delta \quad \text{with arbitrary temporal delay } \delta. \tag{7.7}$$

These y_v correspond to data points $\vec{x}_v$, with v marking the point in time $\vec{x}_v$ arrived at. Associated predictions $\hat{y}_v$ have been made. Labels and their predictions are gathered as associated pairs in a queue Y and data points in a queue X (cf. line 23), both sorted temporally. For the special case of labels being immediately available ($\delta = 0$, $v = t$) and continuous-training-type regressor models being used, queues X and Y are redundant, and $\{\mathbf{X}_t, \vec{y}_t\}$ are batches of associated pattern-label pairs arriving at points in time t.

Supervised member development stipulates that each of E^{est}'s members is trained on a separate data window with fixed start (cf. line 25), thus on a separate time range. This induces diversity by representing separate concepts along the stream. In this regard, training can be done in an e.g. continuous or batch-continuous fashion. The windows dynamically extend to maximally κ data points, depending on the availability of values y_v mapped to the associated temporally sorted $\vec{x}_v$. For s, the window is initiated with its pre-deployment training dataset. If its size is smaller than κ, the window grows after deployment and training continues.
Therefore, assuming a worst-case training complexity $O(f^{est}(\kappa))$ for the employed regressor, a fully occupied E^{est} still maximally has polynomial training complexity $O((1 + n^{est}_{max}) \cdot f^{est}(\kappa))$. Data windows are implemented by references to certain ranges within data and label queue X and Y. The oldest unreferenced ranges in these are regularly cleared to maintain memory efficiency. As represented by line 24 und 25, E^{est} is evaluated in a prequential manner, i.e. all members are separately scored on unseen data before being trained on it. Training is optionally preceded by any required data preprocessing steps.

Finally, E^{est}'s members are reweighed (cf. line 26). At every t that includes training and a $|\mathbf{R}| > 0$, the sum of all $w^{r_i}_t$ is computed before deriving individual member weight values. In the first post-deployment round, the sum value defaults to the accretion rate a, which generally is a small positive real value. In subsequent rounds, it is computed by multiplying its previous value, i.e. at time u, with $1.0 + a$, as formalized in (7.8). The stable member's weight w^s_t is computed by decreasing its previous value w^s_u accordingly to keep the sum of all member weights at 1.0 at all times. This follows the rationale that a concept valid before a concept drift is preserved by s but loses importance over time represented by its decreasing weight. To ensure that s is never entirely superseded by the r_i, a lower bound w^s_{min} permanently applies for w^s_t. This is formalized in (7.8) and (7.9).

$$\sum_i w^{r_i}_t = min((1.0 + a) \cdot \sum_i w^{r_i}_u, 1.0 - w^s_{min}) \quad \text{with } u < t \tag{7.8}$$

$$w^s_t = 1.0 - \sum_i w^{r_i}_t \quad \text{by implication } w^s_t \geq w^s_{min} \text{ with } u < t \tag{7.9}$$

Each $w^{r_i}_t$ is then recomputed as a share of the aforementioned sum proportional to its relative experience. In this regard, experience is quantified as the number of data points seen by an r_i until t during training. Consequently, reweighing requires no measurement of a member's performance. Given that all members are maximally trained on κ data points, during long periods without concept drift, the $w^{r_i}_t$ converge to an equal share of the maximum sum over all $w^{r_i}_t$, i.e.

$$w^{r_i}_\infty = \frac{1 - w^s_{min}}{|\mathbf{R}|}. \tag{7.10}$$

If $|\mathbf{R}| = 1$, trivially, $w^{r_1}_t$ directly follows from (7.8) as it is assigned the sum value. In cases of $|\mathbf{R}| > 1$, the individual weight growth of the youngest reactive member r_{-1} is amplified to accelerate its ensemble-global importance gain. As a result, E^{est}'s ability to quickly adapt to new concepts is also amplified. In the first reweighing round after this member's instantiation, $w^{r_{-1}}_t$ is initialized with the value of a. In subsequent rounds, it is computed as

$$w^{r_{-1}}_t = min(w^{r_{-1}}_u \cdot (1.0 + a), w^{r_i}_\infty) \quad \text{with } u < t. \tag{7.11}$$

In other words, $w^{r_{-1}}$ is subject to direct weight accretion while being capped by the value each r_i converges to. This compensates for its possibly, with respect to the older r_i, relatively little experience determining a quite small share of the $\mathbf{R}$-total weight accretion. In contrast, this amplification is not required for the case of $|\mathbf{R}| = 1$, as the single r_1 already achieves fast growth by not sharing the $\mathbf{R}$-total weight accretion with other r_i.

This amplification of the youngest reactive member's weight growth is done before the remaining $w^{r_i}_t$ are computed in that round. The latter are then assigned their respective experience-based share of the weight sum's remainder resulting after subtracting $w^{r_{-1}}_t$.

The afore-described weighing-based means aim at e.g. representing member experience, easing-in, diminishing and protecting the minimal importance of concepts. Therefore, they contribute to the research goal of maintaining data stream concepts of **RQ V-2**.

Evolutionary Concept Drift Adaptation

Nature's ways of solving fitness problems can offer a valuable basis for imitation via e.g. construct properties and algorithmic mechanisms. The proposed method takes an evolutionary-biological understanding of naturally occurring processes as template. Among other aspects, these templates are reflected by ensemble member attributions. In particular, the stable member s represents a knowledgeable individual of E^{est}'s population. As described above, it can be augmented by instantiating r_i that can be interpreted as offspring individuals of s.

Member rotation, which is done before training and reweighing, is another instance that implements the aforementioned template. This mechanism, with all of its components, addresses both of the primary research goals raised by **RQ V-2**—to maintain and discard data stream concepts. It is applied upon each change detection of D and adapts E^{est}'s population as follows: As long as the current $|\mathbf{R}| < n^{est}_{max}$, a new r_i is instantiated and simply added to $\mathbf{R}$ (cf. line 21).
Otherwise, E^{est} undergoes member rotation based on natural selection principles. For these, the maturity of members needs to be considered. An r_i reaches maturity once it has been trained on at least μ data points, with necessarily $\mu <= \kappa$.
If $\mathbf{R}$ contains mature r_i, they challenge s for its stable member status, indicated by lines 11 to 16 of Algorithm 1. A challenge's victor is determined by the lowest MAE computed over its entire lifetime to prevent overfitting on recent samples. Two feasible cases need to be distinguished:

a. If an r_i is victorious, it assumes the stable member role. The prior s and the likely outdated concept it represents are discarded from E^{est}. The new s then receives the sum of its existing $w_u^{r_i}$ and w_u^s of the prior s.

b. If the current s is not defeated, the oldest r_i is dropped from $\mathbf{R}$ (cf line 15). In that case, the remaining r_i each receive a share of its weight proportional to the ratio of their respective $w_u^{r_i}$ and the sum over all $w_u^{r_i}$.

If there are no mature r_i to begin with, the action of case b. is executed, too (cf line 18). This is done to satisfy the requirement that all member weights sum up to 1.0 at all times. Based on these mechanisms, high-performing members are more likely to be kept.
After natural selection, a new r_i is always instantiated and added to $\mathbf{R}$ to represent more recent concepts in the stream (cf. line 21 again). Its w_u^{r-1} is initialized with 0.0.

For the sake of completeness, E^{est} can be assigned a full parametrization $\theta^{E^{est}}$. It is formalized as a set

$$\theta^{E^{est}} = \{f^{est}, n^{est}_{max}, \mu, \kappa, a, w^s_{min}\}, \tag{7.12}$$

with its elements being introduced throughout the paragraphs above.

Algorithm 1 Application of the proposed method

```
Input: data stream $\{\mathbf{X}_t, \vec{y}_t\}$, parametrization $\theta^{E^{est}}$
Output: estimation vectors $\hat{\vec{y}}_t$

$s \leftarrow$ initializeStableMember($f^{est}$)
$\mathbf{R} \leftarrow \emptyset$
$E^{est} \leftarrow$ initializeEnsemble($s, \mathbf{R}$)
$D \leftarrow$ initializeChangeDriftDetector()
$X, Y \leftarrow$ initializeDataQueue(), initializeLabelQueue()
while true do
    $\hat{\vec{y}}_t \leftarrow E^{est}$.predict($\mathbf{X}_t$)
    $change \leftarrow D.detectChange(\hat{\vec{y}}_t, Y)$
    if $change == true$ then
        if $|\mathbf{R}| == n^{est}_{max}$ then
            if $\mathbf{R}$.containsMatureMember($\mu$) $== true$ then
                if $\mathbf{R}$.challengeStableMember($s$) $== success$ then
                    $s \leftarrow \mathbf{R}$.popVictoriousReactiveMember()
                else
                    $\mathbf{R}$.dropOldestMember()
                end if
            else
                $\mathbf{R}$.dropOldestMember()
            end if
        end if
        $\mathbf{R}$.addReactiveMember($f^{est}, a$)
    end if
    $X$.queueData($\mathbf{X}_t$), $Y$.queueLabels($y_v, \hat{\vec{y}}_t$)
    $E^{est}$.score($Y$)
    $E^{est}$.fit($X, Y, \kappa$)          ▷ Queues allow for training on $\kappa$ points and delayed labels
    $E^{est}$.updateWeights($a, w^s_{min}$)
end while
```

7.3.3 Practical Integration

For an effective application of the proposed method in industrial data stream scenarios, several practical aspects must be considered prior to and after deployment. These are elaborated on in the following paragraphs.

Pre-deployment considerations. A sufficiently large labeled historical dataset should be available to train a suitable initial stable member s. Despite s being trained on merely a small amount of data, the proposed method can still be applied in a feasible manner. However, its regression performance might be limited until E^{est} has been provided with more data. With a large body of data being available, one might differ between two cases regarding its utilization:

a. It is a valid assumption that the data does not contain any concept drifts.

b. One cannot rule out the existence of concept drifts in the data.

If case a. applies and f^{est} has a sufficiently high capacity to model the variance in the data (Mitchell 1997, pp. 214–217), s can be trained directly on the entire dataset. One might argue that this case's boundary conditions also legitimize training on more than κ data points. Doing so enables the stable member to represent a highly reliable reference concept during early post-deployment stages.
If case b. applies, one might synthetically treat the dataset as a stream, i.e. bring forward the deployment. The data can then be passed to the method in batches of size b while invoking its full functionality in a strictly supervised fashion. This way, one may avoid contaminating E^{est} with diluted or outdated concepts.

Parametrization. The proposed method's behavior can be finely tuned via its parametrization $\theta^{E^{est}}$; cf. (7.12). The parameter n^{est}_{max} controls an ensemble's capacity to represent multiple simultaneously valid concepts. While smaller values restrict this capacity, higher ones may increase it, however, at the cost of increased computational demands.
Regarding the maturity threshold μ, as already introduced above, $\mu \leq \kappa$ must hold to fundamentally enable maturation within an r_i's lifespan. Additionally, it is worth noting that increasing the difference between both values is a feasible way to enable member training being continued even after maturity. Furthermore, increasing κ is advisable, if concepts are expected to prevail for extended time spans or are difficult to learn, i.e. feature high data variance. Nevertheless, again, this requires sufficient computing capacity being available.
Low values of the accretion rate a (e.g. 0.01) help mitigate the impact of under-fitted members, while higher ones facilitate faster adaptation to novel concepts. The latter may especially be advantageous in quickly evolving data stream scenarios.
Careful tuning of the stable member's minimum weight w^s_{min} is also required to control E^{est}'s stability. If the negative impact of concept drifts on regressor performance is expected to be low, one can choose a high value. This way of fostering stability can additionally be amplified by pairing it with setting low values for a to further preserve the concept knowledge of the stable member and to ensure it contributing strongly to the ensemble-global output. If the impact is expected to be high, the inverse is recommended.
In corner cases implicating the assumption of the initial concept represented by s being valid indefinitely, one might choose to omit it being challenged by the r_i altogether. Then, natural

selection mechanisms would be narrowed down to dropping the oldest r_i on concept drift detections. This would, consequently, still enable handling inferior concepts that may be temporarily valid in addition to the indefinite one.

Forgetting strategies. In the field of continuous learning, forgetting can be described as a means of implementing estimator plasticity. Within the context of ensembles, concrete methods revolve around modifying or cycling members and thereby the memory of the data-bound concepts they represent (Jaber et al. 2013). Figure 7.8 additionally depicts a taxonomy based on dynamic properties of such methods.

The proposed method implements forgetting strategies within several mechanisms to primarily address the goal of discarding concepts of **RQ V-2**: Dynamic weighing of the stable and reactive members with decreasing weights representing the partial forgetting of concepts, discarding of the oldest reactive member for full abrupt forgetting of concepts, and initiating challenges against the stable member to reinforce the memory of more recent concepts while forgetting possibly outdated ones.

The author of this thesis additionally defined a project with the goal of utilizing the proposed method's parametrization to realize various degrees of forgetting within different scenarios. This project was carried out by undergraduate student Ott (2024). He found that exercising abrupt forgetting by dropping members has great potential for fast concept drift adaptation and pointed out the increased capacity of ensembles with many members to maintain and version recurring concepts.

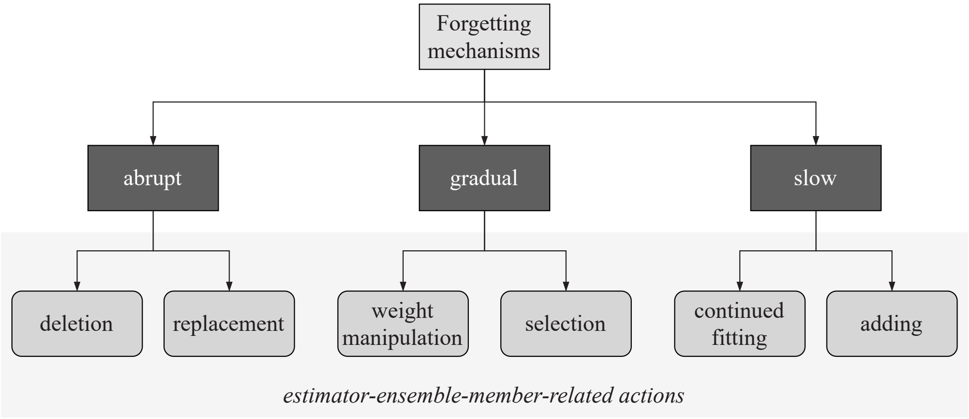


Figure 7.8: Dynamic properties of various forgetting strategies in estimator ensembles for data stream scenarios. The last row indicates various concrete measures to exercise fine-grained control of ensembles' forgetting capabilities via their member sets. Figure based on Ott (2024).

Transferability. While the proposed method is presented within the context of a specific estimator ensemble, it can, nevertheless, be transferred to other ensemble-based concept drift adaptation methods. While mechanisms like the member rotation and reweighing can straightforwardly be applied to different ensemble architectures, the method requires adaption for working with different problem types, such as classification.

Other aspects. The proposed method assumes delayed availability of ground truth for supervised training of ensemble members and concept drift detection, in case a supervised detector is selected by the user. Nonetheless, not all components require ground truth. In particular, the experience-based member reweighing operates without depending on member performance information and is therefore applicable in semi-supervised or unsupervised scenarios. This mechanism follows the rationale that models improve their performance via continuous exposure to new data. As a result, their weights increase for greater influence regarding the ensemble-global estimation.
Also, a range of preparatory processes are not specified by the proposed method and thus left open for the user. These include the initial selection and hyper-parameter tuning of model f^{est} as well as the concrete design of preprocessing pipelines transforming data before being passed to ensemble members or the concept drift detection solution.

Outlook. An interesting avenue for future research lies in dynamically adjusting the parameters $\theta^{E^{est}}$ during runtime, to better handle possibly evolving characteristics of concept drifts. For instance, in response to different dynamic profiles, the parameters could be adjusted to accelerate or decelerate the responsiveness of adaptation measures. This might further enhance the method's applicability in volatile industrial environments.
The method supports mixtures of members employing batch and continuous-training-type models. If deemed suitable, the latter ones could be trained indefinitely, i.e. not bound by κ.

7.4 Composing a Method from Concept Drift Detection and Adaptation Components

This section is dedicated to the consolidation of the novel methods for concept drift handling introduced earlier in this chapter. By doing so, it proposes the equally novel stream-applicable method for concept drift adaptation in regression scenarios (SAM-CDAR). The modular design of SAM-CDAR is based on architectural principles of robust inference process systems. This as well as further remarks on its rationale and application are elaborated on in the following.

7.4.1 Rationale and Motivation

In the preceding Sections 7.2 and 7.3, two ensemble-based methods are introduced that each leverage different sources of diversity: The method E^{det} fosters diversity by combining different concept drift detectors and accounts for complex concept drift dynamic profiles and types whereas E^{est} fosters diversity among regressors along the time axis in a nature-inspired fashion to adapt well to concept drifts.

This section proposes the composite method SAM-CDAR. It combines these two aforementioned methods to achieve both detection and treatment of concept drifts with a high performance potential. The explicit-concept-drift-adaptation aspect of E^{est} is preserved and internalized in SAM-CDAR to further address the associated and still prevailing research gap (Krawczyk et al. 2017). Also with respect to E^{det}, this aspect counters the lack of available research on CDDEs being employed for concept drift adaptation as discussed in Section 6.3. A schematic overview of SAM-CDAR is depicted in Figure 7.9.

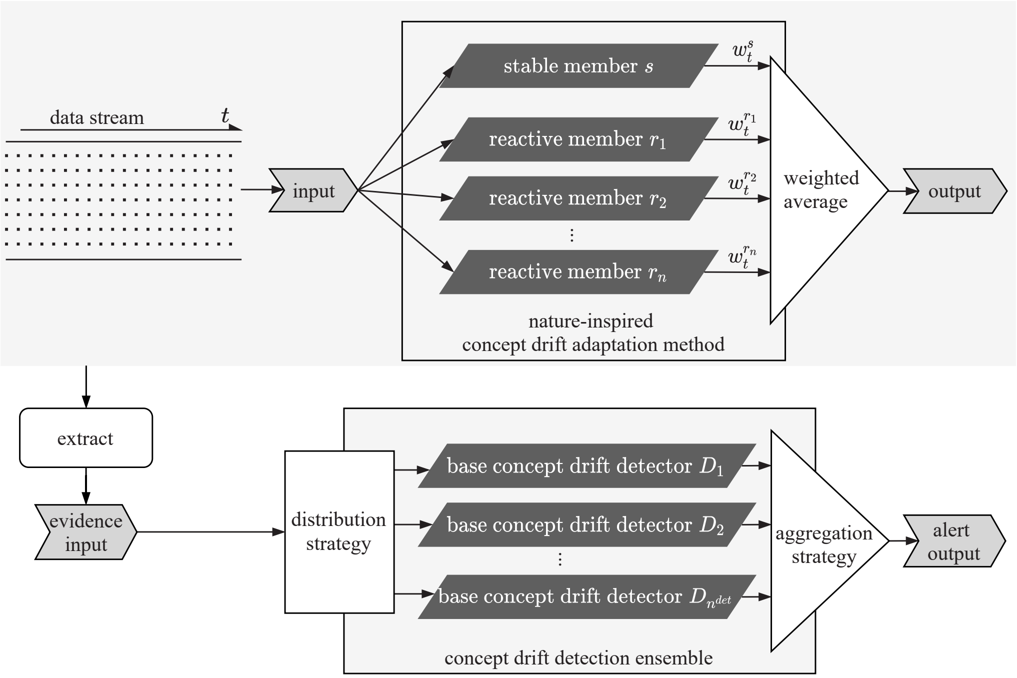


Figure 7.9: The proposed stream-applicable method for concept drift adaptation in regression scenarios (SAM-CDAR) composited from the methods introduced Sections 7.2 and 7.3. It combines their respective strengths to monitor for and handle concept drift occurrences in the data streams of regression problems.

7.4.2 Design

Essentially, the proposed method SAM-CDAR is a combination of two ensemble methods introduced in earlier sections of this chapter. Its design, which follows robust-inference-process-system principles, as well as the specific composition of its components are outlined in the following paragraphs.

Reference-Architecture-based Approach

The concept and design of SAM-CDAR is supported by the reference architecture for robust inference process systems, which is introduced in Section 5.5 and visualized in Figure 5.6 as well as in figures throughout Section 5.5.2. As this reference architecture constitutes a template, SAM-CDAR results from specializing it, i.e. specifying several of its subjects and messages. Doing so results in an architecture model with an SID presented in Figure 7.10. The respective SBDs correspond to the ones introduced in Section 5.5.2. Therefore, this approach provides a response to **RQ V-3**.

The main of the aforementioned specifications concern the inference unit and the concept drift detection unit. The former is specified as the method E^{est} and processes input data and outputs regression value predictions for further use by consuming entities. Any workloads related to retraining its members are executed within the context of the model (re-)training unit, i.e. the compute capacities abstracted by it. These are requested by the concept drift adaptation unit, which represents evolutionary mechanisms and algorithmics of E^{est}. The latter is specified as the method E^{det}. It receives error score values as input from the evaluation unit as defined by its aggregation strategy S^{det}_{dist}. Its outputs are in turn consumed by the concept drift adaptation unit.

Further Design Remarks

While E^{est} could technically be employed with any concept drift detection approach, i.e. also using mere base detectors, SAM-CDAR employs E^{det} as its detector. In SAM-CDAR's proposed form, exclusively change-level alerts of E^{det} are considered, despite it being designed to also aggregate warning-level alerts as explained in Section7.2. This design choice is motivated by the goal to focus concept-drift-detection-triggered adaptation of estimator ensembles to rather drastic reset-like operations. Such are specified by the ensemble development and evolutionary mechanisms of E^{est}, which foster and leverage the estimation capabilities and diversity of a continuously trained estimator ensemble.

Lower-magnitude alerts, which include warnings, are often utilized as triggers to e.g. newly

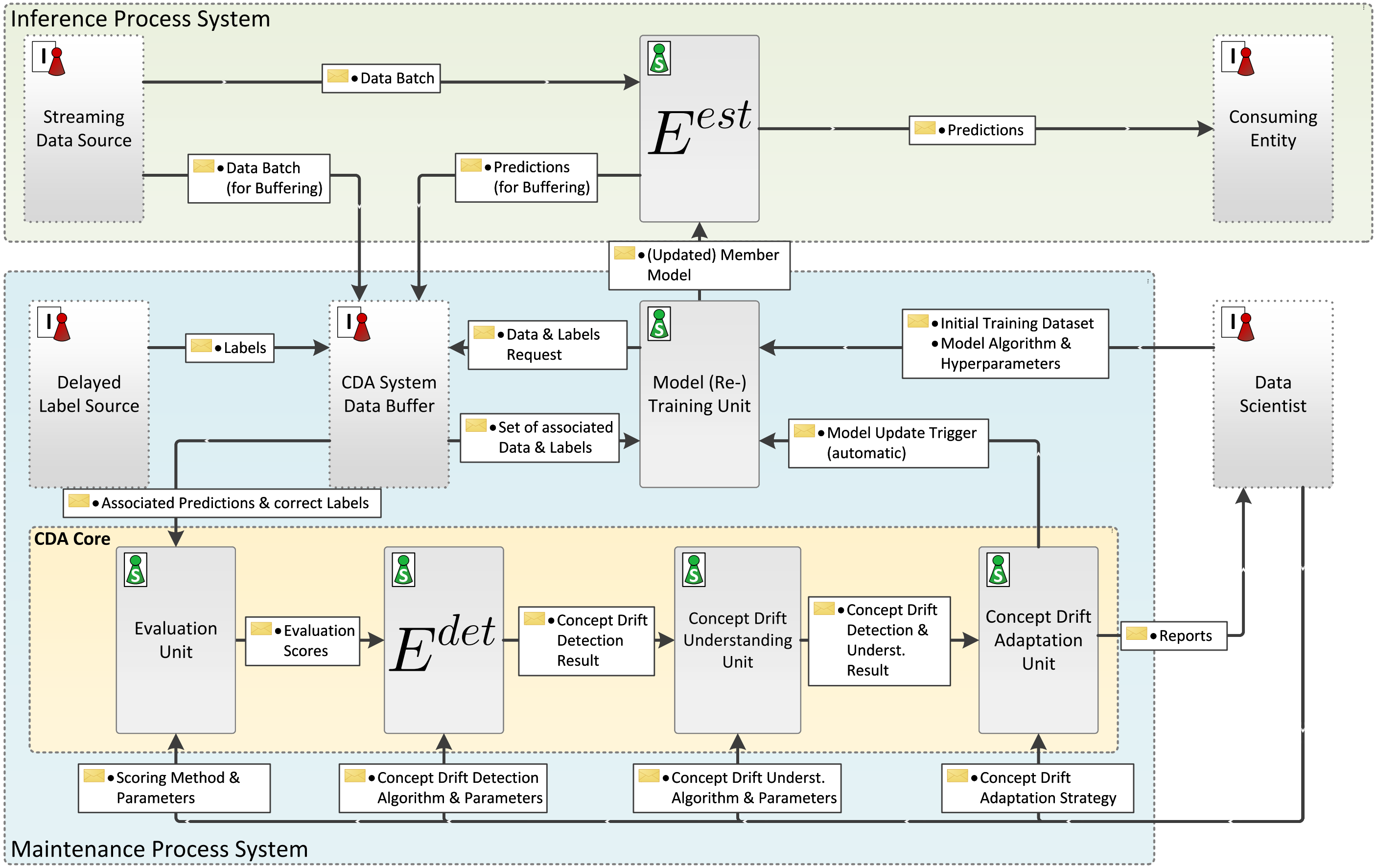

Inference Process System
Streaming Data Source
•Data Batch
E^{est}
•Predictions
Consuming Entity
•Data Batch (for Buffering)
•Predictions (for Buffering)
•(Updated) Member Model
Delayed Label Source
•Labels
CDA System Data Buffer
•Data & Labels Request
Model (Re-) Training Unit
•Initial Training Dataset
•Model Algorithm & Hyperparameters
Data Scientist
•Set of associated Data & Labels
•Model Update Trigger (automatic)
•Associated Predictions & correct Labels
CDA Core
Evaluation Unit
•Evaluation Scores
E^{det}
•Concept Drift Detection Result
Concept Drift Understanding Unit
•Concept Drift Detection & Underst. Result
Concept Drift Adaptation Unit
•Reports
•Scoring Method & Parameters
•Concept Drift Detection Algorithm & Parameters
•Concept Drift Underst. Algorithm & Parameters
•Concept Drift Adaptation Strategy
Maintenance Process System

Figure 7.10: The subject interaction diagram (SID) of the architecture model of the proposed stream-applicable method for concept drift adaptation in regression scenarios. It results from specializing the reference architecture template introduced in Section 5.5, with its SID being visualized in Figure 5.6. The main specifications result in the methods E^{est} and E^{det} being employed as inference and concept drift detection units, respectively. The subject behavior diagrams (SBDs) presented in figures throughout Section 5.5.2 complete its Subject-Oriented architecture description.

instantiate and begin the training of background estimators. As one might argue that these represent rather modest ensemble-modifying operations, they are not exploited by SAM-CDAR (following Woźniak 09.10.2023).

7.4.3 Practical Integration

In the following, a set of remarks on how to use SAM-CDAR effectively shall be provided. Further considerations, e.g. the parametrization as well as limitations of its main components E^{est} and E^{det}, can be obtained from previous sections of this chapter.

Further design specialization. The architecture model of SAM-CDAR, which is displayed in Figure 7.10, is a result of specialization from a reference architecture template as explained above. However, it can still be further specialized to concrete application problems if needed. For instance, human roles, other than the exemplarily described data scientist, can be integrated as desired, which is also outlined in Section 5.5.4. The delayed label source models ground truth becoming available with arbitrary delay. The concept drift adaptation system data buffer compensates for this aspect and gathers data and labels until they are usable for training or concept drift detection. In cases of the delay being infinite, i.e. ground truth being unobtainable, labels need to be manually provided to gain access to a range of functions of SAM-CDAR as outlined in previous sections. The concept drift understanding unit constitutes an optional component to foster insights into concept drift occurrences.

Compute capacity. The method SAM-CDAR is a composite approach of two machine learning ensemble methods. Consequently one might argue that this results in relatively high time and memory complexity traces when compared with the utilization of a base estimator and concept drift detector. However, as argued by Read and Žliobaitė (2023), the environments of real-world practical scenarios often provide sufficient compute capacity for such a rather extensive use of ensemble methods. On the contrary, such may even be under-used with scale-ups presenting economically feasible options. This is confirmed by the author of this thesis based on observations gathered during various industry research projects regularly involving German small and medium-sized enterprises.

7.5 Summary

This chapter, guided by research question **RQ V**, introduces a range of novel methods for handling concept drifts and sustaining machine learning regressor robustness in data streams. The technical as well as algorithmic details of these methods are explained and supplemented with remarks regarding the constraints of their use in real-world industry scenarios. Integral aspects of this thesis concerning robust inference process system design as well as detecting concept drifts are brought together by building on and synthesizing insights from Chapters 5 and 6, respectively.

As this chapter's core contribution to science, the stream-applicable method for concept drift adaptation in regression scenarios (SAM-CDAR) is proposed. It is a composition of a concept drift detection ensemble (CDDE) as powerful tool to detect robustness-harming concept drift occurrences and an ensemble approach allowing the model-algorithm-independent adaptation of machine learning regressors. In the subsequent Chapter 8, this method is subject to experimental evaluation on data from various real-world industry scenarios.

With regard to the research on CDDEs in general, SAM-CDAR is a method advancing its field. Based on a literature analysis conducted in Chapter 6, this can be argued as the majority of works on CDDEs focus on their concept drift detection performance. In contrast, SAM-CDAR places emphasis on employing them as core tool for concept drift adaptation. Additionally, the above-mentioned ensemble approach for maintaining machine learning regressors constitutes a contribution to research promoting the continued usability of batch-based machine learning algorithms in data stream scenarios.

8 Evaluation

The preceding Chapter 7 introduces the novel stream-applicable method for concept drift adaptation in regression scenarios (SAM-CDAR), which represents the core contribution of this thesis. This chapter continues from there and is dedicated to the careful practical validation of this method and its components. It takes place amidst real-world industrial scenarios obtained from various research projects. They are characterized by the occurrence of concept drifts harming the performance of applied machine learning regressors.

The outline of this chapter begins with Section 8.1 operationalizing the research question that is conceptualized to scientifically ground the above-mentioned practical validation. This validation involves an experimental approach, which is additionally visualized in Figure 8.1. The associated real-world industrial contexts are characterized in Section 8.2. Section 8.3 outlines the experiment setting designed to evaluate the proposed method SAM-CDAR and its components. The respectively observed results are discussed in Section 8.4.
In parts, Sections 8.3 and 8.4 may contain verbatim passages from a work previously published by the author of this thesis (Trat et al. 2024). These are not individually cited when the author is responsible for the intellectual content and research insights. Contributions from co-authors as well as figures and tables are explicitly cited.

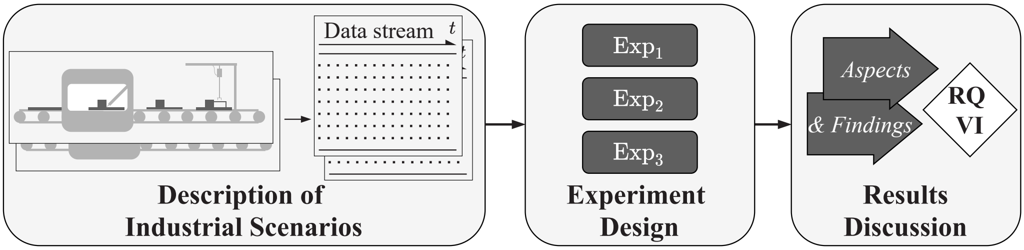


Figure 8.1: The experimental evaluation approach of this thesis. It is segmented into the description of the real-world industrial contexts, the definition of conducted experiments (Exp$_i$) and the discussion of obtained results guided by research question **RQ VI**.

8.1 Research Questions

This chapter revolves around research question **RQ VI** that investigates the extent to which the proposed method SAM-CDAR sustains the robustness of productive machine learning regressors exposed to real-world scenarios. To make this overarching scientific approach tractable, it is divided into three sub-questions that reflect complementary evaluation perspectives:

RQ VI To what extend does the proposed method sustain the robustness of productively applied machine learning regressors in real-world evolving industrial data streams?

- **RQ VI-1** How does the performance of the proposed method compare to relevant baseline approaches with respect to regression error and concept drift adaptation capability?
- **RQ VI-2** To what extent do individual components of the proposed method, and variations in their parametrizations, contribute to sustaining robustness?
- **RQ VI-3** What is the impact of employing machine learning ensembling within the proposed method?

RQ VI-1 and **RQ VI-2** both point towards an evaluation of SAM-CDAR's performance potential, demanding the analysis of concrete metrics. While the former focuses on a comparison with realistic and popular baselines, the latter shifts the perspective inwards, asking to what extent the method's individual components and their parametrizations contribute to achieving the states goal.

Careful attention to the principle of machine learning ensembling, which is employed across multiple components of the proposed method, is additionally paid through **RQ VI-3**. It seeks to identify resulting potentials for coping with concept drifts as well as practical guidance for further development.

8.2 Description of Industrial Scenarios

In this section, two scenarios from real German small and medium-sized enterprises are outlined. Both share the common motivation to achieve a long-term-robust regression performance for estimation problems involving industrial manufacturing and logistics-related processes. While the respectively gathered data are contained as complete sets, they are characterized as streams in their original environment, i.e. data points are produced in a continuous and evolving fashion. The properties of these streams are compared in Table 8.1.

8.2.1 Lead Time Estimation

A company offering products in a make-to-order fashion contributes an industrial lead time (LT) case study. Having more than 200 employees, it can be characterized as medium-sized. It manufactures a wide spectrum of freeform forging parts for various types of high-tech machinery, power generation and other domains.

Associated product orders may be placed on short notice and need to be produced quickly, occasionally even with increased priority. Nevertheless, the company attaches great importance to delivering products on time. To achieve this, the estimation of LTs is required to be as precise as possible to enable accurate production planning. In this context, in abstract terms, a LT quantifies the duration of manufacturing a product from its company-internal registration until its completion. It can, for instance, be expressed as the sum of the individual durations of all involved manufacturing process steps (Bender 2024, Bender et al. 2022). A schematic visualization of a feasible exemplary process and its LT is provided in Figure 8.2.
Initially, the LTs were manually or heuristically estimated once by experienced personnel without further updates. Since these values were quite imprecise and costly to obtain, the team around the thesis author contributed to a research project to introduce the machine-learning-regression-based estimation of process step durations. This approach now offers significantly higher LT estimation accuracy leading to an improved planning quality (Bender 2024, Bender et al. 2022). It also enabled the company to identify bottlenecks and to put in appropriate countermeasures (Gramespacher et al. 2022).

Despite the project's success, analyses conducted by the thesis author show that this LT estimation approach can heavily be affected by concept drifts. One might argue that this is primarily due to e.g. raw material availability and personnel attendance rate fluctuations, which are possibly influenced by the COVID-19 pandemic, seasonality effects and failures as well as the procurement of new machines. Even other possibly more subtle changes might cause similar effects and lead to the performance decline of employed regressors. It can therefore be hypothesized that appropriate concept drift adaptation measures are highly advisable to counteract this (Bender et al. 2025).

The data utilized as input for the regressor to estimate the duration of the 22 different manufacturing process step types is gathered from the company's enterprise resource planning system. Its collection method was composed within the course of the above-mentioned research project (Gramespacher et al. 2022) to capture all relevant information. The steps comprise production-related ones, e.g. forging, sawing, lathing, milling, drilling and quality assurance, as well as packaging-related ones. The data amounts to approx. 100k data points, spanning two years from January 2019 to February 2021, and is characterized by a set of 21 features. It includes nine categorical and twelve numerical process-related features describing raw material properties, personnel attendance and organizational-planning-related information, and is listed in full in Table A.3 in the appendix of this thesis. In its data stream form, 5–10 batches of size b equal to 25 each typically arrive per day (Gramespacher et al. 2022) (cf. Table 8.1).

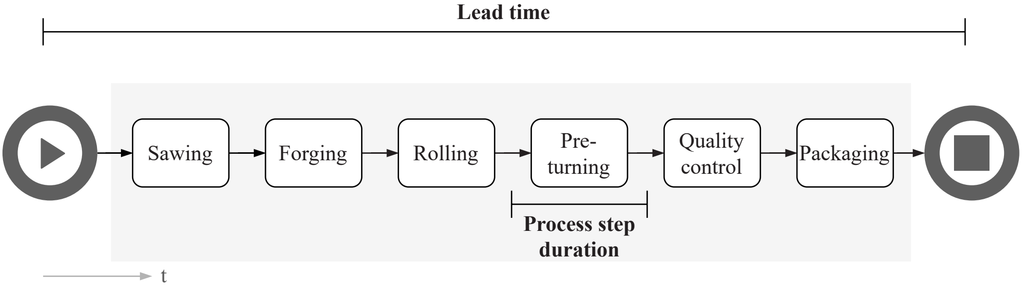


Figure 8.2: The lead time and step durations of a feasible metalworking process (Bender 2024).

8.2.2 Delivery Time Estimation

Another case study is contributed by a company that, having approx. 50 employees, can be characterized as a small to medium-sized one. As a data and process integrator, this company supports a large range of others of the furniture and automotive trade in their diverse innovation affairs. This includes, for instance, digitization projects, data-driven analyses and business model development.

In the specific case considered by this thesis, the company fully handles a large volume of product orders for a large number of actors in the furniture trade. This, among other aspects, involves the processing of manufacturing orders, dispatching and handover-supporting services for retailers, wholesalers, logistics service providers and manufacturers. The case's core issue of estimating the

delivery time (DT), i.e. the duration between a product's order until its delivery at the customer's site, was identified as one that is of great importance for these actors. Possibly one of the primary reasons for this is that consumers are known to react highly sensitively to changes in DTs of furniture products (Marino et al. 2018). Nevertheless, the issue remains largely unresolved across the industry, as emerged from discussions conducted within a research project (Sönke and Trat 2025) involving the company, a diverse board of trade representatives and the team around the author of this thesis. The representatives regard access to DTs as even offering a wide range of additional potentials and benefits for their various concerns: These include but are not limited to enhanced customer satisfaction and reduced inquiries via notice of estimated DT, inventory and production planning optimization, data-driven decision making, as well as improved adaptability to market and trend volatility (Sönke and Trat 2025).
Therefore, a machine-learning-regressor-based DT estimation approach was developed during the project. It could already be anticipated that this approach is likely to be affected by concept drifts, arguably due to raw material availability and demand fluctuations, especially with respect to timber, seasonality effects, and market trends. Consequently, the development already integrated a range of robustness-preserving aspects from the beginning (Sönke and Trat 2025).

Based on their well established data-lake-based infrastructure as well as standing in the furniture trade, the company could already draw on a structured data gathering approach that can potentially be made available as a continuous data stream. In contrast to the LT case study, this data only implicitly contains information on manufacturing, but primarily on logistics-related processes as indicated by Figure 8.3. Also, it represents activities involving a large number of different companies instead of merely one and thus captures patterns across the trade.
The case study is based on approx. 1.5M data points characterized by a set of eight features, divided into five numerical and three categorical ones. These include information on ordered products, such as quantity and price, as well as on actor locations and are listed in full in Table A.4 in the appendix of this thesis. The data spans 1.5 years from January 2022 to June 2023. As a data stream, 5–10 batches of size b equal to 250 each arrive per day (Sönke and Trat 2025) (cf. Table 8.1).

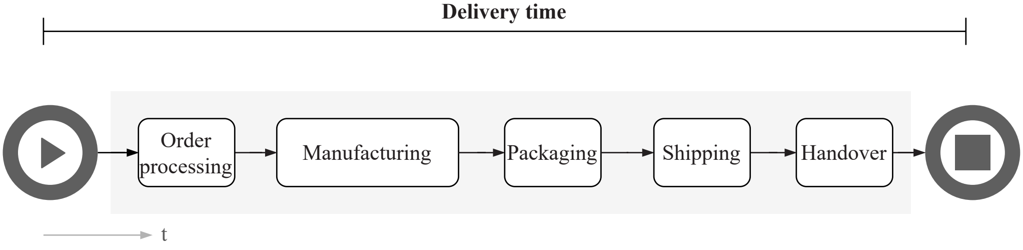


Figure 8.3: The delivery time of a product traversing through a value creation network of the furniture trade (Sönke and Trat 2025).

Table 8.1: A comparison of the data streams of the lead time and delivery time case study.

	Lead time	Delivery time
Number of datapoints	100k	1.5M
Number of features	21	8
Spanned time horizon	2 years	1.5 years
Number of batches daily	5–10	5–10
Batch size	25	250
Estimation goal	Process step duration	Delivery duration
Context	Company-internal manufacturing processes	Trade-wide product orders

8.3 Experiment Design

The experiments conducted within the context of this thesis are designed based on research question **RQ VI** and its sub-questions. This yields a series of experiments Exp$_i$ validating methods proposed in this thesis with respect to their long-term robustness in practically relevant regression tasks by following a case-study-like schema. The data for these are obtained from the industrial scenarios outlined above in Section 8.2. In this section, their setting and context are outlined. If not indicated otherwise, they follow the methodic principles introduced in Section 2.3.

Fundamentally, the Exp$_i$ represent steps of a sequential optimization and evaluation approach evaluating SAM-CDAR by gradually expanding its functionality from a component to a full-system level. Firstly, the critical adaptation component E^{est} featuring a simple concept drift detector is compared with popularly employed baseline approaches in Exp$_1$ and Exp$_2$. Secondly, an optimized E^{est} is augmented by E^{det} to form the full method SAM-CDAR, which is then applied throughout Exp$_3$. This approach therefore measures not only the final overall performance of the composite method SAM-CDAR but facilitates separate analyses of its components and their respective contribution. It is additionally presented in Figure 8.4.

In the following, general settings applying to all Exp$_i$ are laid out initially. Subsequently, experiment-specific details are expanded upon.

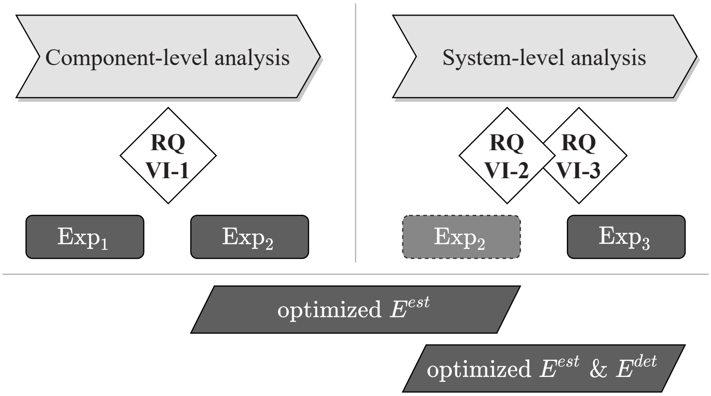


Figure 8.4: The sequential optimization and evaluation approach. It stipulates an isolated component-level evaluation of the concept drift adaptation method E^{est} initially. In its optimally parametrized configuration, it is then employed alongside E^{det} for a full system-level evaluation determining the latter's optimal parameters and performance potential. This approach is designed to gather evidence for the research questions **RQ VI-1** through **RQ VI-3**.

8.3.1 General Settings

In the following, several design decision, parameter values and assumptions that are valid across all experiments are outlined.

Preprocessing

The industrial real-world data considered in this thesis is characterized by features that divide into categorical and numerical ones. While the latter can directly be processed by the estimators employed in this thesis, the former require transformation. For simplicity's sake, categorical features are incrementally label-encoded before being processed further, e.g. by estimators or concept drift detectors.

Employed Regressors

Given that great importance is attached to comparability across all methods, tree-based model algorithms f^{est} are consistently employed as base regressors. Forest-type regressors, i.e. ensembles of trees, consistently hold a static number of members n^{est} equal to 10. This decision builds on insights from preliminary experiments. Although it cannot be precluded that varying this parameter might improve results, its influence lies outside the experiments' focus.

For any estimator to be able to yield any output, it needs to be trained on at least a small portion of data. In all experiments of this thesis, regressors are consistently trained on the 10k oldest data points before being deployed into the data stream. These data points are not included in the calculation of any performance metrics.

Further Remarks

As indicated by Section 2.3, the selected framework prequential evaluation assumes immediate access to labels. This also holds for training and concept drift detection approaches.
Given that this thesis places emphasis on testing proposed methods on real-world data, exact information on the location of concept drifts is generally unavailable. Without this information the direct evaluation of concept drift detectors, e.g. via scores introduced in Section 3.2.4, is impossible. In this regard, the prequential evaluation framework still provides a means to quantify the performance impact of concept drift adaptation measures without requiring any information on concept drift locations in the data (Lu et al. 2018).

The data streams of both industrial scenarios feature certain batch sizes b as explained in Section 8.2. Several experiments vary b to measure its impact on method performance. While their value cannot be increased, it can be decreased, which represents breaking up data batches into smaller portions.

The mean absolute error (MAE) is consistently averaged data-stream-globally and applied as primary evaluation metric. Given that all case studies focus on estimating durations, this metric choice is motivated by the MAE's property of providing errors expressed in the same unit as the data. Within the context of hyper-parameter optimization, it is employed as target metric with a minimization goal.
As another important evaluation criterion, the mean batch processing duration (MBPD) is computed as the duration of training and concept drift detection operations averaged over the entire stream's data batches. Comparing it across methods enables insights into their respective time complexity.

Implementations of the base concept drift detectors employed throughout the experiments are obtained from the Python framework for data stream mining River (Montiel et al. 2021).

8.3.2 Applying Baseline Approaches

In an initial series of experiment runs (cf. Section 2.3 for basic principles), specific established pre-existing approaches are evaluated on the stream data of both industrial scenarios. They are denoted as Exp_1. This is done to provide a benchmark for the methods proposed in this thesis as targeted by **RQ VI-1**.

Experimental Configuration

The tree-based approaches adaptive random forest regressor (ARFR), which is a specific ensemble of Hoeffding tree regressors (HTRs), and streaming random patches regressor (SRPR), which is an ensemble of regressors featuring continuous learning algorithms, are selected as baselines as they are popularly employed in practice. Both ensembles as well as the HTR are introduced in greater detail in Section 4.2.1 and 3.1.1, respectively. The SRPR could technically be equipped with n^{est} members featuring various f^{est} types. For comparability reasons, it is equipped exclusively with ARFR sub-ensembles, denoted as $\text{SRPR}_{\text{ARFR}}$, or HTRs, denoted as SRPR_{HTR}, within this thesis.

As concept drift detector, Adaptive Windowing (ADWIN) is employed. Details of this detector are provided in Section 3.2.4. This choice applies for the ARFR and the SRPR, which both feature both warning and change-level detectors. However, only change-level detectors are subject to optimization while warning-level ones are kept at default values. Its confidence is controlled via the parameter Δ and it receives the MAE (cf. Section 3.1.3), computed from the outputs of the regressors, as input.

As another baseline, the random forest regressor (RFR) (cf. Section 3.1.1), i.e. a non-adaptive version of the ARFR, is applied without being updated after deployment. In the following, it is denoted as $\text{RFR}_{\text{static}}$. While it represents a rather weak baseline, it reflects a fragment of the widespread industrial reality of under-maintained productive estimators (Vela et al. 2022).
A rather naïve approach to make the RFR robust is to continuously update it in an emulated-batch-continuous fashion as also outlined in Section 3.1.1. The practical feasibility of this update approach is severely limited as regressor training needs to be done each time from scratch on a dataset that is continuously expanded with every incoming data batch. As a result, the memory associated to this growing dataset tends towards infinity. Nevertheless, as the considered industrial scenarios have a finite size, it is evaluated as well and denoted as $\text{RFR}_{\text{naïve}}$.

Implementations of the continuous-learning-type baseline approaches are also obtained from River (Montiel et al. 2021). These, as defined by River, expect data to be passed with a batch size b of 1. For this reason, they are configured to process each received data batch in a point-wise fashion within Exp_1. The implementation of the RFR is obtained from Scikit-learn (Pedregosa et al. 2011). Being updated in a emulated-batch-continuous fashion, it can process data batches with arbitrary b.
As compute context, a workstation equipped wih an AMD Ryzen 9 3950X CPU and 64 GB main memory is utilized.

Parametrization

The parameters of these approaches are varied via hyper-parameter optimization as defined in Table 8.2. Unspecified parameters as well as those associated to the RFR$_{\text{static}}$ approach are kept at their default values.

Table 8.2: Parameter ranges for the experiments applying baseline approaches. Square brackets indicate sets of parameter values that are applied in separate experiment runs.

	ARFR	SRPR$_{\text{ARFR}}$	SRPR$_{\text{HTR}}$
Member f^{est}	HTR	HTR	ARFR
n^{est}	10	[1, 2, 3]	[10, 15, 20]
Δ*	[0.001, 0.004, 0.007, 0.0100]		
b	1		

* In the case of nested ensembles, only the parameter associated to the highest ensemble-hierarchical level is varied; sub-ensemble or member-associated ones are kept at their default values.

8.3.3 Applying the Proposed Nature-Inspired Concept Drift Adaptation Method

The nature-inspired concept drift adaptation method E^{est} is the component of SAM-CDAR that defines the algorithmic mechanisms for updating productive regressors in cases of concept drifts. Driven by the motivation behind **RQ VI-1** and **RQ VI-2**, this component shall first undergo isolated experimentation within the context of the sequential optimization and evaluation approach, as indicated in Figure 8.5. This is done in a series of experiment runs denoted as Exp$_2$.

Experimental Configuration

The technical details of E^{est} are introduced in Section 7.3. It is applied as robustness-sustaining approach replacing the mere and naïve use of RFRs. Nevertheless, it is evaluated with these regressors as members, denoted as E^{est}_{RFR}, as well as with ARFRs, denoted as E^{est}_{ARFR}. A hyper-parameter-optimization-based experiment, as outlined in Section 2.3, is conducted to measure its performance potential on the data of both industrial scenarios.

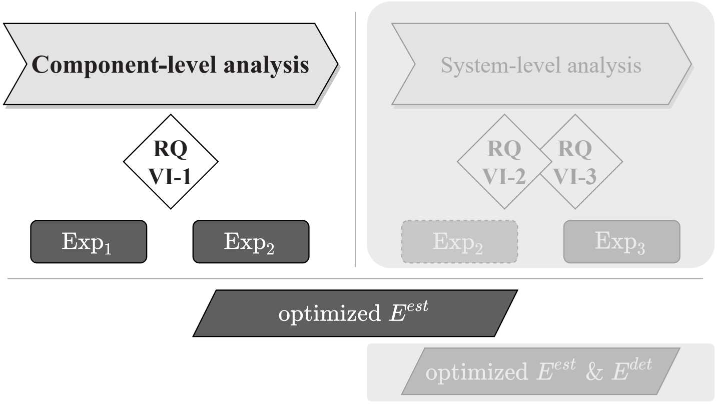


Figure 8.5: The component-level analysis of the sequential optimization and evaluation approach. It focuses on the evaluation of the concept drift adaptation method E^{est} initially. An optimally parametrized configuration is identified for its use in the subsequent analysis step. Being compared with baseline approaches, evidence for research question **RQ VI-1** is gathered.

E^{est} can process the concept drift alerts of any detector. In the first part of the sequential optimization and evaluation experimentation approach, however, it is applied with ADWIN as base concept drift detector. It takes as input the MAE computed from outputs of E^{est}.

Analogously to the baseline experiments, the same workstation equipped with an AMD Ryzen 9 3950X CPU and 64 GB main memory is utilized.

Parametrization

For E^{est}, those parameters of its full parametrization $\theta^{E^{est}}$ that are highly relevant for its evolutionary-biology-inspired mechanisms and the, in turn, resulting concept drift adaptation behavior, are varied: The stable member's weight minimum w^s_{min}, reactive members' accretion rate a and maximum training data volume κ. The maturity threshold μ and the maximum number of reactive members n^{est}_{max} are constants with value 1k and 3, respectively. ADWIN's Δ values are varied analogously to the baseline-related experiments. Also, the batch size b is varied within its applicable limits. Table 8.3 specifies all parameter ranges. Values of not explicitly mentioned parameters are kept at their defaults.

Table 8.3: Parameter ranges for the experiments applying the nature-inspired concept drift adaptation method E^{est} on data from the industrial scenarios lead time (LT) and delivery time (DT). Square brackets indicate sets of parameter values that are applied in separate experiment runs.

	E^{est}_{ARFR}	E^{est}_{RFR}
Member f^{est}	ARFR	RFR
n^{est}_{max}	3	
a	[0.01, 0.02]	
w^{s}_{min}	[0.2, 0.4, 0.6]	
κ	DT: [50k, 100k, 250k, 500k, 750k, 1M] LT: [50k, 100k]	
μ	1k	
Δ	[0.001, 0.004, 0.007, 0.0100]	
b	DT: [125, 250] LT: 25	

8.3.4 Applying the Proposed Composite Method

The literature analysis conducted within the context of Chapter 6 highlights the astonishingly positive verdict on the fitness of concept drift detection ensembles (CDDEs) in various contexts. However, it also reveals that experiments on real-world data are scarce to date, especially with regard to evaluations of their performance potential for concept drift adaptation measures, and that these constructs are difficult to parametrize (Korycki and Krawczyk 2019). Therefore, the second step of the sequential optimization and evaluation approach, as visualized in Figure 8.6, is designed to analyze SAM-CDAR on a full-system level, i.e. by employing E^{det} as concept-drift-detecting component for E^{est}'s adaptation mechanisms. To gather further evidence for responding to **RQ VI-2** and **RQ VI-3**, emphasis is placed on obtaining insights into the performance of the former. The respective experiment runs are pooled under the identifier Exp$_3$.

Experimental Configuration

Section 7.2 provides the technical details of the concept drift detection ensemble E^{det}, which takes as parallel inputs several performance representations computed from estimations of E^{est}. It is hypothesized that employing the former method increases the concept drift detection quality compared to employing a base detector as done in Exp$_2$. Only change-level concept drift detections

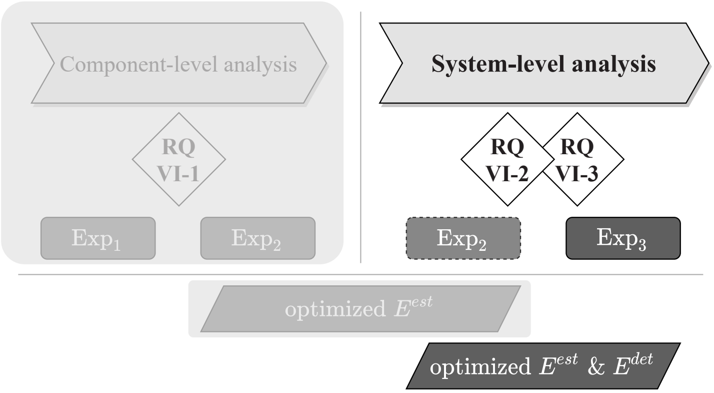


Figure 8.6: The system-level analysis of the sequential optimization and evaluation approach. It focuses on the evaluation of the concept drift detection ensemble E^{det} alongside a parameter-optimized instance of the adaptation method E^{est}. Evidence for research questions **RQ VI-2** and **RQ VI-3** is gathered.

are aggregated by E^{det}. In this experiment, the hyper-parameter-optimization-based experimental approach introduced in Section 2.3 is also utilized on the data of both industrial scenarios.

The method SAM-CDAR can be operated on standard computation hardware. However, to ensure its in-depth analysis, the dimensionality of the space spanned by the parameters varied in Exp$_3$ is vast. This is further expanded upon below. For this reason, the experiment is carried out on compute nodes of the high-performance cluster BwUniCluster2.0[14]. Each of these is equipped with two AMD EPYC 9454 CPU sockets that have a total number of 96 cores and a potentially available main memory of 384 GB.

Parametrization

Throughout Exp$_3$, optimized best-performing instances of E^{est}, which are separately determined during Exp$_2$ per industrial scenario, are employed without further parameter variation. Emphasis, however, is placed on varying E^{det}'s parametrization $\theta^{E^{det}}$ as, among other aspects, targeted by **RQ VI-2**. Given the trade-off between not overlooking even subtle but potentially harmful concept drifts in the stream and favoring exclusively high-confidence detections, finding a suitable parametrization is not trivial. Also, it is individually depending on properties of each of the considered industrial scenarios. This trade-off is visualized in Figure 8.7. To measure its

[14] Iterations of the computational resource bwUniCluster are funded by the Ministry of Science, Research and the Arts Baden-Württemberg and the Universities of the State of Baden-Württemberg, Germany, within the framework program bwHPC. The author of this thesis gratefully acknowledges the utilization of this resource.

performance contribution, the following parametrization is used:
The set of E^{det}'s members $\mathbf{D}$ comprises a fleet of $n^{det} = 4$ concept drift detectors D_i, with each featuring ADWIN as detection algorithm f^{det}. Their sensitivity is controlled via their parameter Δ. As formalized via the distribution strategy S^{det}_{dist}, these take as input the error scores mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE) and mean absolute percentage error (MAPE).
Regarding the MAPE input, two treatments are analyzed during Exp_3. Firstly, the two different computation-stabilizing modifications MAPE^{mod^1} and MAPE^{mod^2} (cf. (7.4) and (7.5) in Section 7.2) are separately applied with varied values of their common offset parameter o. This is done to compare the respective potential of these modifications to convey concept drift evidence. Secondly, the replacement of ADWIN with the Page-Hinkley test (PH) as MAPE-processing f^{det} is evaluated. The rationale of considering this replacement is based on MAPE values exhibiting a smaller variance than other error scores, as observed during preliminary experiments. This property can be exploited by PH using a sensitivity-controlling threshold τ^{PH} on its inputs as outlined in Section 3.2.4. This actually constitutes a variation from E^{det}'s definition in Section 7.2, which characterizes the error-value-processing sub-ensemble as a homogeneous one. Employing PH instead of ADWIN renders it a heterogeneous one.
Also, a classification-based representation of the industrial regression scenarios, formulated via discretization, is considered within the context of S^{det}_{dist} as another treatment. As explained in Section 7.2, discretization of the range of the stream's regression target values is memory-efficiently done via reservoir sampling. Its exponential decay dynamic is controlled via the bias parameter and the maximum reservoir size being statically assigned the values 0.001 and 12500, respectively. The actual quantile-based discretization involves a varied number of bins z. Values from bin-assigned estimated regression values are subsequently aggregated via the threshold τ^{det}_{class}. The resulting value is processed by an additional D_i featuring Drift Detection Method (DDM) as f^{det}. Therefore, n^{det} assumes the value 5 during the corresponding experiment runs. Only its change-level detections, sensitivity-controlled via parameter τ^{DDM}, are considered.

The aggregation of the alert outputs of E^{det}'s individual members D_i is done based on the aggregation strategy S^{det}_{agg}. It specifies a varied τ^{det}_{agg} as the minimally required number of alerting D_i to output a global concept drift detection. The alerts may occur within a varied time horizon m^{det}.

Together, Tables 8.4, 8.5 and 8.6 provide an overview of the ranges of utilized parameters. Such that are not explicitly mentioned are kept at their default values.

To keep this above-outlined parameter space manageable, Exp_3 is divided into three sequentially executed experiment run groups. The second and third of these each add a treatment while only keeping parameter values or value ranges that occur among the three best runs of the respectively

Table 8.4: Parameter ranges associated to E^{det}'s distribution strategy S^{det}_{dist} evaluated on data from industrial scenarios lead time (LT) and delivery time (DT). Square brackets indicate sets of parameter values that are applied in separate experiment runs.

	Error-score-processing member f^{det}
MAE	ADWIN
MSE	
RMSE	
MAPE	[ADWIN, PH]
Classification-based representation	[*none*, DDM]

Table 8.5: Parameter ranges associated to E^{det}'s member concept drift detector sensitivity evaluated on data from the industrial scenarios lead time (LT) and delivery time (DT). Square brackets indicate sets of parameter values that are applied in separate experiment runs.

	Sensitivity control
ADWIN: $\Delta*$	MAE: [0.0010, 0.0040, 0.0070, 0.0100]
	MSE, RMSE, MAPE: [0.0010, 0.0040, 0.0070, 0.0100]
PH: $\tau^{\text{PH}}**$	LT: [2000, 4555, 7111, 9666, 12222, 14777, 17333, 19888, 22444, 25000]
	DT: [200k, 455.5k, 711.1k, 966.6k, 1222.2k, 1477.7k, 1733.3k, 1988.8k, 2244.4k, 2500k]
DDM: τ^{DDM}	[2.2, 2.4, 2.6, 2.8, 3.0, 3.2]

* The ADWIN detectors processing MSE, RMSE and MAPE values are consistently assigned the same Δ value. The one processing MAE values is parametrized independently from the afore-stated ones.

** The LT-related parameter values result from discretizing the range from 2k to 25k into 10 equidistant steps. For DT, these values are heuristically multiplied by 100.

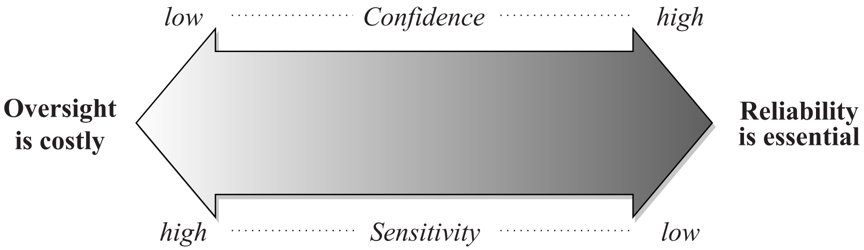


Figure 8.7: The trade-off in detecting those concept drifts that are crucial to be considered for adaptation measures. Its two extremes span a continuum. On the right side, high confidence refers to the expectation of alerts indicating actual and rather grave concept drifts, which requires low detector sensitivity to filter out false alerts. On the left side, in contrast, high sensitivity describes the positive tendency of processed concept drift evidence leading to relatively many alert emissions, which, consequently, can be characterized by a typically rather low confidence. This thesis posits that concept drift detection ensembles are well suited for supporting in finding viable solutions within this trade-off.

Table 8.6: Parameter ranges associated to E^{det}'s general settings. This includes member input computation modifications, parameters specifying the aggregation strategy S^{det}_{agg} and data-stream-related ones. Associated experiments are executed on data from the industrial scenarios lead time (LT) and delivery time (DT). Square brackets indicate sets of parameter values that are applied in separate experiment runs.

	Parameter value ranges
n^{det}	4
MAPE modification	[MAPE$^{mod^1}$, MAPE$^{mod^2}$]
MAPE o	[0.01, 0.10, 1, 10]
z*	[5, 7, 10, 15]
τ^{det}_{class}*	[0.50, 0.75]
τ^{det}_{agg}	[1, 2, 3, 4]
m^{det}	[50, 100, 150]
b	DT: 125 LT: 25

* Only relevant if a classification-based representation is considered among the inputs for E^{det}'s members. In associated runs, $n^{det} = 5$.

preceding group for further evaluation. These treatments aim at comparing different means for improving the quality, robustness and diversity of the concept drift evidence signals processed by E^{det}'s members. This approach is additionally visualized in Figure 8.8. The three groups contain the following experiment runs:

1. Runs employing exclusively the MAPE$^{mod^1}$ with a static o value of 0.1, and comparing ADWIN with PH as MAPE-processing f^{det}.
2. Runs comparing the computation modifications MAPE$^{mod^1}$ and MAPE$^{mod^2}$, and varying the value of o.
3. Runs considering classification-based representations of the industrial regression scenarios via adding a D_i featuring DDM as f^{det}.

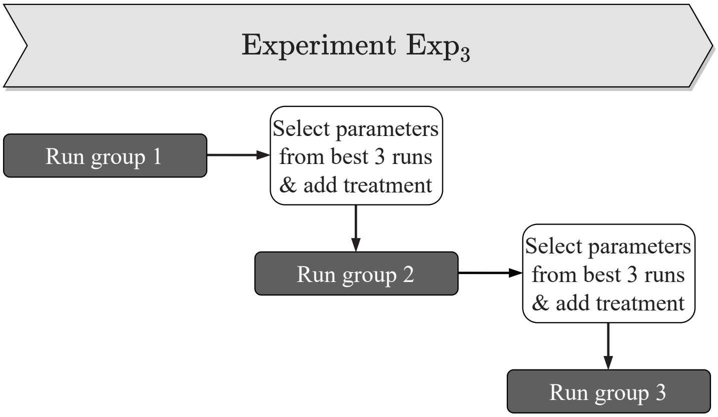

Experiment Exp$_3$
Run group 1
Select parameters from best 3 runs & add treatment
Run group 2
Select parameters from best 3 runs & add treatment
Run group 3

Figure 8.8: The approach of experiment Exp$_3$ featuring a separation into three sequential run groups to limit the parameter space to a feasible extend. From the first and second group, the parameter values or value ranges occurring throughout the respectively three best, i.e. lowest-error-achieving runs are selected for the respective subsequent run group. Each group introduces an additional treatment for evaluation.

8.4 Results Discussion

This section discusses the results obtained from conducting the experiments Exp_i as steps of the sequential optimization and evaluation approach outlined in Section 8.3. From these, findings and insights are derived and answers to research question **RQ VI** and its sub-questions are formulated. This comprises an analysis of the impact of proposed concept drift adaptation measures, parameter and robustness-related findings as well as a discussion of SAM-CDAR's ensembling components. Also, further notable remarks are provided and limitations are outlined.

8.4.1 Impact of Concept Drift Adaptation Measures

In the following, findings observed within the context of the experiments Exp_i with respect to concept drift adaptation measures of SAM-CDAR are discussed. These findings are quantified by comparatively reporting error and complexity-related scores.

Performance of the Proposed Nature-Inspired Concept Drift Adaptation Method over the Baselines

Comparing the results obtained from Exp_2 with those from Exp_1 provides evidence to base responses to **RQ VI-1** and **RQ VI-2** on. This represents the first step of the sequential optimization and evaluation.

Table 8.7 presents the MAE and MBPD values of the best-performing, i.e. achieving the lowest-error, run per approach, as observed during these experiments. Minima associated to the respective industrial scenario are formatted in bold. Scores achieved by the approaches $\text{RFR}_{\text{static}}$ and $\text{RFR}_{\text{naïve}}$ are listed in the two leftmost columns. When brought into context with E^{est}_{RFR}, if so inclined, they represent steps of an ablation study to be followed from right to left: Significantly reducing the adaptation and economization capabilities of E^{est}_{RFR} is loosely comparable to $\text{RFR}_{\text{naïve}}$. $\text{RFR}_{\text{static}}$ stands entirely without update mechanisms. When analyzing the MAE values, E^{est}_{RFR} clearly dominates. The successful contribution of E^{est}'s adaptation capabilities becomes apparent in both industrial scenarios. Effectively, utilizing E^{est}_{ARFR} instead of $\text{RFR}_{\text{static}}$ results in a drastic MAE reduction of approx. 43.40% in the LT scenario. With respect to the DT scenario, E^{est}_{RFR} realizes an even greater reduction of approx. 60.42%.

Naturally, $\text{RFR}_{\text{static}}$ would exhibit the lowest, i.e. best MBPD value as it is deliberately designed to not perform any regressor updates. It is thus beyond comparison here. $\text{RFR}_{\text{naïve}}$ beats E^{est}_{RFR} in both scenarios but, as explained in Section 8.3, would not do so on longer data streams as its data buffer would need to grow indefinitely.

Table 8.7: Data-stream-global mean absolute error (top two rows) and mean batch processing duration (bottom two rows) in seconds, observed during the experiments Exp_1 and Exp_2. The first and third row correspond to the lead time (LT) industrial scenario, while the second and fourth row correspond to the delivery time (DT) scenario (Trat et al. 2024).

	batch			continuous			
	$\text{RFR}_{\text{static}}$	$\text{RFR}_{\text{naïve}}$	E^{est}_{RFR}	ARFR	$\text{SRPR}_{\text{ARFR}}$	SRPR_{HTR}	E^{est}_{ARFR}
LT	7.51e+03	5.30e+03	4.38e+03	4.95e+03	4.80e+03	1.01e+07	**4.25e+03**
DT	1.00e+06	5.00e+05	**3.97e+05**	4.89e+05	4.23e+05	1.77e+06	5.10e+05
LT	-	0.87	1.41	**0.13**	1.82	0.35	0.23
DT	-	2.92	6.74	**2.18**	16.41	4.81	5.07

Instead of employing batch-training-type f^{est}, as indicated by the first three columns of Table 8.7, various continuous-training-type ones are employed in the case of the following four (cf. Section 3.1.1). Among these, both E^{est} approaches also outperform all others on LT data by a considerable margin as exhibiting the lowest, i.e. best MAE values. Specifically, E^{est}_{ARFR} achieves an MAE roughly 11.47% lower than the one observed for the next best baseline $\text{SRPR}_{\text{ARFR}}$. With respect to the DT scenario, only E^{est}_{RFR} outperforms the continuous-training-type baselines. It undercuts $\text{SRPR}_{\text{ARFR}}$, which is still the next best baseline, by approx. a moderate 6.18% in terms of the observed MAE. Among all approaches, SRPR_{HTR} is heavily underperforming in both scenarios. This might be due to its members lacking regressor complexity. What is still worth mentioning, is $\text{SRPR}_{\text{ARFR}}$'s volatile performance across various parameter configurations observed on DT data. Its relatively low MAE in Table 8.7 is actually the exception. Already slight deviations in n^{est} and Δ cause significantly higher MAE values with a mean of approx. 3.49e+07 s and a standard deviation (STD) of approx. 5.86e+07 s indicating a vast dispersion. A clear relationship is not apparent. In contrast, E^{est}_{RFR} behaves less sensitive in terms of MAE when varying its parameters with a mean of approx. 5.85e+05 s and a STD of approx. 1.34e+05 s.

Regarding the time constraints imposed by the industrial scenarios, all MBPD clearly fall within feasible value ranges. The approach $\text{SRPR}_{\text{ARFR}}$, however, has a remarkably high MBPD on DT data. As observed for its MAE across all of its parametrizations, this MBPD is even exceptionally low. It tends to increase with the value of n^{est} and exhibits an ample mean of approx. 58.39 s and an STD of approx. 36.12 s. In conclusion, parametrizing this approach is far from trivial and major pitfalls are possible. Again, with a mean of approx. 4.05 s and a STD of approx. 1.96 s, E^{est} approaches achieve far lower MBPD values in the same scenario, which also behave less sensitive when varying its parameters.

When analyzing $\text{SRPR}_{\text{ARFR}}$'s MAE and MBPD as observed during LT-associated experiments, such grave dispersion is absent. Among all experiments, ARFRs achieve the lowest MBPD values of all approaches. Regarding SRPR_{HTR}, its above-remarked inferiority in terms of the MAE is not reflected in the MBPD. It achieves quite competitive scores for the latter and outperforms both E^{est} approaches on DT data while being defeated by E^{est}_{ARFR} on LT data.

To provide further insight into the concept drift adaptation behavior exhibited by E^{est}_{RFR} and one of the few well-performing $\text{SRPR}_{\text{ARFR}}$ on the 1.5 years of DT data, their MAE is plotted in Figure 8.9. For a common reference axis, the MAE values are aggregated to consistently reflect the errors that would be computed for data batches with a constant b equal to 250. Additionally, to aid performance interpretation, the curves are then smoothed via a moving-average filter with a kernel size of 50 data batches. One can see that both approaches behave competitively in certain periods. Change-level concept drifts alerted by E^{est}_{RFR}'s concept drift detector are denoted by gray vertical dashed lines. In most instances, changes are handled well by E^{est}'s adaptation mechanisms as a reduced or maintained MAE can be noted subsequently. Of the total 42 detections, the first three lead to new member instantiations filling up E^{est}_{RFR}. The following 39 result in dominance challenges between reactive and stable members. Among those, eleven result in defeats of the stable member and thus the dismissal of the concepts it represents. In the remaining 28 cases, i.e. approx. 71.79%, the stable is victorious and thus the respectively oldest reactive member is suspended to make room for a new one (cf. Section 7.3). In contrast, only two concept drifts are detected by E^{est}_{ARFR} with its best-performing parameter configuration on LT data. Its adaptation behavior and observed MAE values are plotted in Figure 8.10. This finding and figure, however, receive closer attention in later parts of this chapter. As the implementation of the SRPR allows no tracking of individual detections, it is not further analyzed.

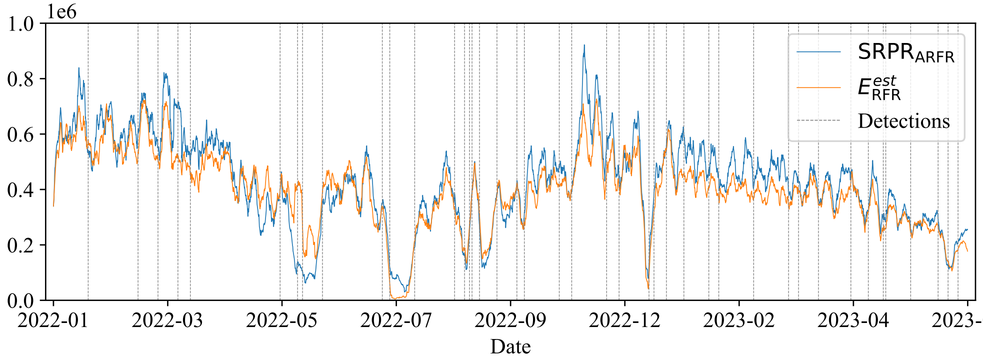


Figure 8.9: A comparison of the batch-wise mean absolute error of the suggested E^{est}_{RFR} and $\text{SRPR}_{\text{ARFR}}$ approach in seconds on delivery time (DT) data. The former is represented by the blue and the latter by the orange curve. Change-level concept drift detections of the former are marked by gray vertical dashed lines (Trat et al. 2024).

In conclusion, **RQ VI-1** is answered as follows: Even with the proposed method SAM-CDAR being reduced to E^{est}, one can consistently produce instances of it with parameter configurations that clearly outperform the considered baselines. This finding is based on the MAE and concept drift adaptation behavior observed within the context of the two considered industrial scenarios.

Performance of the Proposed Composite Method

Experiment Exp_3 represents the execution of the second step of the sequential optimization and evaluation approach analyzing the proposed method SAM-CDAR on a full-system level. The obtained results provide evidence relevant for, among others, **RQ VI-2**. As remarked in Section 8.3, optimized best-performing E^{est} instances are used throughout all associated experiment runs without further parameter value variation. On data from the industrial scenarios LT and DT, the approaches E^{est}_{ARFR} and E^{est}_{RFR} are thus consistently employed, respectively. Both associated parametrizations are given in Table 8.9 further below. For further discussion, the identifiers SAM-CDAR(E^{est}_{ARFR}) and SAM-CDAR(E^{est}_{RFR}) are used to mark the respectively employed E^{est} instance. This, consequently, shifts the focus to the analysis of the concept drift detection ensemble E^{det}, which is varied in its parameter values, and provides evidence relevant for **RQ VI-3**.

Table 8.8 presents the MAE and MBPD values of the best-performing E^{est} approaches from Exp_2 in the first two columns and the SAM-CDAR variants in the last two columns. For a direct comparison, scores of the former are merely repeated from Table 8.7. Minima associated to the respective industrial scenario are formatted in bold.

On LT and DT data, the respectively best-performing SAM-CDAR(E^{est}_{ARFR}) and SAM-CDAR(E^{est}_{RFR}) both outperform their counterparts E^{est}_{ARFR} and E^{est}_{RFR} by achieving MAE values that are approx. 0.80% and 1.03% lower, respectively. Employing E^{det} instead of a single base concept drift detector thus consistently leads to an improvement, albeit not a vast one. When comparing their performance with the baseline $\text{RFR}_{\text{static}}$ in both industrial scenarios, the SAM-CDAR instances realize error reductions of 43.85% and 60.82%, respectively.

Regarding the MBPD, SAM-CDAR(E^{est}_{ARFR}) exhibits a value that is slightly higher than the one of its E^{est}-only counterpart on LT data. Given the increased workload of processing concept drift evidence by more than one detector, this is as expected. It is therefore all the more surprising that the opposite is true in the DT scenario. There, SAM-CDAR(E^{est}_{RFR}) exhibits a value that is even considerably lower than the one of its E^{est}-only counterpart[15]. A possible explanation for this is that the CDDE of SAM-CDAR(E^{est}_{RFR}) issued 67 ensemble-global concept drift detections on the DT data stream. Given the 42 detections of its E^{est}-only counterpart in Exp_2, one might therefore argue that its E^{est} more frequently cycles through its reactive members. This can be confirmed by the fact that stable members, which often already completed their training in the past, prevail

[15] The experiments Exp_2 and Exp_3 are conducted on different compute hardware. Slight restrictions therefore apply when comparing MBPD values across these. Nevertheless, given the comparable clock frequency and performance of the respective CPUs, these restrictions are neglected in the context of this discussion.

over reactive competitors in 47, i.e. approx. 73.44%, of the total 64 challenges. This case occurs slightly more often than in Exp_2. Consequently, reactive members tend to get discarded relatively early and thus train on less data on average, which reduces batch processing effort. Accordingly, one might also argue that SAM-CDAR realizes a more economical label use. In contrast, the best-performing SAM-CDAR detected merely one concept drift within the LT data stream.

Table 8.8: Data-stream-global mean absolute error and mean batch processing duration in seconds, observed during the experiments Exp_2 and Exp_3. The first and third row correspond to the lead time (LT) industrial scenario, while the second and fourth row correspond to the delivery time (DT) scenario.

	E^{est}_{RFR}	E^{est}_{ARFR}	SAM-CDAR(E^{est}_{RFR})	SAM-CDAR(E^{est}_{ARFR})
LT	4.3763e+03	4.2482e+03	-	**4.2141e+03**
DT	3.9661e+05	5.1043e+05	**3.9253e+05**	-
LT	1.4076	**0.2297**	-	0.2788
DT	6.7353	5.0687	**2.7846**	-

For each industrial data scenario, Figure 8.10 provides insights into the concept drift adaptation behavior of the respectively two best-performing approaches. The top sub-figure displays change-level concept drift alerts and MAE curves of E^{est}_{RFR} and SAM-CDAR(E^{est}_{RFR}) on DT data. Analogously, the bottom sub-figure displays these insights for E^{est}_{ARFR} and SAM-CDAR(E^{est}_{ARFR}) on LT data. Aggregation and smoothing of the curves is done in the same way as for Figure 8.9. Regarding the DT scenario, one can see that more MAE-critical occasions are effectively identified and treated by SAM-CDAR than by E^{est} alone. Still, however, several occasions are even characterized by a decreasing MAE, which would technically require no adaptation. Regarding the LT scenario, SAM-CDAR outperforms E^{est} by detecting and beginning adaptation to concept drift at a point when a critical high-MAE phase starts. This way, the former more effectively sustains estimation performance throughout this phase. Furthermore, as Exp_3 reveals, a greater number of concept drift detections triggering adaptation measures at further points of increasing MAE throughout the stream do not lead to any performance improvement. With the given configuration, the upward error variation is already critically reduced. It therefore might not be possible to further reduce it with the considered means.

In summary, one might argue the following, which is also relevant for **RQ VI-2**: Even if SAM-CDAR constitutes an improvement over merely using its individual components, there is potential for improvement across industrial scenarios. E^{det} evidently does indicate valuable concept drift occasions for concept drift adaptation but lacks selectivity as explained above. Possible avenues for achieving associated improvements are suggested later on in this section.

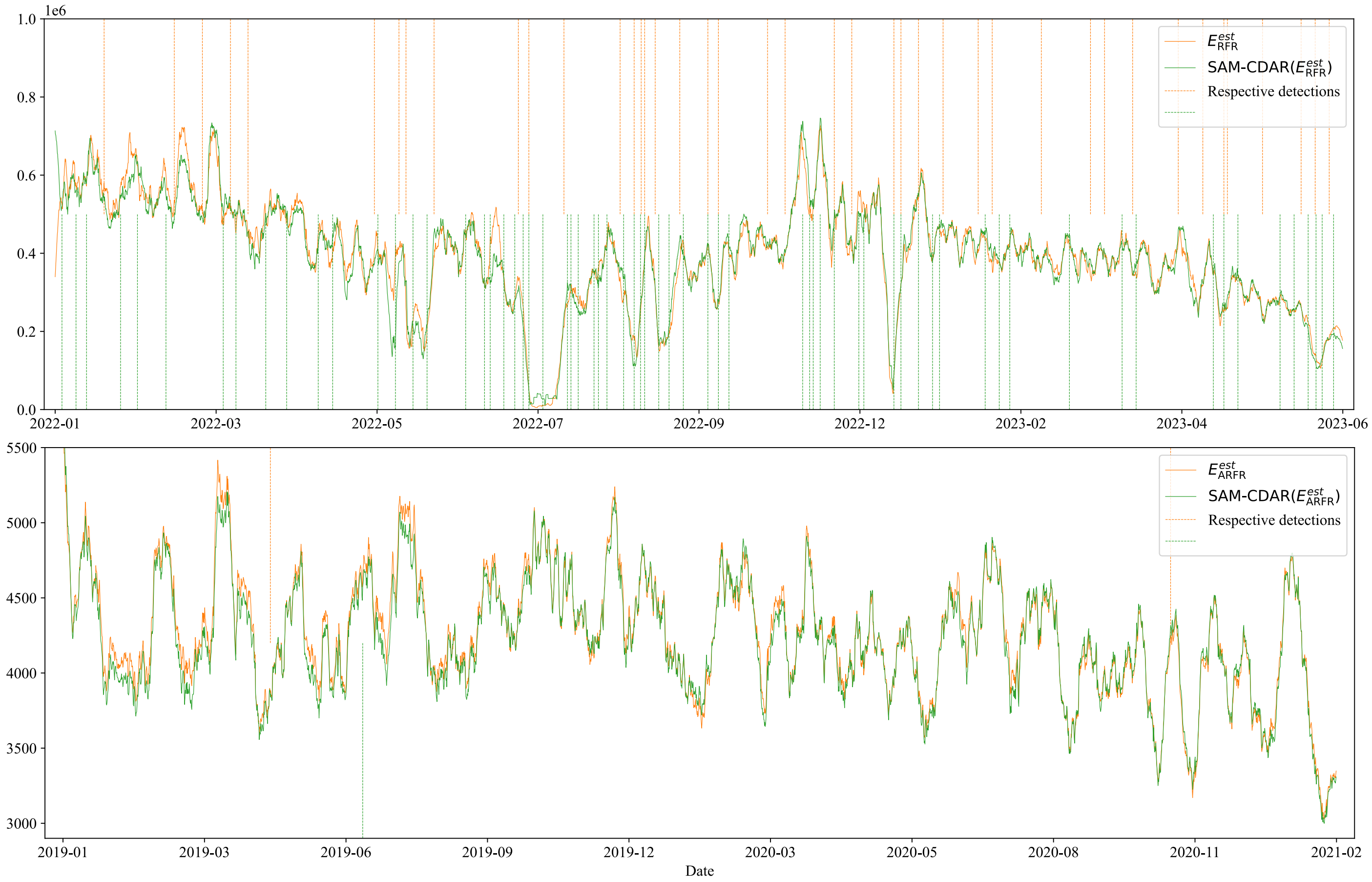


Figure 8.10: A comparison of the batch-wise mean absolute error of the respectively best-performing E^{est} and SAM-CDAR approaches on delivery time (DT) (top) and LT (bottom) data. The former are represented by orange and the latter by green curves. Change-level concept drift detections are marked by dashed lines colored according to the approaches that issue them.

8.4.2 Parametrization and Robustness

Detailed insights into the relevance of selected parameters of the proposed method SAM-CDAR and its components are discussed below. In particular, their influence on regressor robustness is highlighted, as targeted by **RQ VI-2**. Additionally, the parameter configurations identified as optimal for the industrial scenarios are provided.

The Proposed Nature-Inspired Concept Drift Adaptation Method

Conducting the experiments Exp_1 and Exp_2 reveals the parameter configurations presented in Table 8.9 to lead to the best $\text{SRPR}_{\text{ARFR}}$ and E^{est} approaches. These experiments reveal that both approaches E^{est}_{RFR} and E^{est}_{ARFR} outperform all other baseline approaches on LT data (cf. Table 8.7). What is remarkable, however, is the fact that this holds for all parameter configurations examined for the E^{est} variants. This reveals the proposed method to be strictly better for sustaining regressor robustness in this scenario, regardless of how it is parametrized. Also, employing the ARFR as E^{est}'s member f^{est} results in better performance than employing RFRs. Interestingly, doing so is strictly better, as all E^{est}_{ARFR}-related parameter variations completely outperform its RFR-member-based counterpart. Across all associated experiment runs, the former detect on average around 1.53 concept drifts in the data stream (with an STD of approx. 0.56), whereas the latter consistently detect none.

Table 8.9: Parameter configurations resulting in the lowest data-stream-global MAE values of selected approaches in the lead time (LT) and delivery time (DT) scenario (Trat et al. 2024).

	$\text{SRPR}_{\text{ARFR}}$	E^{est}_{RFR}	E^{est}_{ARFR}
LT	n^{est}: 2 Δ: 0.010	κ: 100k, b: 25 w^s_{min}: 0.4, a: 0.01 Δ: 0.007	κ: 100k, b: 25 w^s_{min}: 0.6, a: 0.01 Δ: 0.007
DT	n^{est}: 1 Δ: 0.001	κ: 750k, b: 125 w^s_{min}: 0.2, a: 0.02 Δ: 0.010	κ: 1M, b: 250 w^s_{min}: 0.4, a: 0.02 Δ: 0.007

The inverse can be observed for the DT scenario: E^{est}_{RFR} achieves the lowest MAE across not only E^{est} variants but all benchmarked approaches. This achievement of the proposed method is made possible by splitting each batch received from DT's data stream in half, effectively resulting in a batch size b of 125, and processing both parts sequentially. In contrast, as maintaining a b of

250 does not enable E^{est}_{RFR} to outperform SRPR$_{\text{ARFR}}$, one might assume that reducing b is vital to enable superior concept drift adaptation capability (Trat et al. 2024). This reduction leads to a decreased delay before E^{est} can regularly start adaptation measures, which possibly compensates for an advantage SRPR$_{\text{ARFR}}$ might draw from a b of 1. Given the already small b of 25 for the LT scenario, no associated further experiments are conducted. These findings highlight the importance of careful selection of f^{est} and b values.

Other parameters also require careful setting as the experiments reveal no single method with outstanding performance in both industrial scenarios. For instance, higher values of the stable member weight minimum w^s_{min}, i.e. 0.4 or 0.6, and the lower value of 0.01 of the accretion rate a (cf. Table 8.9) result in lower MAE values in the LT scenario. This finding hints towards the possibility of extended concept drifts, i.e. incremental or gradual, to be present among the few ones in this data. This can be argued as these parameter values ensure a high degree of the stable member's decision weight, i.e. the one representing longer-lasting concepts, and an attenuated experience-based weight increase for reactive members. The results also indicate that these concept drifts are treated well by E^{est}_{ARFR} while requiring the maximum training volume κ of 100k data points.

On DT data, however, an inverse relationship applies: Lower w^s_{min}, i.e. 0.2 or 0.4, and the higher a of 0.02 result in lower MAE values. This points towards abrupt concept drifts being present in the data, which E^{est} adapts to well by establishing plasticity, i.e. favoring rather volatile concept representations, among its members. Since several detections overlap with peak phases of the Russian invasion of Ukraine (cf. Figure 8.9), one might argue that resulting timber resource shortages in furniture supply chains can be an underlying cause for these concept drifts. Regarding label demand, κ values of 500k and 750k enable E^{est}_{ARFR} to achieve a quite compatible MAE of approx. 3.98e+05 s and the scenario-wide minimum of approx. 3.97e+05 s, respectively. Further increasing κ yields no performance improvement. Therefore, these approaches still consume the majority of available labels. Given the many concept drifts in this scenario, this is as expected. When compared with the considered baseline approaches, which consume all available labels per default, E^{est}'s design is, nevertheless, more resourceful. It automatically reduces label consumption for member training purposes in periods of no or lower-magnitude concept drifts and is, moreover, able to handle label delay.

Together with the findings from Exp$_2$, the discussion of these aspects aids in answering **RQ VI-2**: The proposed method E^{est} can be finely parametrized to handle even various types of concept drifts in the industrial scenarios. It can be argued that this is a key property for sustaining the robustness of productively employed regressors.

The Proposed Composite Method

The parameter configurations of the best-performing SAM-CDAR approaches, identified in Exp$_3$, are presented in Table 8.10. As previously demonstrated, SAM-CDAR outperforms its E^{est}-only variant in both industrial scenarios. The parameters of the latter are retained unchanged while adding the CDDE E^{det} to compile the former, including unvaried b values.

Table 8.10: Parameter configurations resulting in the lowest data-stream-global MAE values in the lead time (LT) and delivery time (DT) scenario, observed for the approaches SAM-CDAR(E^{est}_{ARFR}) and SAM-CDAR(E^{est}_{RFR}), respectively.

		SAM-CDAR(E^{est}_{ARFR})	SAM-CDAR(E^{est}_{RFR})
f^{det}	n^{det}	4	5
	MAPE-processing	PH	
	Classification-based-representation-processing	*none*	DDM
Preprocessing	z	*none*	7
	τ^{det}_{class}	*none*	0.5
Sensitivity	ADWIN: Δ	MAE: 0.0070	
		MSE, RMSE: 0.0040	MSE, RMSE: 0.0070
	PH: τ^{PH}	19888	1477.7k
	DDM: τ^{DDM}	*none*	2.8
MAPE	modification	MAPE$^{mod^1}$	
	o	0.01	0.10
S^{det}_{agg}	τ^{det}_{agg}	3	2
	m^{det}	100	

Arguably, E^{det}'s aggregation strategy S^{det}_{agg} plays a central role in achieving this success. From Exp$_3$'s runs, the following values emerge as MAE-optimal for the threshold τ^{det}_{agg}, which marks the minimally required number of member detectors D_i emitting a change-level alert to issue an ensemble-global alert. On LT data, the value 3 is identified, representing a relatively great number of D_i, i.e. more than half of the entire ensemble. Possibly, as the total number of concept drift detections is lower when compared to Exp$_2$, this scenario demands a high alert confidence level to identify actually harmful concept drifts and ignore performance-irrelevant ones. On DT data, the alerts of 2 D_i, i.e. less than half of the entire ensemble, suffice to emit an ensemble-global alert. As visualized in Figure 8.11, the MAPE-processing D is observed to contribute to almost all, i.e.

61 of the 67 ensemble-global alerts, followed by the MAE-processing D contributing to 34. In this regard, it is worth noting that none of the D_i reacts overly sensitive. Most D_i emit no or almost no alerts that do *not* lead to ensemble-global alerts. Only the MAPE-processing D emits 197 alerts in total. This does not constitute an excessive detection behavior given the 11739 evaluated 125-data-point-sized batches that could theoretically all lead to a detection. Consequently, one might argue that the MAPE provides evidence hinting to many subtle concept drifts, which have negligible impact on regression performance, while the remaining error scores hint to more grave ones.

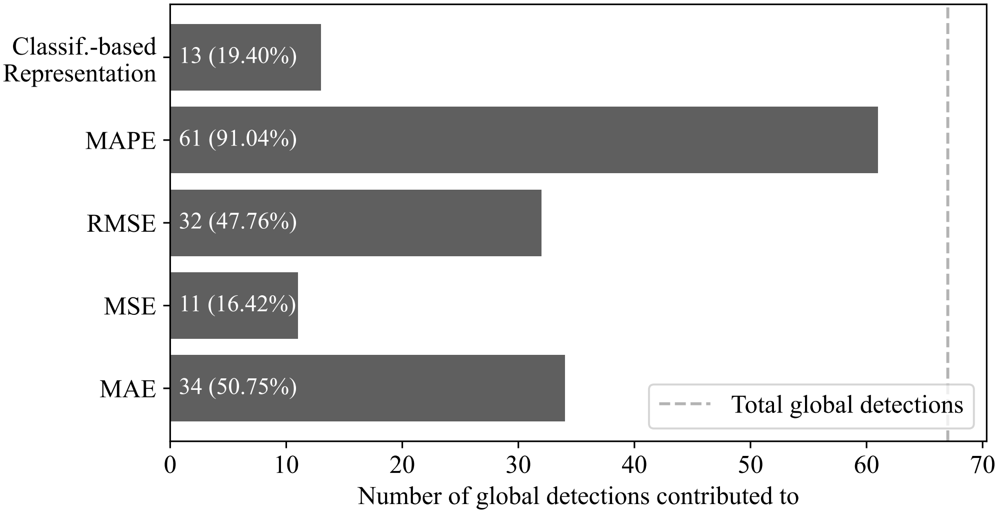


Figure 8.11: The contribution of each of E^{det}'s members to ensemble-global detections as observed when applying SAM-CDAR(E^{est}_{RFR}) on data from the delivery time (DT) industrial scenario. The bars denote the number of contributing alerts emitted by the individual members processing the set of regression error scores, which are resolved on the vertical axis. The total number of 67 ensemble-global alerts is marked by the vertical gray dashed line.

If one would increase τ^{det}_{agg} to 3 on DT data, the number of ensemble-global detections would be drastically reduced to 26. This would therefore correspond to an increased alert confidence requirement.

The distribution of the 67 concept drift detections across the DT data stream yield an average horizon of 87.49 batches between the first D_i-individual alert and the established ensemble-global one. With the parameter value 100 for the aggregation time horizon m^{det}, during which the above-referenced member alerts need to be emitted, increasing it would not substantially increase the ensemble-global detection sensitivity. Reducing it to 50 would decrease the total number of concept drift detections to 56, yielding an average horizon of 49.00 batches. Consequently, this would, similarly as increasing τ_{agg}^{det}, increase the density of high-confidence alerts while risking the oversight of still less or similarly harmful concept drifts.

The various treatments introduced via the set of sequentially executed experiment run groups (cf. Figure 8.8) can be observed to have varying impacts across the considered industrial scenarios. The first run group reveals PH as the better choice for processing the MAPE error score signal for both scenarios. Nevertheless, it should be noted that this concept drift detector is harder to parametrize as ADWIN across different data scenarios, which are characterized by quite different degrees of typical error severity. The reason for this is that τ^{PH} has a direct relation to the raw input value magnitude and not to its data-distribution-related properties (cf. Section 3.2.4). Also, a further conclusion is that the resulting heterogeneous CDDE architecture may offer advantages over a purely ADWIN-populated homogeneous one in these scenarios.
Variations in computing the MAPE are subject of the second run group. Also in both scenarios, employing the MAPE$^{mod^1}$ as numerically stabilizing measure strictly dominates MAPE$^{mod^2}$. However, the optimal value for the associated offset o varies across the scenarios. The values 0.01 and 0.10 are identified for the LT and DT scenario, respectively.
The integration of a classification-based representation of the respective regression problem via discretization is, in contrast, not a consistently dominant choice. This constitutes one of the main findings of the third run group. On LT data, adding a DDM member processing the associated evidence from the representation does not result in an increased concept drift adaptation and thus estimation performance. Doing so to process the DT data stream, however, does. Consequently, one might argue that this representation makes accessible valuable concept drift evidence and that yet more potential can be extracted from an even greater architectural heterogeneity. With DDM's default τ^{DDM} value of 3.0 and its underlying three-sigma rule assumption, this detector would output concept drift alerts only if the continuously observed error rate exceeds its historical minimum by more than three standard deviations (cf. Section 3.2.4). Figure 8.12 presents the MAE when varying exclusively τ^{DDM} of the best-performing approach. It shows that the values 2.6 and 2.8 for τ^{DDM} lead to a better and optimal performance, respectively, although the error range is not vast. Therefore, a slightly more sensitive DDM configuration, i.e. a lower threshold on the error rate, is beneficial for sustaining regressor robustness.

As a concluding response to **RQ VI-2**, Exp$_3$ reveals that E^{det} has a boosting effect on regression performance and robustness. It leverages concept drift evidence and detector diversity, and exposes parameters that can exploit the confidence-overlooking-risk continuum (cf. Figure 8.7) in a fine-grained fashion.

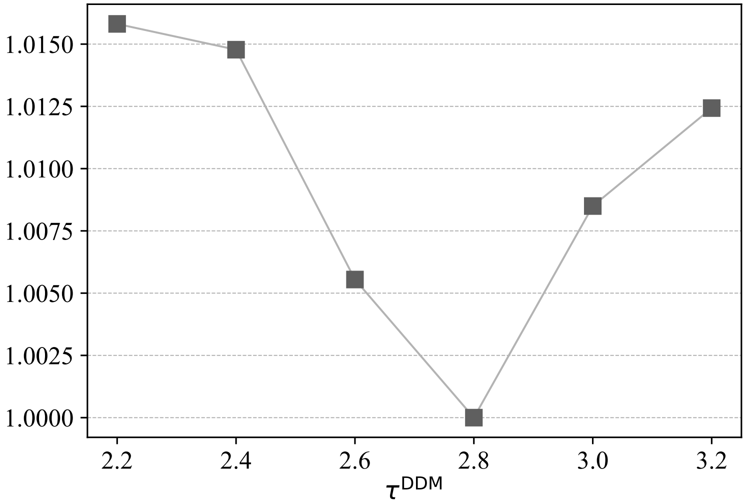


Figure 8.12: Impact of varying the concept drift detector Drift Detection Method's parameter τ^{DDM} on the mean absolute error achieved by SAM-CDAR(E^{est}_{RFR}) on delivery time (DT) data as observed during experiment Exp_3 (cf. Table 8.8). The errors plotted against the vertical axis are normalized by the performance peak value achieved with τ^{DDM} equal to 2.8.

8.4.3 Impact of Ensembling

The following is specifically devoted to **RQ VI-3** as the use of machine learning ensemble methods is of particular importance for the design of the proposed method SAM-CDAR. Fueling this choice is the hypothesis that ensembles foster diversity and, as a result, improve concept drift adaptation performance. As symbolically represented by Figure 8.13, SAM-CDAR features such constructs for both regressors and concept drift detectors.

As already expanded upon during the preceding discussion, E^{est} exerts a profoundly robustness-sustaining influence. This becomes apparent, for instance, through the results observed for experiments Exp_1 and Exp_2. On LT data, both E^{est} approaches outperform the considered popular baselines, partially even largely independently of the selected parametrizations. It makes more extensive use of ensemble capacities, and thus increases regression model complexity, by scaling up and cycling members, which may provide a critical advantage over the baselines. With respect to simpler static or naïve baselines, the conducted ablation analysis also demonstrates E^{est}'s superiority driven by ensembling.
Within in this context, the notion of Exp_1 and Exp_2 representing a comparative evaluation not on equal terms should be commented. The baselines are characterized by a static member cardinality while E^{est} provides for its increase following concept drift detections. This can be countered by referring to associated E^{est} mechanisms as constituting explicit design choices. The baselines

do not consider this choice and instead make scaling up the responsibility of the user via manual parametrization.

The increase in diversity achieved by E^{det} is also not to be dismissed. Particularly on DT data, it outperforms the use of merely a base detector by evidently identifying more concept drift occurrences that demand adaptation to sustain regressor robustness. Simultaneously, it even reduces the observed MBPD, rendering SAM-CDAR more computationally efficient. The data-stream-global MAE reduction achieved this way, however, is only moderate in both industrial scenarios. Therefore, one might argue that ensembling concept drift detectors is beneficial. This seems to even hold with D_i that would exhibit poor performance when employed in a stand-alone fashion, i.e. without ensembling. Such a conclusion can be drawn when again considering the findings of the work of Voß (2025), which was supervised by the author of this thesis. They indicate that employing a stand-alone DDM on this data results in an adaptation performance inferior to utilizing continuous-valued-input-processing concept drift detectors, such as ADWIN and PH. Nevertheless, given the previously outlined results, DDM still constitutes a CDDE-enriching D_i as adding it evidently increases diversity and improves adaptation performance.
The trade-off continuum spanned by the extremes of avoiding the oversight of concept drifts and favoring high-confidence detections is described initially (cf. Figure 8.7). Implicitly, the former extreme refers to the risk of excessive concept drift detection alerts issued by E^{det}. Based on the discussed results, this risk turns out to be manageable via ensemble-level parameters as explained before.

Using ensembles for inference can incur costs in terms of increased memory and time complexity and thus compute requirements as pointed out by Figure 8.13. In the considered industrial scenarios, however, these costs are highly reasonable. The observed complexities are straightforwardly manageable with the pre-existing hardware of the companies. This finding is in line with those of Read and Žliobaitė (2023). Regarding the developing research field of data stream mining, they furthermore argue that a more nuanced definition of associated requirements would in order. As supported by the results obtained in this thesis, many practical scenario contexts do not have high demands with respect to batch processing durations and methods extensively making use of ensembling can feasibly be operated.

In summary, while the observed results hint towards greater improvement gained from E^{est}, E^{det} still has a consistently robustness-sustaining impact. Ultimately, a comparison of different approaches to increasing diversity through ensembling is one of the central aspects addressed by this thesis. Therefore, as also pointed out by Woźniak (09.10.2023), one might respond to **RQ VI-3** as follows: Ensembling regressors possesses greater robustness-sustaining potential than ensembling concept drift detectors. Nevertheless, if the compute capacity of the respective scenario permits it, integrating the latter is advisable. Furthermore, especially in scenarios in

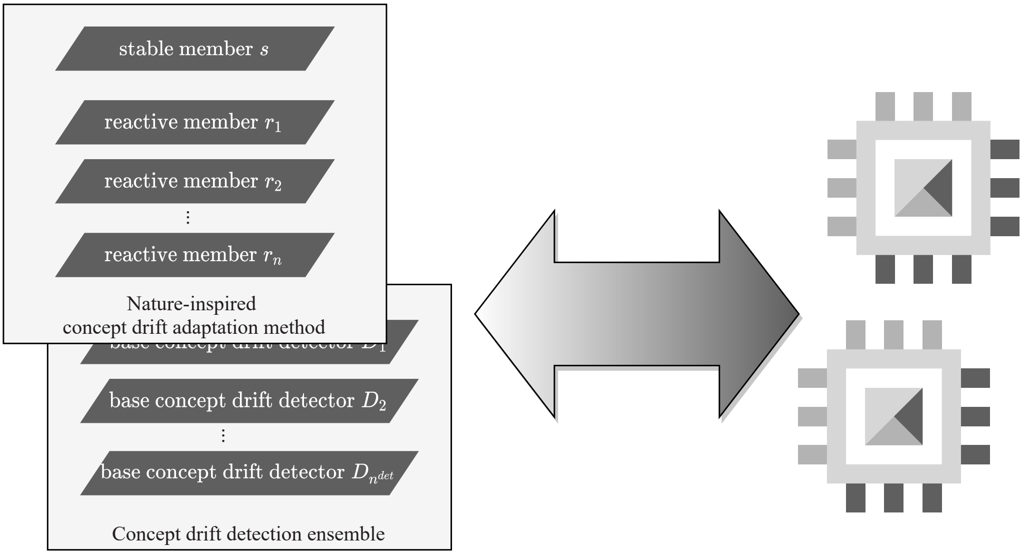

stable member s
reactive member r_1
reactive member r_2
⋮
reactive member r_n
Nature-inspired
concept drift adaptation method
base concept drift detector D_1
base concept drift detector D_2
⋮
base concept drift detector $D_{n^{det}}$
Concept drift detection ensemble

Figure 8.13: The robustness-sustaining potential and costs attributed to employing ensemble methods. The proposed method SAM-CDAR comprises both ensembles combining regressors as well as such combining concept drift detectors, which is displayed on the left side of the figure. The right side symbolically depicts that this method can potentially require great compute capabilities.

which many concept drifts of various types are to be suspected, even greater improvement can possibly be gained from ensembling detectors.

8.4.4 Further Remarks

In the following, several remarks on other notable insights gained from the experiments Exp_i shall be provided.

Error evidence evaluation. As explained in the previous Section 8.3, the MAE is used as primary evaluation metric for the Exp_i. Apart from this metric, several others, such as the RMSE, are also consistently recorded. However, since their values qualitatively confirm the findings of the MAE, they are not reported.

Rendering batch estimators stream-capable. Industrial practitioners are increasingly faced with the task of transferring estimators successful on static datasets to productive streaming use. This is a non-trivial task as the choice of and experience with estimators featuring continuous-training-type algorithms f^{est} is severely limited. The suggested method E^{est} alleviates this issue as it is model-agnostic and enables the user to also employ estimators featuring batch-training-type f^{est} (cf. Section 3.1.1). In many practical contexts, these types of estimators are well understood and often already in use (Lima et al. 2022, Read and Žliobaitė 2023). This fact should therefore be highlighted as a distinct strength of E^{est} for practical use, especially when compared to the baselines' limitations in this regard. At the same time, this property of E^{est} can also leverage training speed-ups. In cases of data arriving in large batches, batch-training-type f^{est} can be superior to continuous-training-type ones as evidenced by E^{est}'s MBPD values being consistently lower than those of the competing SRPR$_{\text{ARFR}}$ (cf. Table 8.7). However, this results in a trade-off between high batch sizes b for speed-ups and lower b for quicker adaptation capabilities as discussed above. In contrast, in cases of data arriving in small batches or even point-wise at high velocities, employing estimators featuring continuous-training-type f^{est} can be advisable, which is e.g. demonstrated by the unrivaled MBPD values of ARFR baselines.

Outlook. Future experiments analyzing SAM-CDAR should gather additional empirical evidence on more benchmark datasets. This includes synthetic ones to further increase the concept drift type and location variability, and real-world ones involving different magnitudes of label delay or deficits. The latter could also be supplemented with synthetic concept drifts, e.g. with highly realistic properties as expanded upon in Section 6.4.3, for further analysis.

Additionally, it would be desirable to gather more evidence on how the label-economizing capabilities of E^{est} unfold on even longer data streams. For long-term practical use, such insights would help to better assess performance potentials. Otherwise, careful monitoring and re-evaluation of different approaches are strongly advisable.

Regarding concept drift detection and adaptation for regression problems, several potentials to further improve associated methods falling out of the scope of this thesis are worth exploring. As the author of this thesis describes in the work of Sturm et al. (2024), concept drifts can also be categorized based on their respective local raw-data context. In the referenced work, the loss continuously recorded during foundation model fine tuning serves as detector input data. Detected concept drifts are categorized as either positive or negative if being localized within loss-decreasing or increasing ranges, respectively, identified via e.g. computing local gradients. Positive concept drifts can then be ignored, as models actually exhibit a desired behavior in such ranges, while exclusively negative ones are subject to adaptation. This heuristic can be directly transferred onto analogously categorized error score trends within the context of regression problems by only treating negative concept drifts.

Furthermore, other approaches for the preprocessing of concept drift detector input data could be evaluated. Instead of raw error score data, its first or second-order derivative could be utilized as this shifts emphasis onto changes in error behavior. Likewise here, positive concept drifts could be ignored. Members processing such scores can be added to E^{det}, while possibly setting differing weights determined via optimization.

8.4.5 Limitations

The following paragraphs address limitations that apply for the conducted experiments Exp_i.

Hyper-parameter optimization. The experimental parameter optimization during the Exp_i, as remarked in Section 2.3, is done in a grid-search-based fashion within a manually defined discrete finite space. As a result, unexplored yet potentially beneficial parameter configurations are conceivable. Nevertheless, further exploration is not done in this thesis to limit the experimentation complexity to a manageable extent. This limitation is also reflected by the following experiment design choices: Only the MAE-processing member concept drift detector D has its sensitivity-controlling parameter Δ varied separately. The other D_i processing the MSE, RMSE and MAPE receive a common Δ value varied across experiment runs. To a certain extent, this is relaxed by having PH process the MAPE in subsequent experiment runs. Still, especially the CDDE capabilities stemming from homogeneous architecture properties could be further explored by decoupling the sensitivity-controlling parameter variation for all D_i. Also, with Exp_3 being designed as three sequentially executed experiment run groups (cf. Figure 8.8), several configurations are not evaluated as certain parameters are kept constant during subsequent runs. While these might introduce new findings, their exploration is out of scope for this thesis. Further experiments could therefore address these limitations by e.g. augmenting the search space or conducting a random-sampling-based search on continuous parameter ranges.

Supervisedness. In practice, handling labels can be subject to difficulties as visualized in Figure 8.14. This includes costs associated with obtaining them, quality related issues, as well as delays in their arrival (Žliobaitė et al. 2016). As initially outlined, immediate label access is assumed throughout the Exp_i. This choice is based on the motivation to determine upper

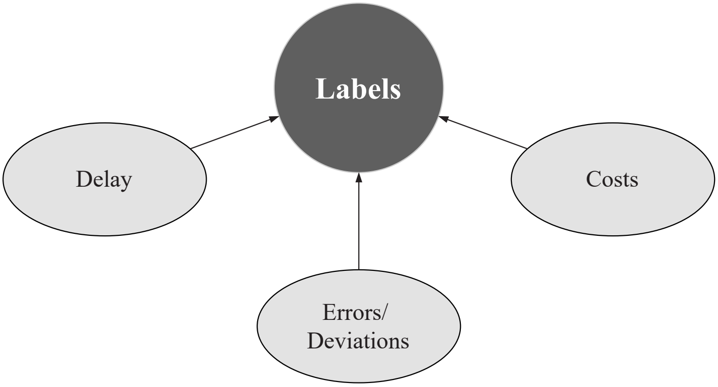


Figure 8.14: Difficulties associated with labels. Obtaining them can incur costs, they may be of poor quality or delayed.

bounds of the performance potential of the methods proposed in this thesis. Nevertheless, these methods can be employed in semi-supervised, e.g. via active learning, and unsupervised settings, as explained in Chapter 7.

To address this limitation, the project defined by the thesis author, previously referred to within the context of forgetting strategies in Section 7.3, considers the case of arbitrarily delayed labels. More specifically, the undergraduate student Ott (2024) carrying out this project quantitatively analyzes the impact of the delay on regression performance in the DT scenario. He finds a considerable increase of E^{est}'s error when compared to results from Exp_2 while still well outperforming static baselines. As solutions, he recommends further careful parametrization as well as exploiting warning-level concept drift detections for background regressor training to accelerate the adaptation of new concepts.

Another project defined by the thesis author evaluates the performance of CDDEs when employed in an unsupervised fashion. Here, E^{det} instances featuring unsupervised concept drift detectors, i.e. such processing input data not derived from labels, as D_i are applied to the DT regression as well as to benchmark problems. The postgraduate student Kraus (2025) conducting the associated experiments finds that E^{det} is still able to detect robustness-harming concept drifts. He implicitly evaluates the detection quality by, similar to the Exp_i, measuring the resulting performance of adaptation using labels exclusively for regressor retraining. As heuristically retraining as often as concept drifts are detected by E^{det} at constant intervals and using solely one base detector both result in inferior adaptation performance, this finding is further substantiated. As expected, however, the supervised configuration evaluated in Exp_3 is not outperformed. Using specifically designed features to increase the concept drift evidence content can be a further means to improve the detection quality in an unsupervised fashion. For instance, one could employ such that might not be of primary importance for regression but carry valuable concept-drift-indicating information, e.g. based on timber market price data. Another example is to task experts with engineering features, e.g. as linear combinations of multiple ones with weights reflecting knowledge of their individual importance with respect to the problem stationarity.

8.5 Summary

This chapter validates the stream-applicable method for concept drift adaptation in regression scenarios (SAM-CDAR), which is proposed in Chapter 7, within the context of two real-world industrial regression problems. The associated experimental approach is scientifically guided by research question **RQ VI**, which is concerned with SAM-CDAR's performance with respect to handling concept drifts. Details of the experiment design as well as assumptions are provided and the obtained results, limitations and specific future directions are discussed.

Most prominently, it is found that SAM-CDAR is not only able to adapt to major concept drifts, e.g. caused by timber market disruptions from acts of war or personnel attendance problems due to global health crises, but also subtle ones. Consequently, it is capable to effectively sustain regressor robustness and outperforms popularly employed baseline approaches. Nevertheless, also weaknesses of SAM-CDAR are identified, e.g. with regard to a lack of selectivity in the detection of concept drifts, and corresponding avenues for improvement are proposed.

Within the broader scientific context, this chapter helps to close certain research gaps. On the one hand, it addresses a persistent lack in the body of research on concept drift detection ensembles: While existing works primarily focus on their concept drift detection performance, the evaluation presented in this chapter establishes a novel focus by providing evidence for their impact on adaptation performance. Also, novel design approaches for concept drift detectors suitable for use for regression problems are provided. On the other hand, further practical-finding-backed insights into employing regressor ensembles in evolving data streams are gathered. These include measures for economizing ground truth demand as well as yet another argument in favor of a more application-oriented dimensioning of such methods not unnecessarily constrained by overly restrictive memory or processing assumptions (cf. Read and Žliobaitė 2023).

9 Conclusion and Outlook

The thesis is concluded in the following. This is done by summarizing the most important aspects and findings of the preceding chapters as well as by proposing potential future work.

9.1 Conclusion

In Chapter 1, the motivation for this work within the growing industrial adoption of machine learning is established. It emphasizes that real-world industrial conditions frequently expose productive machine learning systems to concept drift and related disturbances. As a result, sustaining the robustness of their estimators over time poses a critical challenge. Despite existing research in this field, the lack of consistent integration strategies, particularly for regression problems and industrially deployed systems, persists. To address these gaps, a practically grounded research approach aiming at improving not only concept drift handling measures but also the rigorous design of associated methods and systems is selected for this thesis.

Chapters 2 and 3 establish the groundwork for the following ones. The former outlines the methodological foundations that guide the careful research approach followed by this thesis. Methods and principles for gathering knowledge, condensing it in model-based forms and conducting experimental evaluations in practical environments are introduced.
The latter chapter provides a theoretical basis for the methods applied in this thesis. Foremost, key properties of the central thesis topic are elaborated on: Concept drift and the vital importance of handling it to sustain estimator robustness in practical industrial scenarios via detecting it as well as adapting to it. Core principles and prominent representatives of batch-based and continuously trainable machine learning estimator models as well as how to measure their performance are covered. Aligned with the thesis focus, emphasis is consistently placed on industrial regression contexts. Subsequently, the following methods are additionally introduced: Machine learning ensembles, as a recurring principle for improving both estimator and concept drift detector diversity, and the description method Subject Orientation rigorously ensuring sound systems modeling devoid of ambiguities. The contextualization of these aspects within practical frameworks is taken into account by discussing several popular data science process models.

Concept drift detection and adaptation basics are complemented by recent state-of-the-art approaches in Chapter 4, while establishing a focus on explicit methods for the latter. As detection methods for classification problems have persistently received significantly more attention, a serious gap in research on regression-related ones is identified. Regarding both detection and adaptation, information on existing approaches employing ensemble-based methods is gathered: On the one hand, research on combining detectors, referred to as concept drift detection ensembles (CDDEs), is found as still being in its early stages. Combining estimators, on the other hand, has already received research attention but still lacks in terms of the variety of models that can be integrated.

This chapter also explores the field of productive machine learning systems. It finds that employing concept drift adaptation within such to sustain robustness is an under-researched aspect. Together, the aforementioned research gaps directly motivate the novel contributions developed in the subsequent chapters.

Chapter 5 delves deeper into the productive use of machine learning estimators in industrial environments. Initially, it defines further key terms: Firstly, machine learning robustness is formalized as the sustained performance of an estimator combined with its ability to rapidly recover from concept drift. Secondly, inference process systems, previously referred to as machine learning systems, are characterized as the software and infrastructural context for productive estimators.

The systematic literature review conducted for this chapter tends to the limitations of modeling approaches for inference process systems with robustness-sustaining components. It reveals several shortcomings with respect to the coherence of suggested approaches, i.e. the relatively low degree of the respectively resulting models' consistency, clarity and informative content. Further insights are derived from analyzing the most mature data science process models identified in Chapter 3, such as PAISE and CRISP-ML(Q), which show insufficient emphasis on robustness in operational phases. As a direct response, a reference architecture for inference process systems that explicitly supports the coherent modeling of robustness-sustaining components as well as concrete guidelines for its practical use are proposed.

Chapter 6 profoundly explores CDDEs. In the course of an extensive systematic literature review, a bibliometric and a qualitative analysis are conducted. In-depth insights of their scientific field reveal that CDDEs are still under-researched despite their great performance potential. To support their proper careful configuration for practical use cases, a taxonomy and an ensemble-knowledge-backed blueprint, which examines CDDE design aspects and their interaction in detail, are proposed. In subsequent parts of this thesis, they are leveraged as powerful component for guiding adaptation measures and thus sustaining regressor robustness in practical industrial scenarios. This constitutes a further novelty as it augments the previously mostly detection-performance-oriented research focus.

Chapter 7 then consolidates the previously proposed practically applicable artifacts into a composite method for the under-researched problem of sustaining regressor robustness in industrial data streams. Being denoted as stream-applicable method for concept drift adaptation in regression scenarios (SAM-CDAR), it combines a CDDE with a novel machine learning ensemble approach for adaptation, also proposed in this chapter. The latter leverages nature-inspired algorithmic mechanisms to improve concept drift adaptation measures while adhering to realistic compute constraints. By additionally being model-agnostic, it facilitates the straightforward transfer of arbitrary regressors, which may already be known to perform well on historical data, to concept-drift-robust use in data streams.
The design approach of SAM-CDAR utilizes the reference architecture as well as the design principles for CDDEs proposed in Chapters 5 and 6, respectively. As it is not solely defined for supervised but also for semi-supervised scenarios, it possesses a highly practical relevance for real-world industrial contexts.

In Chapter 8, the method SAM-CDAR is experimentally validated in two real-world industrial scenarios, impacted by various types of harmful subtle as well as severe concept drifts. Their regression tasks are the estimation of durations of forging-industry manufacturing process steps and of furniture product delivery, respectively. The large-scale evaluation, conducted under supervised conditions, further reveals that it is able to outperform static baselines, realizing significant reductions in regression error by 43.85% and 60.82% in the two scenarios, respectively. This result qualitatively holds even if SAM-CDAR is reduced to the aforementioned nature-inspired method, i.e. being stripped of its CDDE component. Then, error reductions of still 43.40% and 60.42% can be observed. Utilizing the same configuration, more competitive baselines are outperformed by still considerable 11.47% and 6.18%. In summary, the evaluation provides evidence for effective concept drift adaptation in real-world industrial regression problems, which represents a critically under-explored domain.
The identified shortcomings and limitations point towards several improvement potentials, such as SAM-CDAR's selectivity in concept drift detection. In a broader scientific context, the gained insights significantly advance the practical applicability of CDDEs and regressor ensembles for adaptation means within evolving data streams. They also offer guidance for performance and resource-conscious decision regarding the use of ensembling in general as well as associated deployment considerations for real-world industrial environments.

9.2 Outlook

Beyond the contributions of this thesis, selected avenues for future research shall be highlighted while others can be obtained from the individual chapters. They address the further development of both the reference architecture for robust inference process systems and the contributions encompassed by the method SAM-CDAR and its components. Potentially, they may further advance the state of the art in continuously maintaining inference process systems in evolving industrial environments.

With regard to the reference architecture, one promising avenue concerns evaluating its usability for highly specific inference process system conditions that may require extensive modification. This includes e.g. modeling decentralized components of cloud-edge-distributed environments (Shayesteh et al. 2022) or federated learning scenarios featuring estimator artifacts being trained across different sites (Casado et al. 2023).
The reference architecture can additionally be aligned with ongoing industrial transformations. While many companies still implement Industry-4.0-directed solutions, the vision of Industry 5.0 augments it with new guiding aspects such as sustainability, human-centricity and resilience. It is vital for these companies to not abandon their approaches but to merely expand them accordingly (Xu et al. 2021). The proposed architecture would contribute to this goal by

- embedding human roles, including their required space for interactive learning and system control, explicitly through Subject Orientation.
- enabling more resilient machine learning deployments through concept drift adaptation components.
- supporting efficient innovation through front-loaded system design considerations.

Further advancing CDDEs constitutes another avenue. In this thesis, their potentials as concept-drift-detecting components for adaptation measures are demonstrated successfully. This lays the groundwork for additional exploration. For instance, comparative evaluations of different detectors as members are needed to better understand their individual and collective strengths. This explicitly includes unsupervised detectors and also such featuring novel input preprocessing approaches, as proposed in Section 8.4. Also, further approaches considering ensemble optimization and diversity-introducing measures, i.e. the use of ensemble meta algorithms (cf. Section 3.1.2) or dynamic ensemble adaptation are worth evaluating. The work on the nature-inspired machine learning ensemble method further motivates these directions by demonstrating how dynamic weighting, member cycling, and performance-oriented adjustments can be leveraged (Trat et al. 2024). This method, moreover, could be extended to classification problems in future works, which would make it available to an even wider spectrum of industrial domains.

Finally, the composite method SAM-CDAR itself provides a foundation for future exploration, especially regarding its robustness and generalizability. While the presented evaluation focuses on real-world industrial regression tasks, additional experiments could vary a range of aspects in a systematic and controlled fashion. This includes the delay of labels, the type of occurring concept drifts as well as other aspects.

A Appendix

A.1 Analysis of Publications on Robust Inference Process Systems

Table A.1: The attributes employed for analyses of publications on robust inference process systems.

Group	Attribute	Description
Scientific contribution	Contribution type	Nature of the publication’s contribution (e.g. abstract system description, productive system, algorithm, theoretical approach)
	Functional components	List of the functional components of the proposed concept drift adaptation solution
	Concept drift adaptation focus	Specific problem being solved via concept drift adaptation, if any
Modeling evaluation	Modeling method	Utilized modeling method, if any
	Coherence: Consistency	Consistent adherence to selected modeling method or element set if no method
	Coherence: Clarity	Tangibility of intricacies represented by model
	Coherence: Informative Content	Comprehensibility of model without need of explaining text
	Criticism	Other aspects limiting coherence
Context	Remark	Space for further custom remarks

A.2 Analysis of Publications on Concept Drift Detection Ensembles

Table A.2: The attributes employed for qualitative analysis of concept drift detection publications.

Group	Attribute	Description
Method	Name	Name of the CDDE
	Acronym	Name of the CDDE. If non-existent, it is created
Stream/Data	Labeling context	Supervisedness character (supervised, semi-supervised or unsupervised) of data streams targeted by the suggested CDDE
Concept drift target	Emergence type	Concept drift types distinguished within the study
	Subject type	Specialization of the CDDE on certain concept drift types
Base concept drift detector selection	Strategy	Type of strategy (e.g. manual selection)
	Variety	Variety-inducing type (homogeneous or heterogeneous)
	Variety via	Variety-inducing criteria (e.g. detector algorithm or parameters or data target)
	Algorithms/tests	Detector model algorithms
	Monitored score	Input data for detector
	Emission specialization	Different concept drift types that are signaled separately or localization
Ensemble	Meta algorithm	Application of ensemble meta algorithms (e.g. bagging, boosting, stacking)
	Aggregation/combination type	Type of aggregation regarding data fitting or deterministic application

	Aggregation/combination rule	The rule for combining detector outputs (e.g. majority vote, early find early report)
	Aggregation/combination of alert levels	The alert levels for grouping detections
	Aggregation/combination timing	The frequency concept drift detection is done with
	Processing batch size	Batch size received by the CDDE
	Ensemble-level parameters	Parameters of the CDDE
	Ensemble optimization/ diversity measures	Application of measures for continuous adaptation of the CDDE
Restrictions	Any	Restrictions for the application of the CDDE
Implementation	Parallelization	Measures for parallelization of CDDE processes
	Language/Framework	Programming language
	Code published	If and where the code is published
Theory	Definition	Definition attempts of CDDEs (None found)
	Motivation	Underlying motivation for combining detectors
Evaluation	Measurement	Method and score for CDDE performance evaluation
Additionals	Data type	The type of data (e.g. real or synthetic)
	Problem type	Classification or regression
	Concept drift adaptation	If and how concept drift adaptation is conducted
	Datasets	Names of datasets the CDDE is evaluated on
	Concept drift in data	If the presence of drift is known or synthesized

	Training of ensemble or base DD	If the CDDE is parametrized in a systematic way
Context	Citing related works	Citation of any other publications on CDDEs
	Highly related work	Work cited as basis for CDDE conception
	suggested future work	Suggestions of the publication
	Verdict	Concluding remarks on the performance and suitability of the proposed CDDE
	Remark	Space for further custom remarks

A.3 Details of the Industrial Scenarios

A.3.1 Lead Time Estimation Data

Table A.3: The features of the lead time estimation data with their respective descriptions and types.

Feature	Type
Production step identifier	categorical
Aggregated set of customers identifier	categorical
Product-related information	categorical
Material-related information	categorical
Aggregated set of machines identifier	categorical
Priority-related information	categorical
	categorical
	categorical
	categorical
Minimal material forming temperature	numerical
Maximal material forming temperature	numerical
Difference between minimal and maximal forming temperature	numerical
Specific weight of material	numerical
Average utilization of machines of a group	numerical
Attendance of personnel operating machines	numerical
Time between production registration and approval	numerical
Time between production registration and finalization	numerical
Time between production approval and finalization	numerical
Time between production registration and planned delivery	numerical
Time between production approval and planned delivery	numerical
Time between planned delivery and production finalization	numerical

A.3.2 Delivery Time Estimation Data

Table A.4: The features of the delivery time estimation data with their respective descriptions and types.

Feature	Type
Position of a product on an order	numerical
Position of a product on the associated invoice	numerical
Quantity of an ordered product	numerical
Quantity of an ordered product on the associated invoice	numerical
Price of product	numerical
Unique number of a product	categorical
Global location number of a buyer	categorical
Global location number of a supplier	categorical

Publications

Table A.5: List of publications involving the contribution of the thesis author.

Year	Title	Authors	Reference
2022	Benchmarking AutoML-Supported Lead Time Prediction	Janek Bender, Martin Trat, Jivka Ovtcharova	Bender et al. 2022
2022	Unsupervised Anomaly Detection and Root Cause Analysis for an Industrial Press Machine based on Skip-Connected Autoencoder	Chenwei Sun, Martin Trat, Janek Bender, Jivka Ovtcharova, George Jeppesen, Jan Bär	Sun et al. 2022
2023	Energy-Flexible Job-Shop Scheduling Using Deep Reinforcement Learning	Mine Felder, Daniel Steiner, Paul A. Busch, Martin Trat, Chenwei Sun, Janek Bender, Jivka Ovtcharova	Felder et al. 2023
2023	Towards a B2B integration framework for smart services in Industry 4.0	Viktor Schubert, Steffen Kühner, Tobias Krauss, Martin Trat, Janek Bender	Schubert et al. 2023
2023	Sensitivity-Based Optimization of Unsupervised Drift Detection for Categorical Data Streams	Martin Trat, Janek Bender, Jivka Ovtcharova	Trat et al. 2023
2023	Designing Concept Drift Detection Ensembles: A Survey	Martin Trat, Jivka Ovtcharova	Trat and Ovtcharova 2023

Year	Title	Authors	Reference
2024	Artificial-intelligence-enabled dynamic demand response system for maximizing the use of renewable electricity in production processes	Hendro Wicaksono, Martin Trat, Atit Bashyal, Tina Boroukhian, Mine Felder, Mischa Ahrens, Janek Bender, Sebastian Groß, Daniel Steiner, Christoph July, Christoph Dorus, Thorsten Zoerner	Wicaksono et al. 2024
2024	A Nature-Inspired Concept Drift Adaptation Method for Industrial Data Stream Regression	Martin Trat, Philipp Bergmann, Andreas Ott, Jivka Ovtcharova	Trat et al. 2024
2024	Self-supervised representation learning for robust fine-grained human hand action recognition in industrial assembly lines	Fabian Sturm, Martin Trat, Rahul Sathiyababu, Harshitha Allipilli, Benjamin Menz, Elke Hergenroether, Melanie Siegel	Sturm et al. 2024
2025	Modeling A Reference Architecture for Concept Drift Adaptation Systems	Martin Trat, Matthes Elstermann, Jana Deckers, Jivka Ovtcharova	Trat et al. 2025
2025	KI-gestützte Prognose von Durchlauf- und Lieferzeiten in der Einzel- und Kleinserienfertigung	Janek Bender, Martin Trat, Jivka Ovtcharova	Bender et al. 2025

References

Charu C. Aggarwal. On biased reservoir sampling in the presence of stream evolution. In *Proceedings of the 32nd International Conference on Very Large Data Bases*, VLDB ’06, pages 607–618. VLDB Endowment, 2006.

Supriya Agrahari and Anil Kumar Singh. Concept Drift Detection in Data Stream Mining: A literature review. *Journal of King Saud University - Computer and Information Sciences*, 34 (10):9523–9540, 2022. ISSN 13191578. doi: 10.1016/j.jksuci.2021.11.006.

Algorithmia Research. 2020 state of enterprise machine learning. Technical report, DataRobot, 2020. URL https://www.coriniumintelligence.com/2020-state-of-enterprise-machine-learning-algorithmia-whitepaper-download.

Saleema Amershi, Andrew Begel, Christian Bird, Robert DeLine, Harald Gall, Ece Kamar, Nachiappan Nagappan, Besmira Nushi, and Thomas Zimmermann. Software Engineering for Machine Learning: A Case Study. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pages 291–300. IEEE, 2019. ISBN 978-1-7281-1760-7. doi: 10.1109/ICSE-SEIP.2019.00042.

Anaconda Inc. State of Data Science 2021: On the Path to Impact, 2021. URL https://www.anaconda.com/state-of-data-science-2021.

Ature Angbera and Huah Yong Chan. SABeDM: a sliding adaptive beta distribution model for concept drift detection in a dynamic environment. *Knowledge and Information Systems*, 66(3): 2039–2062, 2024. ISSN 0219-1377. doi: 10.1007/s10115-023-02004-3.

Mariette Awad and Rahul Khanna. *Efficient learning machines: Theories, concepts, and applications for engineers and system designers*. The expert’s voice in machine learning. Apress Open, Berkley, 2015. ISBN 978-1-4302-5989-3. doi: 10.1007/978-1-4302-5990-9.

AWS. Well-Architected machine learning lifecycle: Machine Learning Lens, 2023a. URL https://docs.aws.amazon.com/wellarchitected/latest/machine-learning-lens/well-architected-machine-learning-lifecycle.html.

AWS. Machine learning lifecycle: Machine Learning Best Practices in Healthcare and Life Sciences, 2023b. URL `https://docs.aws.amazon.com/whitepapers/latest/ml-best-practices-healthcare-life-sciences/machine-learning-lifecycle.html`.

Stephen H. Bach and Marcus A. Maloof. Paired Learners for Concept Drift. In *8th IEEE International Conference on Data Mining*, pages 23–32. IEEE, 2008. ISBN 978-0-7695-3502-9. doi: 10.1109/ICDM.2008.119.

Lucas Baier, Marcel Hofmann, Niklas Kühl, Marisa Mohr, and Gerhard Satzger. Handling Concept Drifts in Regression Problems - the Error Intersection Approach. In Norbert Gronau, Moreen Heine, Hanna Krasnova, and Key Pousttchi, editors, *Proceedings der 15. Internationalen Tagung Wirtschaftsinformatik 2020*, Entwicklungen, Chancen und Herausforderungen der Digitalisierung / Gronau, N., Heine, M., Krasnova, H., Pousttchi, K. (Hrsg.), pages 210–224. GITO Verlag, Berlin, 2020. ISBN 9783955453350.

Roberto Souto Maior Barros and Silas Garrido T. Carvalho Santos. A large-scale comparison of concept drift detectors. *Information Sciences*, 451-452:348–370, 2018. ISSN 00200255. doi: 10.1016/j.ins.2018.04.014.

Firas Bayram, Bestoun S. Ahmed, and Andreas Kassler. From concept drift to model degradation: An overview on performance-aware drift detectors. *Knowledge-Based Systems*, 245:108632, 2022. ISSN 09507051. doi: 10.1016/j.knosys.2022.108632.

Janek Bender. *AutoML-Supported Lead Time Prediction Enabling Smart Job Scheduling in Make-To-Order Production*. Karlsruher Institut für Technologie (KIT), 2024. doi: 10.5445/IR/1000175793.

Janek Bender, Martin Trat, and Jivka Ovtcharova. Benchmarking AutoML-Supported Lead Time Prediction. *Procedia Computer Science*, 200:482–494, 2022. ISSN 18770509. doi: 10.1016/j.procs.2022.01.246.

Janek Bender, Martin Trat, and Jivka Ovtcharova. KI-gestützte Prognose von Durchlauf- und Lieferzeiten in der Einzel- und Kleinserienfertigung. *Zeitschrift für wirtschaftlichen Fabrikbetrieb*, 120(s1):292–296, 2025. ISSN 0947-0085. doi: 10.1515/zwf-2024-0162.

Janine M. Benyus. *Biomimicry: Innovation inspired by nature*. Harpercollins, New York, N.Y., reissue edition, 2002. ISBN 9780060533229.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(10):281–305, 2012.

A. Bifet. Classifier Concept Drift Detection and the Illusion of Progress. In L. Rutkowski, M. Korytkowski, and R. Scherer, editors, *Artificial Intelligence and Soft Computing*, Lecture

Notes in Computer Science, pages 715–725. Springer, 2017. doi: 10.1007/978-3-319-59060-8_64.

Albert Bifet and Ricard Gavaldà. Learning from Time-Changing Data with Adaptive Windowing. In Chid Apte, David Skillicorn, Bing Liu, and Srinivasan Parthasarathy, editors, *Proceedings of the Seventh SIAM International Conference on Data Mining*, pages 443–448, Philadelphia, Pa., 2007. Soc. for Industrial and Applied Mathematics. ISBN 978-0-89871-630-6. doi: 10.1137/1.9781611972771.42.

Albert Bifet, Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Indrė Žliobaitė. CD-MOA: Change Detection Framework for Massive Online Analysis. In Allan Tucker, Frank Höppner, Arno Siebes, and Stephen Swift, editors, *Advances in Intelligent Data Analysis XII*, pages 92–103, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-41398-8.

Albert Bifet, Ricard Gavaldà, Geoff Holmes, and Bernhard Pfahringer. *Machine Learning for Data Streams: With Practical Examples in MOA*. The MIT Press, Cambridge, 2018. ISBN 9780262346047. doi: 10.7551/mitpress/10654.001.0001.

Jakob Bönsch, Matthes Elstermann, Andreas Kimmig, and Jivka Ovtcharova. A subject-oriented reference model for Digital Twins. *Computers & Industrial Engineering*, 172, 2022. ISSN 03608352. doi: 10.1016/j.cie.2022.108556.

Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001. ISSN 0885-6125. doi: 10.1023/A:1010933404324.

Li Bu, Cesare Alippi, and Dongbin Zhao. Ensemble LSDD-based change detection tests. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 4064–4069, Piscataway, NJ, 2016. IEEE. ISBN 978-1-5090-0620-5. doi: 10.1109/IJCNN.2016.7727728.

Fernando E. Casado, Dylan Lema, Roberto Iglesias, Carlos V. Regueiro, and Senén Barro. Ensemble and continual federated learning for classification tasks. *Machine Learning*, 112(9): 3413–3453, 2023. ISSN 0885-6125. doi: 10.1007/s10994-023-06330-z.

Rodolfo C. Cavalcante and Adriano L. I. Oliveira. An approach to handle concept drift in financial time series based on Extreme Learning Machines and explicit Drift Detection. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE / Institute of Electrical and Electronics Engineers Incorporated, 2015. ISBN 978-1-4799-1960-4. doi: 10.1109/IJCNN.2015.7280721.

Rodolfo C. Cavalcante, Leandro L. Minku, and Adriano L. I. Oliveira. FEDD: Feature Extraction for Explicit Concept Drift Detection in time series. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 740–747, Piscataway, NJ, 2016. IEEE. ISBN 978-1-5090-0620-5. doi: 10.1109/IJCNN.2016.7727274.

Vitor Cerqueira, Heitor Murilo Gomes, Albert Bifet, and Luis Torgo. STUDD: a student–teacher method for unsupervised concept drift detection. *Machine Learning*, 2022. ISSN 0885-6125. doi: 10.1007/s10994-022-06188-7.

Aditya Challapally, Chris Pease, Ramesh Raskar, and Pradyumna Chari. The GenAI Divide: State of AI in Business 2025. Preliminary Findings from AI Implementation Research from Project NANDA, 2025. URL `https://nanda.media.mit.edu/`.

Pete Chapman, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer, and Rüdiger Wirth. CRISP-DM 1.0: Step-by-step data mining guide, 2000. URL `https://www.kde.cs.uni-kassel.de/wp-content/uploads/lehre/ws2012-13/kdd/files/CRISPWP-0800.pdf`.

Michael Chui, Bryce Hall, Helen Mayhew, and Alex Singla. The state of AI in 2022—and a half decade in review, 2022. URL `https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai-in-2022-and-a-half-decade-in-review`.

F. G. da Costa, R. A. Rios, and R. F. de Mello. Using dynamical systems tools to detect concept drift in data streams. *Expert Systems with Applications*, 60:39–50, 2016. ISSN 0957-4174. doi: 10.1016/j.eswa.2016.04.026.

Roberto Souto Maior de Barros and Silas Garrido T. de Carvalho Santos. An overview and comprehensive comparison of ensembles for concept drift. *Information Fusion*, 52:213–244, 2019. ISSN 15662535. doi: 10.1016/j.inffus.2019.03.006.

Magdalena Deckert. Batch Weighted Ensemble for Mining Data Streams with Concept Drift. In Marzena Kryszkiewicz, Henryk Rybinski, Andrzej Skowron, and Zbigniew W. Raś, editors, *Foundations of Intelligent Systems*, volume 6804 of *Lecture Notes in Computer Science*, pages 290–299. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-21915-3. doi: 10.1007/978-3-642-21916-0_32.

Salah Ud Din, Aman Ullah, Cobbinah B. Mawuli, Qinli Yang, and Junming Shao. A reliable adaptive prototype-based learning for evolving data streams with limited labels. *Information Processing & Management*, 61(1):103532, 2024. ISSN 03064573. doi: 10.1016/j.ipm.2023.103532.

Zhongyi Ding, Shujie Yang, Zhaoyang Liu, Tengchao Ma, Zichen Feng, and Mingze Wang. CD-SR: A Real-time Anomaly Detection Framework for Continuous Concept Drift. In *2021 International Conference on Networking and Network Applications (NaNA)*, pages 194–199. IEEE, 2021. ISBN 978-1-6654-4158-2. doi: 10.1109/NaNA53684.2021.00040.

Pedro Domingos and Geoff Hulten. Mining high-speed data streams. In Raghu Ramakrishnan, Sal Stolfo, Roberto Bayardo, and Ismail Parsa, editors, *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 71–80, New York, NY, USA, 2000. ACM. ISBN 1581132336. doi: 10.1145/347090.347107.

Fan Dong, Jie Lu, Yiliao Song, Feng Liu, and Guangquan Zhang. A Drift Region-Based Data Sample Filtering Method. *IEEE transactions on cybernetics*, 52(9):9377–9390, 2022. doi: 10.1109/TCYB.2021.3051406.

L. Du, Q. Song, L. Zhu, and X. Zhu. A Selective Detector Ensemble for Concept Drift Detection. *The Computer Journal*, 58(3):457–471, 2015. ISSN 0010-4620. doi: 10.1093/comjnl/bxu050.

Elsevier. Scopus: Access and use Support Center: How do I search in Scopus?, 2023. URL `https://service.elsevier.com/app/`.

Matthes Elstermann. *Executing Strategic Product Planning - A Subject-Oriented Analysis and New Referential Process Model for IT-Tool Support and Agile Execution of Strategic Product Planning*. Dissertation, Karlsruhe Institute of Technology, Karlsruhe, 2020. URL `https://dx.doi.org/10.5445/KSP/1000097859`.

Matthes Elstermann and Jivka Ovtcharova. Sisi in the ALPS: A Simple Simulation and Verification Approach for PASS. In Christian Stary, editor, *Proceedings of the 10th International Conference on Subject-Oriented Business Process Management*, pages 1–9, New York, NY, USA, 2018. ACM. ISBN 9781450353601. doi: 10.1145/3178248.3178262.

Matthes Elstermann, Jakob Bönsch, Andreas Kimmig, and Jivka Ovtcharova. Human-Centered Referential Process Models for AI Application. In Zimmermann, editor, *Human Centred Intelligent Systems*, volume 244 of *Smart Innovation, Systems and Technologies*, pages 56–65. Springer Singapore, 2021. ISBN 978-981-16-3263-1. doi: 10.1007/978-981-16-3264-8_6.

Conor Fahy, Shengxiang Yang, and Mario Gongora. Scarcity of Labels in Non-Stationary Data Streams: A Survey. *ACM Computing Surveys*, 55(2):1–39, 2023. ISSN 0360-0300. doi: 10.1145/3494832.

Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006. ISSN 01678655. doi: 10.1016/j.patrec.2005.10.010.

Mine Felder, Daniel Steiner, Paul Busch, Martin Trat, Chenwei Sun, Janek Bender, and Jivka Ovtcharova. Energy-Flexible Job-Shop Scheduling Using Deep Reinforcement Learning. In David Herberger, Marco Hübner, and Volker Stich, editors, *Proceedings of the Conference on Production Systems and Logistics (CPSL) 2023*, pages 353–362. publish-Ing, Hannover, 2023. doi: 10.15488/13454.

Albert Fleischmann, Werner Schmidt, Christian Stary, Stefan Obermeier, and Egon Börger. *Subject-Oriented Business Process Management*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-32391-1. doi: 10.1007/978-3-642-32392-8.

Teodor Fredriksson, David Issa Mattos, Jan Bosch, and Helena Holmström Olsson. Data Labeling: An Empirical Investigation into Industrial Challenges and Mitigation Strategies. In Maurizio Morisio, Marco Torchiano, and Andreas Jedlitschka, editors, *Product-Focused Software Process Improvement*, volume 12562 of *Lecture Notes in Computer Science*, pages 202–216. Springer, Cham, 2021. ISBN 978-3-030-64147-4. doi: 10.1007/978-3-030-64148-1_13.

João Gama, Pedro Medas, Gladys Castillo, and Pedro Rodrigues. Learning with Drift Detection. In David Hutchison, editor, *Advances in Artificial Intelligence - SBIA 2004*, volume 3171 of *Lecture Notes in Computer Science*, pages 286–295. Springer Berlin, Heidelberg, 2004. ISBN 978-3-540-23237-7. doi: 10.1007/978-3-540-28645-5_29.

João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):1–37, 2014. ISSN 0360-0300. doi: 10.1145/2523813.

Rosana Noronha Gemaque, Albert França Josuá Costa, Rafael Giusti, and Eulanda Miranda Santos. An overview of unsupervised drift detection methods. *WIREs Data Mining and Knowledge Discovery*, 10(6), 2020. ISSN 1942-4787. doi: 10.1002/widm.1381.

Heitor Murilo Gomes, Jean Paul Barddal, Luis Eduardo Boiko Ferreira, and Albert Bifet. Adaptive random forests for data stream regression. In *26th European Symposium on Artificial Neural Networks*, 2018.

Heitor Murilo Gomes, Jesse Read, and Albert Bifet. Streaming Random Patches for Evolving Data Stream Classification. In *2019 IEEE International Conference on Data Mining*, pages 240–249. IEEE, 2019. ISBN 978-1-7281-4604-1. doi: 10.1109/ICDM.2019.00034.

Heitor Murilo Gomes, Jacob Montiel, Saulo Martiello Mastelini, Bernhard Pfahringer, and Albert Bifet. On Ensemble Techniques for Data Stream Regression. In *2020 International Joint Conference on Neural Networks*, pages 1–8. IEEE, 2020. ISBN 978-1-7281-6926-2. doi: 10.1109/IJCNN48605.2020.9206756.

Josef Gramespacher, Werner Breisacher, Alexander Essig, Janek Bender, and Matthias Gloß. Alto - Algorithmengestützte Optimierung der termingerechten Auftragssteuerung für die Organisationsabläufe der Einzelfertigung. Technical report, Technische Informationsbibliothek, Hannover, 2022.

Kanika Gupta, Saloni Rathare, Sachin Sharma, Vineet Dahiya, Pushpa Negi, and Anishkumar Dhablia. Future Industrial Production Work Designusing Clustering Approach: Transitioning to Cyber Physical Systems. In *2023 International Conference on Data Science and Network Security (ICDSNS)*, pages 1–8. IEEE, 2023. ISBN 979-8-3503-0159-5. doi: 10.1109/ICDSNS58469.2023.10245953.

Constanze Hasterok, Janina Stompe, Julius Pfrommer, Thomas Usländer, Jens Ziehn, Sebastian Reiter, Michael Weber, and Till Riedel. PAISE: Das Vorgehensmodell für KI-Engineering, 2021. URL `https://www.ki-engineering.eu/de/wissen-tools/paise.html`.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer Series in Statistics. Springer New York, New York, NY, 2nd edition edition, 2009. ISBN 978-0-387-84857-0. doi: 10.1007/b94608.

Hanqing Hu. *Solving the challenges of concept drift in data stream classification.* Doctoral Dissertation, University of Louisville, Louisville, Kentucky, USA, 2022.

Hanqing Hu and Mehmed Kantardzic. Heuristic ensemble for unsupervised detection of multiple types of concept drift in data stream classification. *Intelligent Decision Technologies*, 15(4): 609–622, 2022. ISSN 18724981. doi: 10.3233/IDT-210115.

Hanqing Hu, Mehmed Kantardzic, and Lingyu Lyu. Detecting Different Types of Concept Drifts with Ensemble Framework. In M. Arif Wani, Mehmed Kantardzic, Moamar Sayed-Mouchaweh, João Gama, and Edwin Lughofer, editors, *17th IEEE International Conference on Machine Learning and Applications*, pages 344–350, Piscataway, NJ, 2018. IEEE. ISBN 978-1-5386-6805-4. doi: 10.1109/ICMLA.2018.00058.

Hanqing Hu, Mehmed Kantardzic, and Tegjyot S. Sethi. No Free Lunch Theorem for concept drift detection in streaming data classification: A review. *WIREs Data Mining and Knowledge Discovery*, 10(2), 2020. ISSN 1942-4787. doi: 10.1002/widm.1327.

Geoff Hulten, Laurie Spencer, and Pedro Domingos. Mining time-changing data streams. In Foster Provost and Ramakrishnan Srikant, editors, *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 97–106, New York, NY, USA, 2001. ACM Press. ISBN 158113391X. doi: 10.1145/502512.502529.

Rob J. Hyndman and Anne B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688, 2006. ISSN 01692070. doi: 10.1016/j.ijforecast.2006.03.001.

IBM. Analytics Solutions Unified Method: Implementations with Agile principles, 2016.

Elena Ikonomovska, João Gama, Raquel Sebastião, and Dejan Gjorgjevik. Regression Trees from Data Streams with Drift Detection. In João Gama, editor, *Discovery science*, volume 5808 of *LNCS sublibrary*, pages 121–135. Springer, Berlin, 2009. ISBN 978-3-642-04746-6. doi: 10.1007/978-3-642-04747-3_12.

Adriana Sayuri Iwashita and Joao Paulo Papa. An Overview on Concept Drift Learning. *IEEE Access*, 7:1532–1547, 2019. doi: 10.1109/ACCESS.2018.2886026.

Ghazal Jaber, Antoine Cornuéjols, and Philippe Tarroux. Online Learning: Searching for the Best Forgetting Strategy under Concept Drift. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, Minho Lee, Akira Hirose, Zeng-Guang Hou, and Rhee Man Kil, editors, *Neural Information Processing: 20th International Conference*, volume 8227 of *Lecture Notes in Computer Science / Theoretical Computer Science and General Issues*, pages 400–408. Springer, Berlin, 2013. ISBN 978-3-642-42041-2. doi: 10.1007/978-3-642-42042-9_50.

Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, and Jonathan Taylor. *An Introduction to Statistical Learning*. Springer International Publishing, Cham, 2023. ISBN 978-3-031-38746-3. doi: 10.1007/978-3-031-38747-0.

Haseeb Javed, Shaker El-Sappagh, and Tamer Abuhmed. Robustness in deep learning models for medical diagnostics: security and adversarial challenges towards robust AI applications. *Artificial Intelligence Review*, 58(1), 2025. doi: 10.1007/s10462-024-11005-9.

Xuancheng Jin and Yaying Zhang. Adaptive Random Forest with Dynamic Detectors for Evolving Data Stream Classification. In *Proceedings of the 9th International Conference on Computing and Artificial Intelligence*, pages 678–684, New York, NY, USA, 2023. ACM. ISBN 9781450399029. doi: 10.1145/3594315.3594390.

Imen Khamassi, Moamar Sayed-Mouchaweh, Moez Hammami, and Khaled Ghédira. Discussion and review on evolving data streams and concept drift adapting. *Evolving Systems*, 9(1):1–23, 2018. ISSN 1868-6478. doi: 10.1007/s12530-016-9168-2.

Joanna Komorniczak, Paweł Zyblewski, and Paweł Ksieniewicz. Statistical Drift Detection Ensemble for batch processing of data streams. *Knowledge-Based Systems*, 252, 2022. ISSN 09507051. doi: 10.1016/j.knosys.2022.109380.

Lukasz Korycki and Bartosz Krawczyk. Unsupervised Drift Detector Ensembles for Data Stream Mining. In Lisa Singh, editor, *2019 IEEE International Conference on Data Science and*

Advanced Analytics, pages 317–325, Piscataway, NJ, 2019. IEEE. ISBN 978-1-7281-4493-1. doi: 10.1109/DSAA.2019.00047.

Julian Kraus. *Machine Learning in Industrial Applications: Detector Ensembles for Unsupervised Concept Drift Identification in Data Streams*. Master’s Thesis, Karlsruhe Institute of Technology, Karlsruhe, 2025.

Bartosz Krawczyk, Leandro L. Minku, João Gama, Jerzy Stefanowski, and Michał Woźniak. Ensemble learning for data stream analysis: A survey. *Information Fusion*, 37:132–156, 2017. ISSN 15662535. doi: 10.1016/j.inffus.2017.02.004.

Dominik Kreuzberger, Niklas Kühl, and Sebastian Hirschl. Machine Learning Operations (MLOps): Overview, Definition, and Architecture. *IEEE Access*, 11:31866–31879, 2023. doi: 10.1109/ACCESS.2023.3262138.

Cedric Kulbach, Jacob Montiel, Maroua Bahri, Marco Heyden, and Albert Bifet. Evolution-Based Online Automated Machine Learning. In João Gama, Tianrui Li, Yang Yu, Enhong Chen, Yu Zheng, and Fei Teng, editors, *Advances in Knowledge Discovery and Data Mining*, volume 13280 of *Springer eBook Collection*, pages 472–484. Springer International Publishing and Imprint Springer, Cham, 2022. ISBN 978-3-031-05932-2. doi: 10.1007/978-3-031-05933-9_37.

Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, Hoboken, NJ, 2004. ISBN 9780471210788. doi: 10.1002/0471660264.

Victor Kurtz, Jakob Bönsch, and Jivka Ovtcharova. Understanding and Harnessing the Potential of Conversational AI for S-BPM. In Matthes Elstermann, editor, *Subject-Oriented Business Process Management*, volume 1632 of *Communications in Computer and Information Science*, pages 41–57. Springer, Cham, 2022. ISBN 978-3-031-19703-1. doi: 10.1007/978-3-031-19704-8_3.

M. Kwak, J. Nam, K. Kwak, G. Kim, D. Choi, J. Jung, J. Han, G. Kang, K. Lim, Y. Byun, J. Eum, M. H. Azarian, and N. Lee. Development of PHM Algorithm of e-Latch to Prepare for the Era of Autonomous Driving. *Annual Conference of the PHM Society*, 16(1), 2024. doi: 10.36001/phmconf.2024.v16i1.4061.

Andreas Lange. *Introducing formalized Concept Drift to Probabilistic Graphical Models of Industrial Process Data*. Bachelor’s Thesis, Karlsruhe Institute of Technology, Karlsruhe, 2023.

Andrzej Lapinski, Bartosz Krawczyk, Pawel Ksieniewicz, and Michał Woźniak. An Empirical Insight Into Concept Drift Detectors Ensemble Strategies. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, Piscataway, NJ, 2018. IEEE. ISBN 978-1-5090-6017-7. doi: 10.1109/CEC.2018.8477962.

Vincent Lemaire, Christophe Salperwyck, and Alexis Bondu. A Survey on Supervised Classification on Data Streams. In Esteban Zimányi and Ralf-Detlef Kutsche, editors, *Business Intelligence*, volume 205 of *Lecture Notes in Business Information Processing*, pages 88–125. Springer International Publishing, Cham, 2015. ISBN 978-3-319-17550-8. doi: 10.1007/978-3-319-17551-5_4.

Haoli Li and Tao Zhao. A dynamic similarity weighted evolving fuzzy system for concept drift of data streams. *Information Sciences*, 659:120062, 2024. ISSN 00200255. doi: 10.1016/j.ins.2023.120062.

Zhi Li. Navigating Digital Transformation: A Risk-Based Approach for Industry 4.0 Innovation. *Journal of the Knowledge Economy*, 2024. doi: 10.1007/s13132-024-02264-6.

Marília Lima, Telmo Silva Filho, and Roberta Andrade de A. Fagundes. A Comparative Study on Concept Drift Detectors for Regression. In André Britto and Karina Valdivia Delgado, editors, *Intelligent Systems*, volume 13073 of *Lecture Notes in Computer Science*, pages 390–405. Springer International Publishing, Cham, 2021. ISBN 978-3-030-91701-2. doi: 10.1007/978-3-030-91702-9_26.

Marília Lima, Manoel Neto, Telmo Silva Filho, and Roberta A. de A. Fagundes. Learning Under Concept Drift for Regression—A Systematic Literature Review. *IEEE Access*, 10: 45410–45429, 2022. doi: 10.1109/ACCESS.2022.3169785.

Zongying Liu, Chu Kiong Loo, and Manjeevan Seera. Meta-cognitive Recurrent Recursive Kernel OS-ELM for concept drift handling. *Applied Soft Computing*, 75:494–507, 2019. doi: 10.1016/j.asoc.2018.11.006.

Jie Lu, Anjin Liu, Fan Dong, Feng Gu, João Gama, and Guangquan Zhang. Learning under Concept Drift: A Review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12): 2346–2363, 2018. ISSN 1041-4347. doi: 10.1109/TKDE.2018.2876857.

Bruno Iran Ferreira Maciel, Silas Garrido Teixeira Carvalho Santos, and Roberto Souto Maior Barros. A Lightweight Concept Drift Detection Ensemble. In *2015 IEEE 27th International Conference on Tools with Artificial Intelligence - ICTAI 2015*, pages 1061–1068, Piscataway, NJ, 2015. IEEE. ISBN 978-1-5090-0163-7. doi: 10.1109/ICTAI.2015.151.

Osama A. Mahdi, Nawfal Ali, Eric Pardede, Ammar Alazab, Tahsien Al-Quraishi, and Bhagwan Das. Roadmap of Concept Drift Adaptation in Data Stream Mining, Years Later. *IEEE Access*, 12:21129–21146, 2024. doi: 10.1109/ACCESS.2024.3358817.

Gaetano Marino, Giulio Zotteri, and Francesca Montagna. Consumer sensitivity to delivery lead time: a furniture retail case. *International Journal of Physical Distribution & Logistics Management*, 48(6):610–629, 2018. ISSN 0960-0035. doi: 10.1108/IJPDLM-01-2017-0030.

Mansour Zoubeirou A. Mayaki and Michel Riveill. Autoregressive based Drift Detection Method. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022. ISBN 978-1-7281-8671-9. doi: 10.1109/IJCNN55064.2022.9892066.

Leandro L. Minku and Xin Yao. DDD: A New Ensemble Approach for Dealing with Concept Drift. *IEEE Transactions on Knowledge and Data Engineering*, 24(4):619–633, 2012. ISSN 1041-4347. doi: 10.1109/TKDE.2011.58.

Tom M. Mitchell. *Machine learning*. McGraw-Hill series in computer science. McGraw-Hill, New York, NY, international ed. edition, 1997. ISBN 0070428077.

Jacob Montiel, Max Halford, Saulo Martiello Mastelini, Geoffrey Bolmier, Raphael Sourty, Robin Vaysse, Adil Zouitine, Heitor Murilo Gomes, Jesse Read, Talel Abdessalem, and Albert Bifet. River: machine learning for streaming data in Python. *Journal of Machine Learning Research*, 22(110):1–8, 2021. URL `http://jmlr.org/papers/v22/20-1380.html`.

Khanh-Tung Nguyen, Trung Tran, Anh-Duc Nguyen, Xuan-Hieu Phan, and Quang-Thuy Ha. Parameter Distribution Ensemble Learning for Sudden Concept Drift Detection. In Ngoc Thanh Nguyen, Tien Khoa Tran, Ualsher Tukayev, Tzung-Pei Hong, Bogdan Trawiński, and Edward Szczerbicki, editors, *Intelligent Information and Database Systems*, volume 13758 of *Lecture Notes in Computer Science*, pages 192–203. Springer, Cham, 2022. ISBN 978-3-031-21966-5. doi: 10.1007/978-3-031-21967-2_16.

Kyosuke Nishida, Koichiro Yamauchi, and Takashi Omori. ACE: Adaptive Classifiers-Ensemble System for Concept-Drifting Environments. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, Nikunj C. Oza, Robi Polikar, and Fabio Roli, editors, *Multiple classifier systems*, volume 3541 of *Lecture Notes in Computer Science*, pages 176–185. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. ISBN 978-3-540-26306-7. doi: 10.1007/11494683_18.

Yoshihiro Okawa and Kenichi Kobayashi. Concept Drift Detection via Boundary Shrinking. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, Piscataway, NJ, USA, 2021. IEEE. ISBN 978-1-6654-3900-8. doi: 10.1109/IJCNN52387.2021.9533334.

Andreas Ott. *Robuste Ensemble-Methoden für Datenstromszenarien*. Bachelor's Thesis, Karlsruhe Institute of Technology, Karlsruhe, 2024.

Jivka Ovtcharova. On the three pillars of front-loading: personal communication, 05.05.2025.

Jivka Ovtcharova, Polina Häfner, Victor Häfner, Jurica Katicic, and Christina Vinke. Innovation braucht Resourceful Humans Aufbruch in eine neue Arbeitskultur durch Virtual Engineering. In Alfons Botthof and Ernst Andreas Hartmann, editors, *Zukunft der Arbeit in Industrie 4.0*, pages 111–124. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015. ISBN 978-3-662-45914-0. doi: 10.1007/978-3-662-45915-7_12.

E. S. Page. Continuous Inspection Schemes. *Biometrika*, 41(1-2):100–115, 1954. ISSN 0006-3444. doi: 10.1093/biomet/41.1-2.100.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011. URL `http://jmlr.org/papers/v12/pedregosa11a.html`.

Jose Luis M. Perez, Roberto S.M. Barros, and Silas G.T.C. Santos. Statistical Tests Ensemble Drift Detector. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1021–1028, Piscataway, NJ, 2020. IEEE. ISBN 978-1-7281-2547-3. doi: 10.1109/SSCI47803.2020.9308267.

Ali Pesaranghader, Herna Viktor, and Eric Paquet. Reservoir of diverse adaptive learners and stacking fast hoeffding drift detection methods for evolving data streams. *Machine Learning*, 107(11):1711–1743, 2018. ISSN 0885-6125. doi: 10.1007/s10994-018-5719-z.

Fábio Pinto, Marco O. P. Sampaio, and Pedro Bizarro. Automatic Model Monitoring for Data Streams. *KDD-ADF-*, 2019. URL `https://arxiv.org/pdf/1908.04240`.

Robi Polikar. Ensemble Learning. In Cha Zhang and Yunqian Ma, editors, *Ensemble Machine Learning*, pages 1–34. Springer, New York and Heidelberg, 2012. ISBN 978-1-4419-9325-0. doi: 10.1007/978-1-4419-9326-7_1.

Foster Provost and Tom Fawcett. *Data science for business: What you need to know about data mining and data-analytic thinking*. O'Reilly, Sebastopol, CA, USA, first edition edition, 2013. ISBN 9781449374297.

A. F. R. Rahman, H. Alam, and M. C. Fairhurst. Multiple Classifier Combination for Character Recognition: Revisiting the Majority Voting System and Its Variations. In Daniel Philip Lopresti, Jianying Hu, and Ramanujan Kashi, editors, *Document Analysis Systems V*, volume 2423 of *Lecture Notes in Computer Science*, pages 167–178. Springer, New York, 2002. ISBN 978-3-540-44068-0. doi: 10.1007/3-540-45869-7_21.

Jesse Read and Indrė Žliobaitė. Learning from Data Streams: An Overview and Update, 2023. URL `https://dx.doi.org/10.2139/ssrn.4326595`.

Jesse Read, Albert Bifet, Bernhard Pfahringer, and Geoff Holmes. Batch-Incremental versus Instance-Incremental Learning in Dynamic and Evolving Data. In Jaakko Hollmén, F. Klawonn, and Allan Tucker, editors, *Advances in intelligent data analysis XI*, volume 7619 of *LNCS sublibrary. SL 3, Information systems and application, incl. Internet/Web, and HCI*, pages 313–323. Springer, Heidelberg, 2012. ISBN 978-3-642-34155-7. doi: 10.1007/978-3-642-34156-4_29.

A. Reidt, M. Pfaff, and H. Krcmar. Der Referenzarchitekturbegriff im Wandel der Zeit. *HMD Praxis der Wirtschaftsinformatik*, 55(5):893–906, 2018. ISSN 1436-3011. doi: 10.1365/s40702-018-00448-8.

Andreas Reidt. *Referenzarchitektur eines integrierten Informationssystems zur Unterstützung der Instandhaltung*. PhD thesis, Technische Universität München, 2019. URL `https://mediatum.ub.tum.de/1443408`.

Gordon J. Ross, Niall M. Adams, Dimitris K. Tasoulis, and David J. Hand. Exponentially weighted moving average charts for detecting concept drift. *Pattern Recognition Letters*, 33(2):191–198, 2012. ISSN 01678655. doi: 10.1016/j.patrec.2011.08.019.

Jeffrey C. Schlimmer and Richard H. Granger. Incremental learning from noisy data. *Machine Learning*, 1(3):317–354, 1986. ISSN 0885-6125. doi: 10.1007/BF00116895.

Viktor Schubert, Steffen Kuehner, Tobias Krauss, Martin Trat, and Janek Bender. Towards a B2B integration framework for smart services in Industry 4.0. *Procedia Computer Science*, 217: 1649–1659, 2023. ISSN 18770509. doi: 10.1016/j.procs.2022.12.365.

D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, and Dan Dennison. Hidden Technical Debt in Machine Learning Systems. *Advances in Neural Information Processing Systems 28*, pages 2503–2511, 2015.

Tegjyot Singh Sethi and Mehmed Kantardzic. Don't Pay for Validation: Detecting Drifts from Unlabeled data Using Margin Density. *Procedia Computer Science*, 53:103–112, 2015. ISSN 18770509. doi: 10.1016/j.procs.2015.07.284.

Behshid Shayesteh, Chunyan Fu, Amin Ebrahimzadeh, and Roch H. Glitho. Automated Concept Drift Handling for Fault Prediction in Edge Clouds Using Reinforcement Learning. *IEEE Transactions on Network and Service Management*, 19(2):1321–1335, 2022. doi: 10.1109/TNSM.2022.3153279.

Symone G. Soares and Rui Araújo. A dynamic and on-line ensemble regression for changing environments. *Expert Systems with Applications*, 42(6):2935–2948, 2015. ISSN 0957-4174. doi: 10.1016/j.eswa.2014.11.053.

Piotr Sobolewski and Michał Woźniak. Concept Drift Detection and Model Selection with Simulated Recurrence and Ensembles of Statistical Detectors. *Journal of Universal Computer Science*, 19(4):462–483, 2013. doi: 10.3217/JUCS-019-04-0462.

Yiliao Song, Jie Lu, Haiyan Lu, and Guangquan Zhang. Learning Data Streams With Changing Distributions and Temporal Dependency. *IEEE transactions on neural networks and learning systems*, 34(8):3952–3965, 2023. doi: 10.1109/TNNLS.2021.3122531.

Patrick Sönke and Martin Trat. Holistische Customer-Experience-Management-Plattform zur nachhaltigen Optimierung der Wertschöpfungskette in der Möbelbranche (furnFUSION). Technical report, Technische Informationsbibliothek, Hannover, 2025. URL `https://doi.org/10.34657/22654`.

W. Nick Street and YongSeog Kim. A streaming ensemble algorithm (SEA) for large-scale classification. In Foster Provost and Ramakrishnan Srikant, editors, *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 377–382, New York, NY, USA, 2001. ACM Press. ISBN 158113391X. doi: 10.1145/502512.502568.

Stefan Studer, Thanh Binh Bui, Christian Drescher, Alexander Hanuschkin, Ludwig Winkler, Steven Peters, and Klaus-Robert Müller. Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology. *Machine Learning and Knowledge Extraction*, 3(2):392–413, 2021. doi: 10.3390/make3020020.

Fabian Sturm, Martin Trat, Rahul Sathiyababu, Harshitha Allipilli, Benjamin Menz, Elke Hergenroether, and Melanie Siegel. Self-supervised representation learning for robust fine-grained human hand action recognition in industrial assembly lines. *Machine Vision and Applications*, 36(19), 2024. ISSN 1432-1769. doi: 10.1007/s00138-024-01638-9.

Chenwei Sun, Martin Trat, Janek Bender, Jivka Ovtcharova, George Jeppesen, and Jan Bär. Unsupervised Anomaly Detection and Root Cause Analysis for an Industrial Press Machine based on Skip-Connected Autoencoder. In *2022 21st IEEE International Conference on*

Machine Learning and Applications (ICMLA), pages 681–686. IEEE, 2022. ISBN 978-1-6654-6283-9. doi: 10.1109/ICMLA55696.2022.00113.

Shubhangi Suryawanshi, Anurag Goswami, and Pramod Patil. IRBM: Incremental Restricted Boltzmann Machines for Concept Drift Detection and Adaption in Evolving Data Streams. In Deepak Garg, Joel J. P. C. Rodrigues, Suneet Kumar Gupta, Xiaochun Cheng, Pushpender Sarao, and Govind Singh Patel, editors, *Advanced Computing*, volume 2053 of *Communications in Computer and Information Science*, pages 466–475. Springer Nature Switzerland, Cham, 2024. ISBN 978-3-031-56699-8. doi: 10.1007/978-3-031-56700-1_37.

Mark Tabladillo. The Team Data Science Process lifecycle, 2024. URL `https://learn.microsoft.com/en-us/azure/architecture/data-science-process/lifecycle`.

Stefan Thomke and Takahiro Fujimoto. The Effect of "Front-Loading" Problem-Solving on Product Development Performance. *Journal of Product Innovation Management: An International Publication of the Product Development & Management Association*, 17(2):128–142, 2000.

Hajer Toumi, Zaki Brahmi, and Mohammed Mohsen Gammoudi. Extended Hoeffding Adaptive Tree based-Server Load Prediction in Cloud Computing environment. In *Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region*, pages 161–168, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450372367. doi: 10.1145/3368474.3368475.

David Tranfield, David Denyer, and Palminder Smart. Towards a Methodology for Developing Evidence–Informed Management Knowledge by Means of Systematic Review. *British Journal of Management*, 14(3):207–222, 2003. ISSN 1045-3172. doi: 10.1111/1467-8551.00375.

Martin Trat and Jivka Ovtcharova. Designing Concept Drift Detection Ensembles: A Survey. In *2023 IEEE 10th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10. IEEE, 2023. ISBN 979-8-3503-4503-2. doi: 10.1109/DSAA60987.2023.10302492.

Martin Trat, Janek Bender, and Jivka Ovtcharova. Sensitivity-Based Optimization of Unsupervised Drift Detection for Categorical Data Streams. *KIT Scientific Working Papers*, 208, 2023. doi: 10.5445/IR/1000155196.

Martin Trat, Philipp Bergmann, Andreas Ott, and Jivka Ovtcharova. A Nature-Inspired Concept Drift Adaptation Method for Industrial Data Stream Regression. In Yi-Chi Wang, Siu Hang Chan, and Zih-Huei Wang, editors, *Flexible Automation and Intelligent Manufacturing: Manufacturing Innovation and Preparedness for the Changing World Order*, Lecture Notes in Mechanical Engineering, pages 3–13. Springer Nature Switzerland, Cham, 2024. ISBN 978-3-031-74481-5. doi: 10.1007/978-3-031-74482-2_1.

Martin Trat, Matthes Elstermann, Jana Deckers, and Jivka Ovtcharova. Modeling a Reference Architecture for Concept Drift Adaptation Systems. In Tung Bui, editor, *Proceedings of the 58th Hawaii International Conference on System Sciences*, Proceedings of the Annual Hawaii International Conference on System Sciences. Hawaii International Conference on System Sciences, 2025. doi: 10.24251/HICSS.2025.121.

Jesper E. van Engelen and Holger H. Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, 2020. ISSN 0885-6125. doi: 10.1007/s10994-019-05855-6.

Daniel Vela, Andrew Sharp, Richard Zhang, Trang Nguyen, Hoang, and Oleg S. Pianykh. Temporal quality degradation in AI models. *Scientific reports*, 12, 2022. doi: 10.1038/s41598-022-15245-z.

Ernst von Glasersfeld. An Exposition of Constructivism: Why Some Like It Radical. *Journal for Research in Mathematics Education*, 4:19–29, 1990. doi: 10.2307/749910.

Julius Voß. *Parameteroptimierung von überwachten Concept-Drift-Detektionsmethoden*. Bachelor's Thesis, Karlsruhe Institute of Technology, Karlsruhe, 2025.

Scott Wares, John Isaacs, and Eyad Elyan. Data stream mining: methods and challenges for handling concept drift. *SN Applied Sciences*, 1(11), 2019. ISSN 2523-3963. doi: 10.1007/s42452-019-1433-0.

Geoffrey I. Webb, Roy Hyde, Hong Cao, Hai Long Nguyen, and Francois Petitjean. Characterizing concept drift. *Data Mining and Knowledge Discovery*, 30(4):964–994, 2016. ISSN 1384-5810. doi: 10.1007/s10618-015-0448-4.

Jane Webster and Richard T. Watson. Analyzing the Past to Prepare for the Future: Writing a Literature Review. *Management Information Systems Research Center*, 26(2):xiii–xxiii, 2002.

Hendro Wicaksono, Martin Trat, Atit Bashyal, Tina Boroukhian, Mine Felder, Mischa Ahrens, Janek Bender, Sebastian Groß, Daniel Steiner, Christoph July, Christoph Dorus, and Thorsten Zoerner. Artificial-intelligence-enabled dynamic demand response system for maximizing the use of renewable electricity in production processes. *The International Journal of Advanced Manufacturing Technology*, 2024. ISSN 0268-3768. doi: 10.1007/s00170-024-13372-7.

Jobin Wilson, Santanu Chaudhury, and Brejesh Lall. Homogeneous-Heterogeneous Hybrid Ensemble for concept-drift adaptation. *Neurocomputing*, 557:126741, 2023. ISSN 09252312. doi: 10.1016/j.neucom.2023.126741.

R. Wirth and Jochen Hipp. CRISP-DM: Towards a standard process model for data mining. *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining*, 2000.

Ian H. Witten, Eibe Frank, and Mark A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier, 2011. ISBN 9780123748560. doi: 10.1016/C2009-0-19715-5.

D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997. ISSN 1089778X. doi: 10.1109/4235.585893.

David H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992. ISSN 08936080. doi: 10.1016/S0893-6080(05)80023-1.

Michał Woźniak. On the performance of applied machine learning models introduced by ensembling: personal communication, 09.10.2023.

Michał Woźniak, Paweł Ksieniewicz, Bogusław Cyganek, and Krzysztof Walkowiak. Ensembles of Heterogeneous Concept Drift Detectors - Experimental Study. In Khalid Saeed and Władysław Homenda, editors, *Computer Information Systems and Industrial Management*, pages 538–549. Springer, Cham, 2016a. ISBN 978-3-319-45378-1.

Michał Woźniak, Paweł Ksieniewicz, Andrzej Kasprzak, Karol Puchała, and Przemysław Ryba. A First Attempt to Construct Effective Concept Drift Detector Ensembles. In Ryszard S. Choraś, editor, *Image Processing and Communications Challenges 8*, volume 525 of *Advances in Intelligent Systems and Computing*, pages 27–34. Springer, Cham, 2016b. ISBN 978-3-319-47273-7. doi: 10.1007/978-3-319-47274-4_3.

Qiuyan Xiang, Lingling Zi, Xin Cong, and Yan Wang. Concept Drift Adaptation Methods under the Deep Learning Framework: A Literature Review. *Applied Sciences*, 13(11):6515, 2023. doi: 10.3390/app13116515.

Xun Xu, Yuqian Lu, Birgit Vogel-Heuser, and Lihui Wang. Industry 4.0 and Industry 5.0—Inception, conception and perception. *Journal of Manufacturing Systems*, 61:530–535, 2021. ISSN 02786125. doi: 10.1016/j.jmsy.2021.10.006.

Yiming Xu and Diego Klabjan. Concept Drift and Covariate Shift Detection Ensemble with Lagged Labels. In Yixin Chen, editor, *2021 IEEE International Conference on Big Data*, pages 1504–1513, Piscataway, NJ, USA, 2021. IEEE. ISBN 978-1-6654-3902-2. doi: 10.1109/BigData52589.2021.9671279.

Shuxiang Zhang, David Tse Jung Huang, Gillian Dobbie, and Yun Sing Koh. SLED: Semi-supervised Locally-weighted Ensemble Detector. In *2020 IEEE 36th International Conference on Data Engineering*, pages 1838–1841, Piscataway, NJ, 2020. IEEE. ISBN 978-1-7281-2903-7. doi: 10.1109/ICDE48307.2020.00183.

Indrė Žliobaitė, Albert Bifet, Mohamed Gaber, Bogdan Gabrys, João Gama, Leandro Minku, and Katarzyna Musial. Next challenges for adaptive learning systems. *ACM SIGKDD Explorations Newsletter*, 14(1):48–55, 2012. ISSN 1931-0145. doi: 10.1145/2408736.2408746.

Indrė Žliobaitė, Albert Bifet, Bernhard Pfahringer, and Geoffrey Holmes. Active learning with drifting streaming data. *IEEE transactions on neural networks and learning systems*, 25(1): 27–39, 2014. doi: 10.1109/TNNLS.2012.2236570.

Indrė Žliobaitė, Mykola Pechenizkiy, and João Gama. An Overview of Concept Drift Applications. In Nathalie Japkowicz and Jerzy Stefanowski, editors, *Big Data Analysis: New Algorithms for a New Society*, volume 16 of *Studies in Big Data*, pages 91–114. Springer International Publishing, Cham, 2016. ISBN 978-3-319-26987-0. doi: 10.1007/978-3-319-26989-4_4.