HAW HAMBURG

MASTERTHESIS

Nhut-Hoa Huynh

Übersicht und Vergleich von Algorithmen zur Erkennung von Drift

FAKULTÄT TECHNIK UND INFORMATIK
Department Informatik

Faculty of Computer Science and Engineering
Department Computer Science

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN HAMBURG
Hamburg University of Applied Sciences

Nhut-Hoa Huynh

Übersicht und Vergleich von Algorithmen zur Erkennung von Drift

Masterarbeit eingereicht im Rahmen der Masterprüfung
im Studiengang *Master of Science Informatik*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Marina Tropmann-Frick
Zweitgutachter: Prof. Dr. Stefan Sarstedt

Eingereicht am: 11. August 2025

Nhut-Hoa Huynh

Thema der Arbeit

Übersicht und Vergleich von Algorithmen zur Erkennung von Drift

Stichworte

Modelldrift, Drifterkennung, Vergleich von Algorithmen, Literaturrecherche, experimentelle Evaluation

Kurzzusammenfassung

Diese Arbeit untersucht verschiedene Ansätze zur Modelldrift-Erkennung, darunter daten-verteilungsbasierte, leistungsbasierte, mehrfachhypothesenbasierte und kontextbasierte Methoden. Die Algorithmen werden sowohl theoretisch erläutert als auch in Experimenten hinsichtlich ihrer Genauigkeit und Effizienz miteinander verglichen. Die Ergebnisse verdeutlichen, dass es keine universelle Lösung gibt; die Wahl der Methode ist vielmehr abhängig von den Driftarten, Driftgrößen und den spezifischen Anwendungsanforderungen. Die Arbeit unterstreicht daher die Notwendigkeit, die einzelnen Ansätze an die unterschiedlichen Szenarien anzupassen, um sowohl die Erkennungsgenauigkeit zu maximieren als auch die Verarbeitungseffizienz zu optimieren.

Nhut-Hoa Huynh

Title of Thesis

Overview and Comparison of Algorithms for Drift Detection

Keywords

Concept Drift, Drift Detection, Algorithm Comparison, Literature Review, Experimental Evaluation

Abstract

This thesis investigates different approaches to model drift detection, including data-distribution-based, performance-based, multiple-hypothesis-based and context-based methods. The algorithms are both explained theoretically and compared in experiments with respect to their accuracy and efficiency. The results show that there is no universal solution; the choice of method depends on the drift types, drift sizes and the specific application requirements. The work therefore underlines the need to adapt the individual approaches to the different scenarios in order to maximise both detection accuracy and processing efficiency.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Formelverzeichnis

1 Einleitung

1.1 Ziele dieser Arbeit

Die vorliegende Masterarbeit hat das Ziel, Vergleiche zwischen vier verschiedenen Ansätzen zur Erkennung von Modell-Drift zu ziehen: den datenverteilungsbasierten, leistungsbasierten, mehrfachhypothesenbasierten sowie kontextualisierten Methoden. Die Arbeit untersucht, welcher dieser Ansätze in unterschiedlichen Anwendungen am effektivsten sein könnte, indem sie ihre Anwendbarkeit in verschiedenen Szenarien bewertet und Vergleiche hinsichtlich ihrer Effizienz und Genauigkeit anstellt. Das Ziel ist es, eine fundierte Entscheidungshilfe für die Wahl der am besten geeignete Methode für spezifische Szenarien zu liefern, sodass eine praxisorientierte Grundlage für die Auswahl der optimalen Erkennungsstrategie geschaffen wird.

1.2 Gliederung dieser Arbeit

Die Gliederung dieser Masterarbeit umfasst mehrere Kapitel.

Kapitel 2 stellt die Grundlagen vor, beginnend mit der Definition und den Auswirkungen von Modelldriften, gefolgt von einer Untersuchung deren Arten und Ursachen. Zudem werden die Methoden zur Erkennung von Modelldrift vorgestellt.

In Kapitel 3 werden datenverteilungsbasierte Ansätze behandelt, wobei sowohl univariate als auch multivariate Daten analysiert werden. Es werden verschiedene Maße der Verteilungsdissimilarität und Distanzmaße zwischen Beobachtungen vorgestellt, gefolgt von einem Fazit dieser Ansätze.

Kapitel 4 widmet sich leistungsbasierten Ansätzen wie statistische Prozesskontrolle, Fenstertechniken und Ensemble-Lernen, und schließt mit einem Fazit zu diesen Methoden.

In Kapitel 5 werden mehrfache Hypothesen-basierte Ansätze untersucht, mit einem Fokus auf parallele und hierarchische Mehrfachhypothesen-Driftdetektoren. Auch hier wird ein abschließendes Fazit zu diesen Ansätzen gezogen.

Kapitel 6 befasst sich mit kontextbasierten Ansätzen, bevor in Kapitel 7 ein Experiment zu dem Vergleich von Algorithmen durchführt.

Kapitel 8 fasst die Ergebnisse zusammen und gibt einen Ausblick auf zukünftige Forschung.

1.3 State of the Art – Verwandte Forschungsarbeiten

Im Rahmen dieser Masterarbeit wird der aktuelle Forschungsstand zu Modelldriften und deren Erkennung untersucht. In der Literatur wurden zahlreiche Ansätze und Methoden beschrieben, die sich mit der Analyse von Modelldriften befassen. Zu den relevantesten Arbeiten zählen insbesondere die in Abbildung 1 dargestellten Studien. Diese Ansätze werden kritisch miteinander verglichen, um ein tiefgehendes Verständnis der Methoden zu erlangen und deren Eignung für spezifische Anwendungsfälle bewerten zu können.

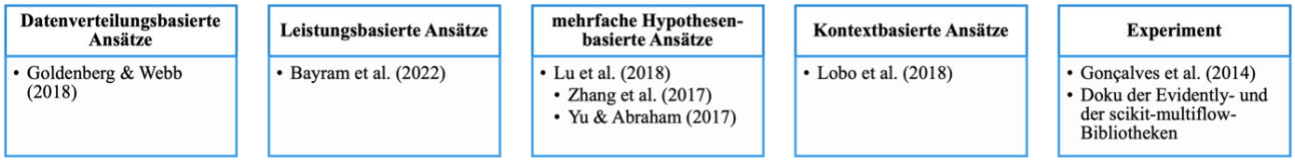


Abbildung 1: Sieben relevantesten Forschungsarbeiten

Quelle: Eigene Darstellung

Diese Arbeit orientiert sich an sieben zentralen Arbeiten, die verschiedene Ansätze und Perspektiven abdecken. Goldenberg und Webb (2018) legen mit ihrem datenverteilungsbasierten Ansatz eine wesentliche Grundlage für die Analyse von Datenveränderungen. Bayram et al. (2022) fokussieren sich auf leistungsbasierte Ansätze. Lu et al. (2018) bilden die Grundlage für mehrfache hypothesenbasierte Ansätze, ergänzt durch die Arbeiten von Zhang et al. (2017) und Yu & Abraham (2017), die als Beispiele für unterschiedliche Anwendungen dienen. Lobo et al. (2018) widmen sich kontextbasierten Ansätzen. Das eigene Experiment stützt sich ebenfalls auf die Arbeiten von Gonçalves et al. (2014) sowie die Dokumentation der Evidently- und scikit-multiflow-Bibliotheken und liefert eine praktische Grundlage zur Validierung der theoretischen Konzepte.

Diese Masterarbeit baut auf relevanten Theorien, früheren Forschungsergebnissen und Daten auf. Die genannten Studien bieten eine fundierte Basis und ermöglichen eine originelle Argumentation, die zur wissenschaftlichen Diskussion beiträgt.

2 Grundlagen

2.1 Definition und Auswirkungen von Modelldriften

Modelldrift, auch als „Model Decay“ bezeichnet, beschreibt das Phänomen, bei dem die Leistung eines maschinellen Lernmodells im Laufe der Zeit nachlässt. Dies führt dazu, dass die Vorhersagen eines betroffenen Modells zunehmend ungenauer oder fehlerhafter werden.

Die Auswirkungen solcher ungenauen Vorhersagen können gravierend sein. Unternehmen, die auf die Prognosen ihrer Modelle angewiesen sind, laufen Gefahr, fehlerhafte Planungen vorzunehmen oder ineffiziente Kampagnen zu entwickeln. In der Produktion können fehlerhafte Modelle dazu führen, dass Geschäfts- oder Produktionsprozesse scheitern oder die Kosten steigen. Aus diesem Grund ist es von entscheidender Bedeutung, Modelldrift frühzeitig zu erkennen und zu beheben, um eine kontinuierlich hohe Modellleistung und -genauigkeit zu gewährleisten.

2.2 Arten und Ursachen von Modelldriften

2.2.1 Arten von Modelldriften

Mathematisch gesehen tritt Modelldrift auf, wenn sich die gemeinsame Wahrscheinlichkeitsverteilung $p(X, Y)$ der Merkmale X und der Zielvariablen Y im Vergleich zur Trainingsphase ändert. Die verschiedenen Driftarten lassen sich wie folgt charakterisieren:

1. Kovariatenverschiebung ($p(X) \neq p'(X),\ p(Y|X) = p'(Y|X)$):

Eine Kovariatenverschiebung liegt vor, wenn sich die Verteilung der Eingabemerkmale ändert, die Beziehung zwischen den Merkmalen und der Zielvariablen jedoch gleichbleibt.

Ein Beispiel dafür ist ein Online-Shop, in dem Kaufempfehlungen auf Basis des Nutzerverhaltens erstellt werden. Wenn eine neue Zielgruppe mit anderen Interessen hinzukommt, ändern sich die Eingabedaten (z. B. Suchanfragen). Die zugrunde liegende Logik der Kaufempfehlungen bleibt jedoch unverändert.

2. Verschiebung der A-priori-Wahrscheinlichkeit ($p(Y) \neq p'(Y),\ p(X|Y) = p'(X|Y)$):

Diese Form der Drift tritt auf, wenn sich die Verteilung der Zielvariablen verändert, während die Beziehung zwischen den Merkmalen und der Zielvariablen konstant bleibt.

Ein praktisches Beispiel ist eine Wettervorhersage, bei der die Wahrscheinlichkeiten für verschiedene Wetterlagen saisonal schwanken. Im Sommer steigt beispielsweise die Wahrscheinlichkeit für

Sonnenschein, während im Winter Schneefall häufiger ist. Die Beziehung zwischen Faktoren wie Temperatur oder Luftdruck und den Wetterlagen bleibt dabei gleich.

3. Konzeptdrift ($p(Y|X) \neq p'(Y|X)$):

Ein Konzeptdrift tritt auf, wenn sich die Beziehung zwischen den Merkmalen und der Zielvariablen ändert. Dies führt dazu, dass die ursprünglichen Annahmen des Modells nicht mehr zutreffen.

Ein typisches Beispiel ist ein Spam-Filter, der auf bestimmten Schlüsselwörtern basiert. Wenn Spammer neue Strategien entwickeln und andere Begriffe verwenden, verliert das Modell an Genauigkeit, da die bisherigen Merkmale nicht mehr ausreichen, um Spam zuverlässig zu erkennen.

Die Veränderungen, die im Zusammenhang mit Modelldrift auftreten, können in verschiedene Muster unterteilt werden, abhängig von der Art und Weise, wie diese Drift im System verläuft. Dabei werden vier Haupttypen unterschieden, die in Abbildung 2 dargestellt sind:

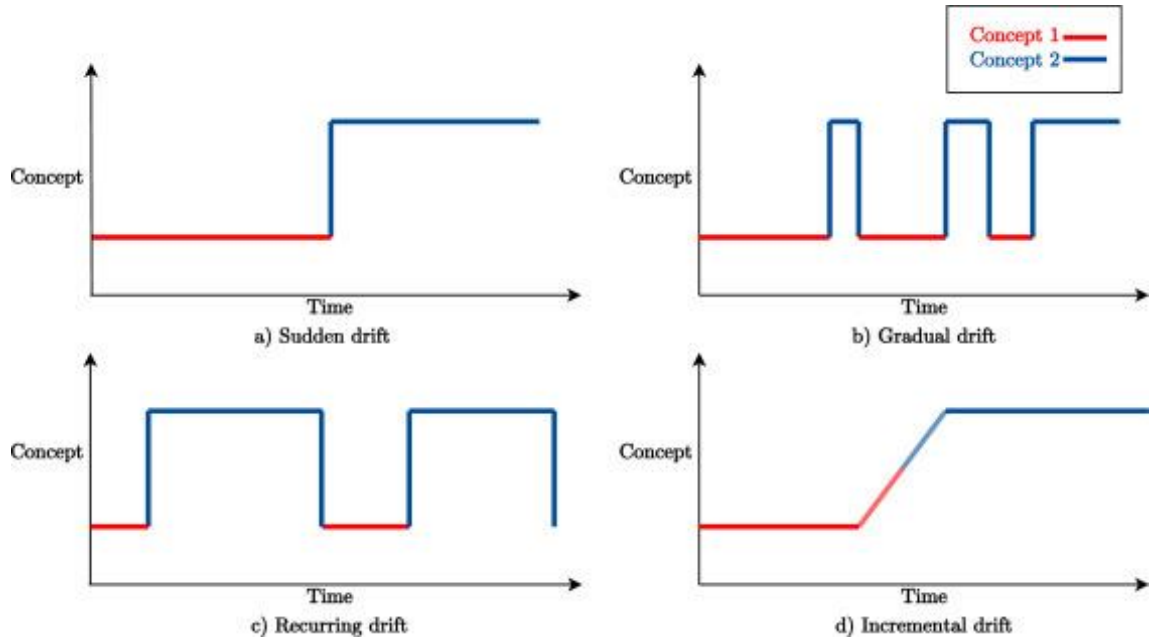


Abbildung 2: Vier Kategorien der Modelldrift basierend auf dem Muster des Auftretens

Quelle: Bayram et al. (2022)

1. Plötzliche Drift:

Plötzliche Drift tritt auf, wenn die Zielverteilung abrupt von einem Konzept auf ein anderes wechselt. Diese Form der Drift ist durch eine schnelle und unerwartete Veränderung gekennzeichnet, die oft schwer vorhersehbar ist.

Ein Beispiel hierfür wäre ein Streaming-Dienst, der seine Benutzeroberfläche radikal ändert, wodurch sich die Klickmuster der Nutzer plötzlich komplett ändern.

2. Graduelle Drift:

Graduelle Drift beschreibt eine schrittweise Veränderung der Zielverteilung, bei der ein bestehendes Konzept allmählich durch ein neues ersetzt wird. Im Gegensatz zur plötzlichen Drift erfolgt diese Veränderung in kleinen, kontinuierlichen Schritten.

Dies könnte beispielsweise bei einem Online-Shop geschehen, in dem die Verkaufszahlen eines neuen Produkts langsam zunehmen und die Präferenzen der Kunden nach und nach beeinflussen.

3. Wiederkehrende Drift:

Wiederkehrende Drift tritt auf, wenn ein zuvor beobachtetes Konzept nach einer gewissen Zeit wiederkehrt. Im Gegensatz zur graduellen Drift, bei der das alte Konzept langsam ausläuft, erscheint das frühere Konzept bei der wiederkehrenden Drift nach einer gewissen Zeitspanne erneut, was oft in zyklischen Mustern geschieht.

Ein typisches Szenario wäre ein Supermarkt, in dem der Absatz von saisonalen Produkten wie Eiscreme oder Weihnachtsgebäck jedes Jahr ähnliche Muster zeigt.

4. Inkrementelle Drift:

Inkrementelle Drift bezeichnet einen kontinuierlichen, langsamen Übergang, bei dem ein neues Konzept das alte ersetzt, ohne dass eine klare Trennung zwischen den beiden Konzepten erkennbar ist.

So ändern sich beispielsweise Nutzerpräferenzen auf Social-Media-Plattformen allmählich durchschleichende Trends, ohne dass es zu abrupten Veränderungen kommt.

2.2.2 Ursachen von Modelldriften

Es gibt zahlreiche mögliche Ursachen für Modelldrift. Veränderungen in den Datenverteilungen, wie etwa bei der Kovariatenverschiebung und der Verschiebung der A-priori-Wahrscheinlichkeit, können durch verschiedene Faktoren ausgelöst werden. Ein Beispiel sind saisonale Schwankungen, wie sie beispielsweise während der Weihnachtszeit auftreten, wenn die Verkaufszahlen steigen. Auch Änderungen im menschlichen Verhalten, wie etwa im Konsumverhalten oder Kaufverhalten, können zu einer Modelldrift führen.

Darüber hinaus können auch technologische Fortschritte eine Rolle spielen. Die Einführung neuer Datenquellen oder Datenformate verändert die Struktur der Daten und kann so die Grundlage eines Modells beeinflussen, was wiederum zu einer Modelldrift führt.

Technologische Neuerungen, wie die Einführung neuer Geräte oder Betriebssysteme, können eine Kontextdrift hervorrufen, da sie die Art und Weise verändern, wie Daten erfasst oder verarbeitet werden. Ebenso können strategische Veränderungen, wie etwa eine Neuausrichtung auf eine andere Zielgruppe, den Kontext beeinflussen und damit ebenfalls zu einer Kontextdrift führen.

Zudem können Anpassungen der Anforderungen an Produkte oder Dienstleistungen die Rahmenbedingungen für die Nutzung des Modells verändern und so eine Kontextdrift auslösen. Diese Veränderungen betreffen nicht nur die Daten selbst, sondern auch die Bedingungen, unter denen das Modell angewendet wird.

2.3 Methoden zur Erkennung von Modelldriften

„Die Erkennung von Drift, auch als Änderungsdetektion bekannt, bezieht sich auf die Methode, mit der der genaue Zeitpunkt oder Zeitraum ermittelt wird, in dem sich die Eigenschaften des Zielobjekts verändern“ (Basseville, M., 1993). Zur Erkennung von Modelldrift werden häufig statistische Tests eingesetzt, um den Datenstrom zu überwachen und die Ähnlichkeit zwischen alten und neuen Datenproben zu quantifizieren. Auf diese Weise können Veränderungen erkannt werden.

„Bestehende Studien zur Modelldrift -Erkennung lassen sich anhand der verwendeten Teststatistiken, die zur Identifikation und Lokalisierung der Veränderung dienen, in verschiedene Kategorien einteilen“ (Bayram et al., 2022), wie in Abbildung 3 dargestellt:

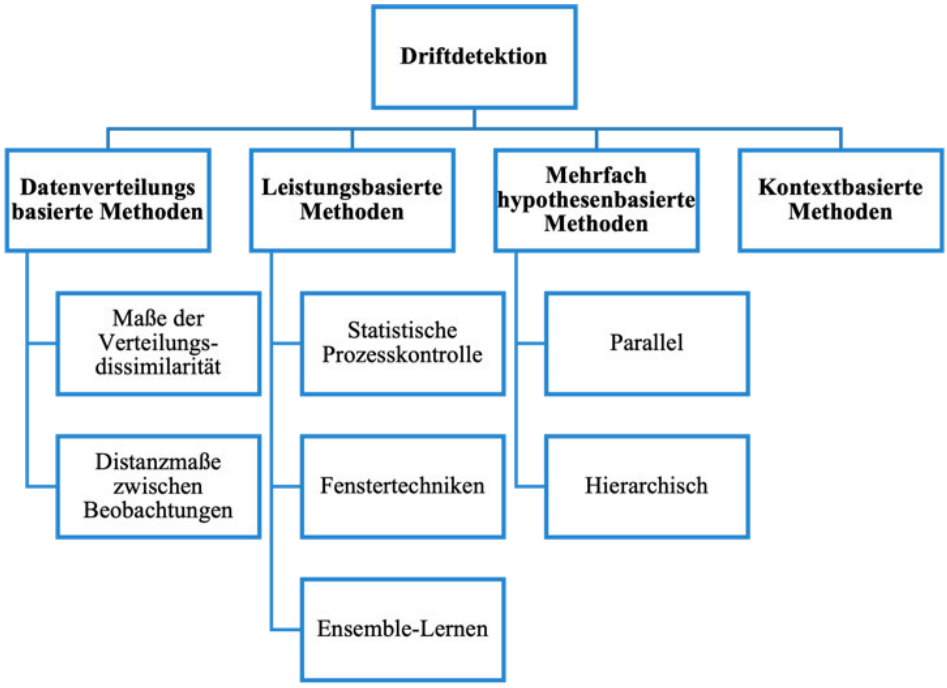


Abbildung 3: Hierarchie der Drift- Detektion

Quelle: Eigene Darstellung

Die erste Kategorie zur Erkennung von Modelldrift konzentriert sich auf die Überwachung von Dateneigenschaften. Dabei wird festgestellt, ob sich Verteilungen oder deskriptive Statistiken wie Minimum, Maximum, Median und Mittelwert signifikant verändert haben. Diese Methode basiert darauf, Veränderungen in den grundlegenden Eigenschaften der Daten zu erkennen, was auf eine mögliche Drift im Modell hinweisen könnte.

Eine weitere Methode besteht darin, Veränderungen in den Leistungsmetriken des Modells zu überwachen. Hierbei können Vorhersagefehler sowie Kennzahlen wie Konfusionsmatrix, Genauigkeit, Rückruf und F1-Score zur Identifikation von Drift verwendet werden. Diese Leistungsmetriken geben Aufschluss darüber, wie gut das Modell mit den aktuellen Daten arbeitet und ob es an Genauigkeit verliert, was auf eine Modelldrift hindeuten könnte.

Obwohl datenverteilungs- und leistungsbasierte Ansätze zu den am häufigsten eingesetzten Techniken zur Erkennung von Modelldrift gehören, existieren auch komplexere Ansätze wie multiple Hypothesen-basierte und kontextbasierte Methoden. Diese Ansätze erweitern die Möglichkeiten zur Drift-Erkennung, indem sie zusätzliche Informationen oder Modelle einbeziehen, die helfen, komplexe Veränderungen in den Daten und im Modell zu identifizieren.

Nachfolgend werden die vier Methodengruppen näher erläutert, um deren spezifische Merkmale, Vorteile und Einsatzmöglichkeiten zu untersuchen.

3 Datenverteilungsbasierte Ansätze

Datenverteilungsbasierte Ansätze konzentrieren sich darauf, Veränderungen in der zugrundeliegenden Datenverteilung eines Modells zu identifizieren. Diese Methoden basieren auf der Annahme, dass Modellleistungen stark von der Datenverteilung abhängen, die während des Trainings vorherrschte. Typischerweise werden statistische Tests oder Distanzmaße verwendet, um Unterschiede zwischen der ursprünglichen und der aktuellen Datenverteilung zu quantifizieren. Eine Modelldrift wird dann erkannt, wenn die beiden Verteilungen deutlich voneinander entfernt sind.

Goldenberg und Webb (2018) unterteilen die datenverteilungsbasierten Methoden in zwei Hauptgruppen: Maße der Verteilungsdissimilarität und Distanzmaße zwischen Beobachtungen. Diese beiden Kategorien werden sowohl für univariate als auch multivariate Daten eingesetzt.

1. Maße der Verteilungsdissimilarität vergleichen globale Eigenschaften von Verteilungen, wie Form, Lage und Streuung, und quantifizieren die Unterschiede zwischen zwei Wahrscheinlichkeitsverteilungen. Typische Beispiele sind der Hellinger-Abstand, die Kullback-Leibler-Divergenz, die Kolmogorov-Smirnov-Statistik oder die Wasserstein-Distanz, die insbesondere für die Erkennung von Kovariatendrift verwendet werden.
2. Distanzmaße zwischen Beobachtungen hingegen konzentrieren sich auf die lokalen Unterschiede einzelner Datenpunkte oder Mittelwerte. Der T-Ratio (univariat) und Hotelling's

T^2 (multivariat) messen beispielsweise die Abstände zwischen Mittelwerten von Gruppen, um spezifische Verschiebungen zu identifizieren.

Die Unterscheidung zwischen diesen beiden Ansätzen ist essenziell, da globale und lokale Drifts unterschiedliche Herausforderungen darstellen. Während Maße der Verteilungsdissimilarität ein Gesamtbild der Veränderung liefern, sind Distanzmaße zwischen Beobachtungen effektiver bei der Identifikation punktueller oder segmentbezogener Veränderungen.

Diese Methoden werden in den Abschnitten 3.1 und 3.2 genauer beschrieben.

3.1 Univariate Daten

3.1.1 Maße der Verteilungsdissimilarität für univariate Daten

Hellinger-Abstand

In dieser Gruppe sticht der Hellinger-Abstand als eine der bekanntesten und am häufigsten verwendeten Methoden hervor. Der Hellinger-Abstand misst die Ähnlichkeit zwischen zwei Wahrscheinlichkeitsverteilungen P_1 und P_2. Für diskrete Verteilungen wird er durch die Formel definiert:

$$D_H(P_1, P_2) = \frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{k}\left(\sqrt{p_i} - \sqrt{q_i}\right)^2} \quad (1)$$

Formel 1: Hellinger-Abstand

Quelle: Goldenberg & Webb (2018)

wobei p_i und q_i die Wahrscheinlichkeiten der i-ten Kategorie in den Verteilungen P_1 und P_2 sind. Der Wert liegt zwischen 0 (identische Verteilungen) und 1 (vollständig unterschiedliche Verteilungen).

Der Hellinger-Abstand wird oft verwendet, um Veränderungen in eindimensionalen Datensätzen zu quantifizieren, insbesondere bei einfachen Drifts, wie etwa bei Klassifikationsproblemen mit wenigen Merkmalen.

Kullback-Leibler-Divergenz

Die Kullback-Leibler-Divergenz (KL-Divergenz) gehört ebenfalls zu den bekanntesten Methoden zur Messung der Differenz zwischen zwei Wahrscheinlichkeitsverteilungen. Sie quantifiziert die Informationsdifferenz zwischen einer Verteilung P und einer anderen Verteilung Q und bewertet, wie viel Information verloren geht, wenn die Verteilung P durch die Verteilung Q approximiert wird. Sie ist besonders nützlich für Anwendungen, bei denen es wichtig ist, den Informationsverlust bei Verteilungsmodellierungen zu messen. Die Formel für die KL-Divergenz lautet:

$$D_{KL}\ (P\ ||Q) = \sum_i p_i \log\left(\frac{p_i}{q_i}\right) \quad (2)$$

Formel 2: Kullback-Leibler-Divergenz

Quelle: Goldenberg & Webb (2018)

wobei p_i und q_i auch die Wahrscheinlichkeiten der i-ten Kategorie in den Verteilungen P und Q sind. Der Wert der KL-Divergenz ist immer größer oder gleich Null, und sie wird nur null, wenn P und Q identisch sind. *„Ein wichtiger Aspekt der KL-Divergenz ist, dass sie nicht symmetrisch ist"* (Goldenberg & Webb, 2018), was bedeutet, dass $D_{KL}(P\ ||Q)$ nicht dasselbe ist wie $D_{KL}(Q\ ||P)$. Diese Asymmetrie impliziert, dass das Ergebnis davon abhängt, welche Verteilung als „ursprünglich" und welche als „neu" betrachtet wird.

Kolmogorov-Smirnov-Statistik

Eine weitere bekannte Methode, um die Differenz oder Ähnlichkeit zwischen Verteilungen zu messen, ist die Kolmogorov-Smirnov-Statistik. Sie misst die maximale absolute Differenz zwischen den empirischen Verteilungsfunktionen zweier Stichproben oder einer Stichprobe und einer theoretischen Verteilung und wird definiert als:

$$D_{KS} = \sup_x |F_1(x) - F_2(x)| \quad (3)$$

Formel 3: Kolmogorov-Smirnov-Statistik

Quelle: Goldenberg & Webb (2018)

Das „sup" bezeichnet das „Supremum", also den größten Wert der Differenz zwischen den Verteilungsfunktionen ($|F_1(x) - F_2(x)|$) über alle möglichen Werte x.

Die Kolmogorov-Smirnov-Statistik eignet sich gut als Driftdetektor, da sie nicht parametrisch ist, ohne Annahmen über die Verteilungsform angewendet werden kann und besonders nützlich ist, um plötzliche oder allmähliche Verschiebungen in univariaten Daten zu erkennen.

Wasserstein-Distanz

Eine weitere Methode ist die Wasserstein-Distanz, auch als „Earth-Mover's Distance" (EMD) bekannt. Sie misst die minimalen Kosten, um eine Verteilung P in eine andere Q zu transformieren. Sie basiert auf der Lösung des sogenannten „Transportproblems" (Rubner et al., 2000), bei dem Cluster innerhalb der Verteilungen durch ihre Mittelwerte (p_i, q_j) und Gewichte $\left(w_{p_i}, w_{q_j}\right)$ repräsentiert werden.

Die EMD wird durch die folgende Formel beschrieben:

$$EMD(P,Q) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij} f_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}} \quad (4)$$

Formel 4: Wasserstein-Distanz

Quelle: Goldenberg & Webb (2018)

Dabei sind d_{ij} die Distanz zwischen den Clustern und f_{ij} der optimierte Massefluss ist.

Die Wasserstein-Distanz ist nützlich, um Veränderungen in der Verteilung von Daten zu erkennen, da sie Form, Position und Streuung berücksichtigt. Sie eignet sich für numerische Daten und mit einer angepassten Distanzmetrik auch für kategoriale Daten. *„Ein Nachteil ist jedoch, dass ihre Berechnung durch die Lösung eines linearen Optimierungsproblems aufwendig sein kann, insbesondere bei großen Datenmengen“* (Goldenberg & Webb, 2018).

3.1.2 Distanzmaße zwischen Beobachtungen für univariate Daten

Innerhalb dieser Gruppe spielt der T-Ratio und seine Grundlage, der T-Test, eine entscheidende Rolle, da sie Unterschiede in den Mittelwerten zwischen zwei Gruppen aufzeigen können.

T-Statistik & T-Test

Der T-Test, der die Basis des T-Ratio bildet, wird eingesetzt, um zu prüfen, ob der Mittelwert X einer Stichprobe signifikant von einem vorgegebenen Wert (μ_0) abweicht. Der Test berechnet den „T-Statistik“, der angibt, wie weit der beobachtete Mittelwert X in Einheiten des Standardfehlers von μ_0 entfernt ist. Die Formel lautet:

$$T = \frac{X - \mu_0}{S/\sqrt{n}} \quad (5)$$

Formel 5: T-Statistik

Quelle: Goldenberg & Webb (2018)

wobei S die Standardabweichung der Stichprobe und n die Stichprobengröße. Ist der T-Statistik groß genug (über einem vordefinierten kritischen Wert), gibt es Hinweise darauf, dass der Stichprobenmittelwert signifikant von μ_0 abweicht. Es werden drei Arten von T-Tests unterschieden:

1. T-Test für unabhängige Stichproben: Dieser Test wird eingesetzt, um die Mittelwerte von zwei unabhängigen Gruppen zu vergleichen.
2. T-Test für abhängige, gepaarte Stichproben: Dieser Test wird eingesetzt, um die Mittelwerte der gleichen Gruppe zu verschiedenen Zeitpunkten zu vergleichen.
3. T-Test für eine Stichprobe: Dieser Test wird eingesetzt, um den Stichprobenmittelwert mit einem bekannten Wert zu vergleichen.

T-Ratio

Für die Drifterkennung ist der T-Test für abhängige, gepaarte Stichproben, auch T-Ratio genannt, relevant. Er basiert auf dem T-Test und bewertet die Differenz zwischen den Mittelwerten zweier Gruppen in Einheiten der Standardabweichung. Die Formel lautet:

$$T = \frac{\overline{X_1} - \overline{X_2}}{S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \tag{6}$$

Formel 6: T-Ratio

Quelle: Goldenberg & Webb (2018)

wobei $\overline{X_1}$ und $\overline{X_2}$ die Mittelwerte der beiden Gruppen sind, n_1 und n_2 ihre Stichprobengrößen und S_p die gepoolte Standardabweichung, berechnet als

$$S_p = \sqrt{\frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2}} \tag{7}$$

Formel 7: Standardabweichung des T-Ratios

Quelle: Goldenberg & Webb (2018)

wobei S_1und S_2 die Standardabweichungen der jeweiligen Gruppen sind.

Der T-Ratio ist besonders relevant für Drift-Szenarien, in denen die Mittelwerte der Daten verschoben werden, wie beispielsweise bei Änderungen der durchschnittlichen Kaufhäufigkeit eines Produkts über verschiedene Zeiträume.

3.2 Multivariate Daten

3.2.1 Maße der Verteilungsdissimilarität für multivariate Daten

Die Analyse von Drifts in multivariaten Daten stellt eine besondere Herausforderung dar, da Veränderungen in den Daten über mehrere Dimensionen hinweg erkannt werden müssen. Zwei wichtige Ansätze zur Erkennung solcher Drifts sind die auf der Principal Component Analysis (PCA) basierende Drifterkennung und die Mahalanobis-Distanzverteilung.

Principal Component Analysis (PCA) basierende Drifterkennung

Die PCA analysiert hochdimensionale Daten durch Berechnung der Hauptkomponenten. Bei der Drifterkennung werden die Daten einer Referenzperiode und einer neuen Periode auf dieselben Hauptkomponenten projiziert. Die Differenz in den Hauptkomponenten wird mit der Kullback-Leibler-Divergenz gemessen, und der höchste Differenzwert weist auf mögliche Drifts hin. Goldenberg & Webb, 2018, argumentieren jedoch, dass *„für viele Zwecke entweder die Leibler-Divergenz oder die Gesamtvariationsdistanz besser geeignet ist“*.

Mahalanobis-Distanzverteilung

Ein weiterer Ansatz ist die Mahalanobis-Distanzverteilung, die den Abstand zwischen einem Punkt und einer Gruppe von Datenpunkten unter Berücksichtigung der Form und Streuung der Daten misst. Die Formel für die Mahalanobis-Distanz (D) ist:

$$D = \sqrt{(x - \mu)^\top S^{-1}(x - \mu)} \tag{8}$$

Formel 8: Mahalanobis-Distanz

Quelle: Goldenberg & Webb (2018)

In der Formel ist x der zu bewertende Punkt, μ der Mittelwert der Datenpunkte und S die Kovarianzmatrix, die die Streuung der Daten angibt, wobei die Inverse (S^{-1}) den Einfluss der Streuung auf den Abstand berücksichtigt.

Laut Goldenberg & Webb, 2018 ermöglicht der Ansatz der Mahalanobis-Distanz die Prüfung einer vermuteten Verteilung, jedoch nicht die Dissimilarität zwischen Proben, wenn die Verteilung weder bekannt ist noch vorausgesetzt wird. Da *„die Mahalanobis-Distanz kein direktes Maß für Drift ist“* (Goldenberg & Webb, 2018), kann sie dennoch zur Erkennung von Drift verwendet werden, indem geprüft wird, ob neue Datenpunkte noch in die ursprüngliche Verteilung passen. Ein Vorteil ihrer Nutzung ist, dass sie das Problem hoher Dimensionen mindern kann.

3.2.2 Distanzmaße zwischen Beobachtungen für multivariate Daten

Ein bedeutendes Distanzmaß in dieser Kategorie ist der Hotelling's T²-Abstand. Dieser stellt *„eine multivariate Erweiterung des T-Ratio“* dar (Goldenberg & Webb, 2018) und misst die Abweichung eines Punktes oder einer Gruppe von der erwarteten multivariaten Verteilung. Für zwei Gruppen mit Mittelwerten $\bar{X}$ und $\bar{Y}$, sowie einer gemeinsamen Kovarianzmatrix S, wird der Abstand definiert als:

$$T^2 = (\bar{X} - \bar{Y})' S^{-1} (\bar{X} - \bar{Y}) \tag{9}$$

Formel 9: Hotelling's T²

Quelle: Goldenberg & Webb (2018)

Dabei wird S wie folgt berechnet wird:

$$S = \frac{(n_1 - 1)S_1 + (n_2 - 1)S_2}{n_1 + n_2 - 2} \tag{10}$$

Formel 10: Kovarianzmatrix des Hotelling's T²

Quelle: Goldenberg & Webb (2018)

wobei S_1 und S_2 die Kovarianzmatrizen der beiden Stichproben sowie n_1 und n_2 die Stichprobengrößen sind.

Nach Goldenberg & Webb, 2018 bietet Hotelling's T^2 mehrere Vorteile, darunter die Fähigkeit, als dimensionslose Maßeinheit für Verschiebungen in den Daten verwendet zu werden. Zudem ist es robust gegenüber Normalitätsannahmen für große Stichproben und funktioniert auch bei hochdimensionalen Daten. Jedoch gibt es auch einige Nachteile: Hotelling's T^2 kann keine Veränderungen in der Verteilung erfassen, und Probleme wie Multikollinearität oder eine singuläre Kovarianzmatrix können die Berechnung erschweren.

3.3 Fazit der datenverteilungsbasierten Ansätze

Zusammenfassend lässt sich sagen, dass Maße wie die Hellinger-Distanz, die Kullback-Leibler-Divergenz und die Kolmogorov-Smirnov-Distanz für univariate Daten beliebiger Größe numerisch approximiert werden können, aber für höhere Dimensionen nicht gut skalieren.

Die Wasserstein-Distanz ist für große Datenmengen nicht skalierbar, was ihre Anwendbarkeit einschränkt. Sie ist jedoch besonders nützlich, wenn die Stichproben in eine kleine Anzahl von Clustern unterteilt werden können.

T-Statistiken und Hotelling's T^2 bieten eine gute dimensionslose Annäherung an den Abstand zwischen den Mittelwerten univariater und multivariater Daten.

Der PCA-basierte Ansatz hilft bei der Messung der Drift in hochdimensionalen Daten, erfordert jedoch weitere Forschung.

Im Allgemein bieten datenverteilungsbasierte Algorithmen den Vorteil, dass *„sie nicht nur den Zeitpunkt einer Verteilungsdrift präzise identifizieren, sondern auch den Ort der Drift bestimmen können"* (Lu et al., 2018). Ein weiterer Pluspunkt ist ihre Vielseitigkeit, da *„sie sowohl auf gelabelte als auch auf unlabelte Datensätze angewendet werden können, da sie lediglich die Verteilung der Datenpunkte berücksichtigen"* (Bayram et al., 2022). Allerdings weisen Bayram et al. (2022) auch darauf hin, dass sich Änderungen in der Datenverteilung nicht immer auf die Modellleistung auswirken, was in einigen Fällen zu Fehlalarmen führen kann. Aus diesem Grund werden im nächsten Kapitel die leitungsbasierten Ansätze erläutert.

4 Leistungsbasierte Ansätze

Leistungsbasierte Ansätze zur Drifterkennung zielen darauf ab, Veränderungen durch die Überwachung und Analyse der Modellleistung im Zeitverlauf zu identifizieren. Ein Rückgang der Modellgenauigkeit

oder eine steigende Fehlerrate wird dabei als potenzieller Hinweis auf das Auftreten eines Drifts interpretiert. Bayram et al. (2022) teilen diese Ansätze nach den Strategien zur Erkennung von Leistungsverlusten in drei Hauptkategorien ein: statistische Prozesskontrolle, Fenstertechniken und Ensemble-Lernen, die in Abbildung 4 dargestellt sind:

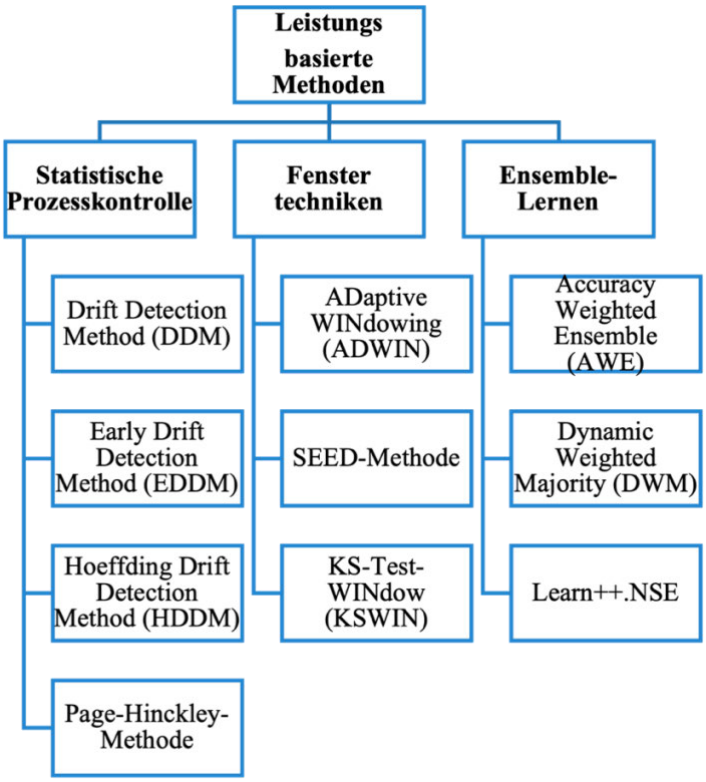


Abbildung 4: Hierarchie der leistungsbasierten Ansätze zur Drifterkennung

Quelle: Eigene Darstellung

4.1 Statistische Prozesskontrolle

„Die Statistische Prozesskontrolle (SPC) wird eingesetzt, um die Qualität des Lernprozesses zu überwachen, indem die Modellleistung kontinuierlich analysiert wird. Dabei wird die Fehlerrate eines sogenannten Basislernenden (engl. base learner) überprüft, um signifikante Verschlechterungen in der Modellgenauigkeit zu erkennen. Ein Modelldrift wird angenommen, sobald der beobachtete Leistungsverlust ein definiertes Signifikanzniveau überschreitet“ (Bayram et al., 2022).

4.1.1 Drift Detection Method (DDM)

Ein prominenter und weit verbreiteter Algorithmus in diesem Bereich ist die Drift Detection Method DDM (Gama et al., 2004), die als *„konzeptionelle Grundlage für viele ähnliche leistungsbasierte Drift-Detektoren“* dient. (Bayram et al., 2022) *„Diese Methode überwacht die Fehlerrate eines Streaming-Daten-Klassifikators und nutzt statistische Eigenschaften, um Veränderungen zu erkennen. DDM behandelt die Fehlerrate als Bernoulli-Zufallsvariable mit einer Binomialverteilung“* (Bayram et al.,

2022). Zum Zeitpunkt t wird die Wahrscheinlichkeit einer Fehlklassifikation (p_t) sowie deren Standardabweichung (s_t) berechnet:

$$s_t = \sqrt{\left(\frac{p_t(1-p_t)}{i}\right)} \tag{11}$$

Formel 11: Standardabweichung einer Fehlklassifikation zum Zeitpunkt t

Quelle: Bayram et al. (2022)

Dabei werden kontinuierlich Minimalwerte p_{min} und s_{min} aktualisiert, wenn $p_t + s_t < p_{min} + s_{min}$. Es werden zwei Zustände definiert:

1. Ein Warnzustand, der ausgelöst wird, wenn $p_t + s_t > p_{min} + 2 \cdot s_{min}$
2. Ein Driftzustand, der eintritt, wenn $p_t + s_t > p_{min} + 3 \cdot s_{min}$

Die Schwellenwerte für den Warn- und Driftzustand ($2 \cdot s_{min}; 3 \cdot s_{min}$) sind standardmäßig festgelegt und basieren auf statistischen Prinzipien, um eine verlässliche Drift-Erkennung zu gewährleisten. In der Praxis können diese Werte jedoch angepasst werden, um die Methode an spezifische Anwendungsfälle anzupassen. Beispielsweise könnten die Schwellenwerte in stabilen Umgebungen erhöht werden, um Falschalarme zu reduzieren, oder in sicherheitskritischen Szenarien gesenkt werden, um eine schnellere Reaktion auf Drifts zu ermöglichen.

4.1.2 Early Drift Detection Method (EDDM)

Um die Leistungsfähigkeit von DDM in verschiedenen Szenarien zu erhöhen, wurde zum Beispiel Early Drift Detection Method EDDM (Baena-Garcıa, M., 2006) entwickelt. *„EDDM erweitert DDM, indem es nicht nur die Fehlerrate überwacht, sondern auch den Abstand zwischen aufeinanderfolgenden Fehlklassifikationen“* (Bayram et al., 2022).

Der durchschnittliche Abstand zwischen zwei Fehlern (p'_i) und seine Standardabweichung (s'_i) werden berechnet. Gespeichert werden die Werte von (p'_i) und (s'_i), wenn ($p'_i + 2 \cdot s'_i$) seinen maximalen Wert erreicht (p'_{max}) und (s'_{max}). Der Wert ($p'_{max} + 2 \cdot s'_{max}$) gibt den Punkt an, an dem die Verteilung der Fehlerabstände am größten ist. Dies tritt auf, wenn das Modell die aktuellen Konzepte in den Daten am besten erfasst. Es werden auch zwei Zustände definiert:

1. Warnstufe: Falls $\frac{(p'_i + 2 \cdot s'_i)}{(p'_{max} + 2 \cdot s'_{max})} < \alpha$
2. Driftstufe: Falls $\frac{(p'_i + 2 \cdot s'_i)}{(p'_{max} + 2 \cdot s'_{max})} <$ β, wird ein Konzeptdrift erkannt. Das aktuelle Modell wird zurückgesetzt und ein neues Modell mit den seit der Warnstufe gespeicherten Beispielen wird trainiert. Gleichzeitig werden (p'_{max}) und (s'_{max}) zurückgesetzt.

Baena-García, M., 2006 legt die Schwellenwerte α und β auf 0,95 bzw. 0,90 fest. Diese Werte wurden durch mehrere Experimente bestimmt.

„Early Drift Detection Method EDDM eignet sich besonders für die Erkennung von allmählichen Drifts“ (Bayram et al., 2022), da er sensibler auf subtile Veränderungen in den Daten reagiert.

4.1.3 Hoeffding Drift Detection Method (HDDM)

Eine weitere Erweiterung der DDM erfolgt durch den Hoeffding Drift Detection Method HDDM (Frias-Blanco et al., 2014). *„Dieser modifiziert die DDM, indem er Hoeffdings Ungleichung nutzt, um signifikante Veränderungen im sich bewegenden Durchschnitt einer Leistungsbewertung zu erkennen“* (Bayram et al., 2022). Die Ungleichung lautet:

$$P(|\bar{X} - \mu| \geq \epsilon) \leq 2\exp(-2n\epsilon^2) \tag{12}$$

Formel 12: Hoeffding's Ungleichung

Quelle: Hoeffding (1963)

Hoeffding's Ungleichung beschreibt, dass die Wahrscheinlichkeit, dass der Unterschied zwischen dem geschätzten Durchschnittswert $\bar{X}$ und dem wahren Erwartungswert μ größer als ein gewisser Schwellenwert ϵ ist, durch die Formel $2\exp(-2n\epsilon^2)$ begrenzt wird, wobei n die Anzahl der Stichproben ist.

Frias-Blanco et al. (2014) stellen 2 Varianten von HDDM vor:

1. HDDM_A: Diese Variante nutzt ein gleitendes Fenster und berechnet den Mittelwert in den zwei Fenstern. Eine Drift wird erkannt, wenn die Differenz der Mittelwerte einen Hoeffding-basierten Schwellenwert überschreitet.

$$|\overline{X_R} - \overline{X_T}| \geq \epsilon \tag{13}$$

Formel 13: HDDM_A

Quelle: Frias-Blanco et al. (2014)

Hierbei sind $(\overline{X_R})$ der Mittelwert des Referenzfensters und $(\overline{X_T})$ der Mittelwert des Testfensters. HDDM_A eignet sich besonders gut für die Erkennung plötzlicher Drifts, da es direkt auf signifikante Unterschiede reagiert.

2. HDDM_W: Diese Variante verwendet ein gewichtetes gleitendes Fenster, bei dem neuere Beobachtungen stärker gewichtet werden. Dies macht sie besonders geeignet für die Erkennung von allmählichen Drifts, da sich die Gewichtung an neue Trends anpasst.

4.1.4 Page-Hinckley-Methode

Eine weit verbreitete Methode der statistischen Prozesskontrolle (SPC) ist die Page-Hinckley-Methode PH (Mouss, H et al., 2004)., die auf der Analyse kumulativer Abweichungen im Zeitverlauf basiert. Ihr Hauptziel ist es, Abweichungen vom Mittelwert frühzeitig zu erkennen, indem kontinuierlich Veränderungen überwacht werden. Die zentrale Formel der Page-Hinckley-Methode lautet:

$$PH(t) = \max_{j=1,\dots,t} \left(\sum_{i=1}^{t} x_i - m_j - \lambda \cdot t \right) \tag{14}$$

Formel 14: Page-Hinckley

Quelle: Mouss. H et al. (2004).

Hierbei gilt

- x_i Beobachtete Datenpunkte, wie z. B. Fehlerraten;
- m_j Gleitender Durchschnitt bis zum Zeitpunkt j, berechnet als ($m_j = \frac{1}{j} \sum_{i=1}^{j} x_i$)
- λ Ein Schwellenwert, der die Sensitivität der Methode bestimmt.

Eine Drift wird erkannt, wenn ($PH(t)$) einen vorab definierten Grenzwert überschreitet. Die Methode ist besonders nützlich für die Erkennung allmählicher Veränderungen in Streaming-Daten.

4.2 Fenstertechniken

„Fenstertechniken basieren darauf, den Datenstrom in Fenster zu unterteilen, die entweder durch eine feste Datengröße oder ein Zeitintervall definiert werden. Diese Fenster werden in einer gleitenden Weise verschoben, sodass die Leistung des Modells auf den neuesten Daten beobachtet und mit einem Referenzfenster verglichen werden kann“ (Bayram et al., 2022).

4.2.1 ADaptive WINdowing (ADWIN)

Eine der bekanntesten Methoden in diesem Bereich ist ADaptive WINdowing ADWIN (Bifet & Gavalda, 2007), „*die mithilfe der Hoeffding-Schranke (ϵ_{cut}) die Unterschiede zwischen den Mittelwerten zweier Unterfenster W_{hist} und W_{new} analysiert“* (Bayram et al., 2022). Eine Drift wird erkannt, wenn „die Differenz der Mittelwerte μ die Schranke $2 \cdot \epsilon_{\text{cut}}$ überschreitet ($\mu_{hist} - \mu_{new} > 2 \cdot \epsilon_{\text{cut}}$), wobei der Grenzwert wie folgt berechnet wird:

$$\epsilon_{\text{cut}} = \sqrt{\frac{1}{2m} \ln \frac{4|W|}{\delta}} \tag{15}$$

Formel 15: Hoeffding-Schranke der ADWIN

Quelle: Bayram et al. (2022)

Hierbei ist m der harmonische Mittelwert der beiden Fenster und $|W|$ die Größe des Gesamtfensters. δ ist ein Konfidenzparameter, der die Sensitivität der Drifterkennung steuert. Ein kleinerer Wert (z. B. $\delta = 0{,}01$ entspricht einem Konfidenzniveau von 99 %) erhöht das Vertrauen in die Erkennung und verringert Fehlalarme, während ein größerer Wert (z. B. $\delta = 0{,}1$ entspricht einem Konfidenzniveau von 90 %) die Reaktion auf Drifts beschleunigt.

4.2.2 SEED-Methode

Ein weiteres Beispiel ist die SEED-Methode (Huang et al., 2014), die ebenfalls auf ADWIN aufbaut, jedoch nicht den Mittelwert der Fenster vergleicht, sondern die Klassifikationsfehler zwischen zwei Unterfenstern. *„SEED verwendet zwei Unterfenster, ein linkes Fenster W_L und ein rechtes Fenster W_R, innerhalb eines Gesamtfensters W. Der Algorithmus analysiert die binären Klassifikationsentscheidungen (1 für korrekt, 0 für fehlerhaft) und nutzt auch die Hoeffding-Ungleichung, um ϵ_{cut} zu berechnen“* (Bayram et al., 2022). SEED verwendet zusätzlich die Bonferroni-Korrektur, um die Schwellenwerte bei mehreren Vergleichen von Unterfenstern innerhalb eines Fensters anzupassen. Im Gegensatz zu ADWIN, das auf dem Vergleich der Mittelwerte zwischen zwei Fenstern W_{hist} und W_{new} basiert, kombiniert SEED diese Korrektur, um Fehlalarme zu reduzieren und gleichzeitig die Empfindlichkeit zu erhöhen.

4.2.3 Kolmogorov-Smirnov-Test-WINdow (KSWIN)

Ein weiteres Verfahren ist der Kolmogorov-Smirnov-Test-WINdow (KSWIN), der den Kolmogorov-Smirnov-Test verwendet, um Veränderungen in Datenströmen zu erkennen. KSWIN basiert auf der Annahme, dass die Daten in einem Referenzfenster (R) und in einem Testsfenster (T) aus der gleichen Verteilung stammen, solange keine Drift auftritt. Der Algorithmus prüft die kumulativen Verteilungsfunktionen (CDFs) der beiden Fenster und berechnet die maximale Differenz zwischen ihnen (D). Eine Drift wird erkannt, wenn $(D > \epsilon_{KS})$, wobei ϵ_{KS} der Schwellenwert ist, der durch die Größe der Fenster und ein Konfidenzniveau α bestimmt wird. Im Vergleich zu ADWIN und SEED ist KSWIN nicht auf Mittelwerte oder Fehler beschränkt, sondern kann Veränderungen in beliebigen Verteilungen erkennen und ist daher vielseitig einsetzbar.

4.3 Ensemble-Lernen

„Ensemble-basierte Modelldrift -Detektoren kombinieren die Ergebnisse mehrerer Basislernmodelle, um eine Gesamtleistung zu erzielen. Die Leistung des Ensembles wird entweder durch die Berücksichtigung der Genauigkeit aller Mitglieder oder der einzelnen Basislernmodelle überwacht“ (Bayram et al., 2022). Dieser Ansatz basiert auf der Annahme, dass jedes Modell spezifische Fähigkeiten besitzt, um unterschiedliche Aspekte des Problems zu lösen.

4.3.1 Accuracy Weighted Ensemble (AWE)

Ein bekanntes Beispiel für einen ensemblebasierten Detektor ist der Accuracy Weighted Ensemble (AWE)-Algorithmus (Wang et al., 2003). *„AWE wählt die besten Modelle aus, indem er eine spezielle Version des mittleren quadratischen Fehlers verwendet, die Wahrscheinlichkeiten nutzt, um die geeignetsten Modelle zu bestimmen und veraltete Modelle mit der größten Leistungsverschlechterung zu verwerfen“* (Bayram et al., 2022). Dadurch wird die Genauigkeit des Ensembles im Laufe der Zeit verbessert, indem weniger effektive Modelle entfernt werden. Dieser Ansatz ermöglicht eine flexible Anpassung an Modelldrift.

4.3.2 Dynamic Weighted Majority (DWM)

Ein weiterer wichtiger Algorithmus in dieser Gruppe ist der Dynamic Weighted Majority (DWM)-Algorithmus (Kolter et al., 2007). *„DWM verwendet einen Gewichtungsmechanismus, der auf dem Weighted Majority Algorithmus (WMA) basiert. Jedes Modell im Ensemble erhält eine Gewichtung β ($0 \leq \beta \leq 1$), die nach einer falschen Vorhersage reduziert wird“* (Bayram et al., 2022). So wird der Einfluss von Modellen, die schlecht performen, verringert, und das Ensemble kann sich besser an Veränderungen im Datenstrom anpassen.

Der Unterschied zwischen dem DWM und seiner Grundlage, der WMA, liegt im Gewichtungsfaktor: Während die Gewichtungsanpassung in WMA in der Regel statisch erfolgt und keine Rücksicht auf die Schwere der Fehler oder die Verbesserungsgeschwindigkeit nimmt, passt DWM die Gewichtungen dynamisch an die Vorhersageleistung nach jedem Zeitschritt an.

Beide Algorithmen, AWE und DWM, nutzen gewichtete Mechanismen zur Auswahl und Anpassung von Modellen basierend auf deren Leistung. Sie unterscheiden sich jedoch in der Art und Weise, wie diese Gewichtungen angepasst werden: AWE verwendet eine Wahrscheinlichkeits-basierte Methode zur Modellselektion, während DWM die Gewichtungen dynamisch nach jedem Schritt basierend auf den Fehlern des Modells anpasst. DWM zeigt daher Stärken bei schrittweisen oder wiederkehrenden Drifts, da es Modelle flexibel anpasst, ohne die Ensemble-Struktur drastisch zu verändern. AWE ist hingegen vorteilhaft, wenn abrupte Drifts auftreten, da es schnell alte Modelle entfernt und sich neu aufstellt.

4.3.3 Learn++.NSE

Ein weiteres bekanntes ensemblebasiertes Verfahren zur Erkennung von Konzeptdrift ist Learn++.NSE („Incremental Learning for Non-Stationary Environments“). *„Learn++.NSE ist die erste Version der bemerkenswerten Learn++-Algorithmusfamilie, die speziell für die Handhabung von Konzeptdrift entwickelt wurde“* (Polikar et al., 2001).

Der Algorithmus trainiert ein Ensemble von Modellen, wobei jedes Modell auf verschiedenen Datenblöcken („Chunks“) basiert. Die Trainingsbeispiele werden dabei gewichtet, abhängig von den Fehlern des Ensembles bei der Klassifikation dieser Beispiele. Wenn ein Beispiel korrekt klassifiziert wird, erhält es ein Gewicht von 1. Andernfalls wird es durch einen Faktor ($w_i = 1/e$) bestraft. Die Gewichtung der Modelle im Ensemble erfolgt mithilfe der Sigmoidfunktion, die die Fehler der Modelle auf alten und aktuellen Datenblöcken berücksichtigt.

Learn++.NSE zeigt Stärken in Umgebungen mit wiederkehrenden Drifts, da es die Modelle im Ensemble beibehält und ihre Relevanz dynamisch anpasst. Gleichzeitig sorgt die flexible Gewichtung dafür, dass aktuelle Veränderungen im Datenstrom berücksichtigt werden, ohne ältere Modelle vollständig zu verwerfen.

4.4 Fazit der leistungsbasierten Ansätze

Zusammenfassend lässt sich sagen, dass statistische Prozesskontrolle (SPC) eine effektive Methode zur Erkennung von Modelldrift bietet, indem sie die Modellleistung analysiert. Prominente Algorithmen basieren auf statistischen Prinzipien wie Bernoulli- und Binomialverteilungen sowie der Hoeffding-Ungleichung, um signifikante Veränderungen in Datenströmen zu identifizieren.

Verfahren wie ADWIN und KSWIN verbessern die Sensitivität durch Fenstertechniken und Verteilungsvergleiche.

Das Ensemble-Lernen ergänzt diese Ansätze durch die Kombination mehrerer Modelle, wodurch eine höhere Flexibilität bei der Anpassung an Modelldrift ermöglicht wird.

Jede der drei Gruppen leistungsbasierter Ansätze hat allerdings ihre eigenen Stärken und Herausforderungen:

- Statistische Prozesskontrolle und Fenstertechniken sind weit verbreitet und bieten eine effektive Lösung, wenn die Fehlerkennzahlen zuverlässig sind und klare Drift-Signale liefern. In stabilen Umgebungen funktionieren sie gut, stoßen jedoch an ihre Grenzen, wenn die Fehlerkennzahlen in dynamischen Szenarien oder bei unbalancierten Klassen unzuverlässig werden, was zu Fehlalarmen und unnötigen Reaktionen führen kann. *„Neuere Ansätze, die auf der Konfusionsmatrix oder dem AUC-Wert basieren, bieten in solchen Fällen präzisere Metriken, insbesondere bei unausgewogenen Datensätzen“ (*Bayram et al., 2022).
- Ensemble-basierte Ansätze sind robuster und flexibler, jedoch zeigen Studien, dass „die Kombination von Modellen nicht immer zu besseren Ergebnissen führt“ (Woźniak et al., 2016), insbesondere wenn die Gewichtungsmechanismen nicht sorgfältig kalibriert sind.

Nach Bayram et al., (2022) ist ein weiteres kritisches Thema bei leistungsbasierten Ansätzen die Auswahl der Basislernmodelle. Traditionelle Modelle wie Hoeffding-Bäume und Naive Bayes haben

sich als effizient erwiesen, um massive Datenströme zu verarbeiten. Neuronale Netze gewinnen zunehmend an Bedeutung, sind jedoch mit Herausforderungen konfrontiert, wie der Schwierigkeit, ihre Architektur in Echtzeit anzupassen, sowie „der mangelnden Transparenz und Interpretierbarkeit“ (Buhrmester et al., 2021 und Wang et al., 2019), was die Nachvollziehbarkeit von Driftereignissen erschwert.

„Die Anwendung leistungsbasierter Ansätze auf komplexe, semi-supervisierte oder unüberwachte Probleme ist ebenfalls problematisch“ (Bayram et al., 2022), da wahre Labels nicht zur Verfügung stehen. In Szenarien ohne wahre Labels müssen Detektoren Schätzungen verwenden, was zu Unsicherheiten führen kann.

5 Mehrfachhypothesen-basierte Ansätze

„Mehrfachhypothesen-basierte Ansätze kombinieren mehrere Erkennungsmethoden, um eine höhere Genauigkeit und Anpassungsfähigkeit zu erzielen. Die Ergebnisse der verschiedenen Methoden werden dabei entweder parallel oder hierarchisch zusammengefasst“ (Bayram et al., 2022).

5.1 Parallele Mehrfachhypothesen- Driftdetektoren

„Parallele Mehrfachhypothesen-Driftdetektionsalgorithmen (Abbildung 5) kombinieren die Ergebnisse verschiedener Drifterkennungsmethoden, um eine endgültige Entscheidung darüber zu treffen, ob ein Modelldrift vorliegt“ (Lu et al., 2018).

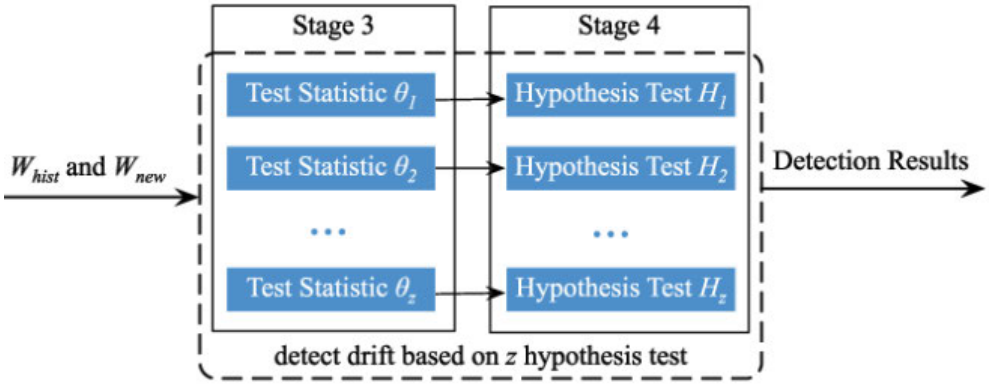


Abbildung 5: Parallele Mehrfachhypothesen - Driftdetektoren

Quelle: Lu et al. (2018)

5.1.1 Beispiel: Dreischichtige Drift-Erkennung IV- Jac

Ein Beispiel für einen solchen Algorithmus ist die dreischichtige Drift-Erkennung, die auf den Konzepten des Informationswerts (IV) und der Jaccard-Ähnlichkeit basiert (IV-Jac). Dieser Algorithmus wurde von Zhang et al. (2017) vorgestellt und adressiert drei unterschiedliche Arten von Drift, wobei jede in einer separaten Schicht behandelt wird:

1. Label-Drift (Schicht I): Hier wird untersucht, ob sich die Verteilung der Labels (Zielvariablen) im Laufe der Zeit verändert hat. *„Alle Drifts, die durch Änderungen im Labelraum entstehen, werden in dieser Schicht erkannt. Dadurch können in den folgenden Schichten nur noch Daten mit denselben Labels berücksichtigt werden, um Feature-Space-Drift und Entscheidungsgrenzen-Drift zu erkennen“* (Zhang et al. 2017).
2. Feature-Space-Drift (Schicht II): In dieser Schicht wird geprüft, ob sich die Merkmale (Input-Variablen) im Datenstrom verändert haben. Dazu wird der Informationswert (IV) extrahiert, der angibt, wie stark ein Feature zur Erklärung des Modells beiträgt.
3. Entscheidungsgrenzen-Drift (Schicht III): *„Wenn Label- und Merkmalsräume stabil sind, wird in dieser Schicht überprüft, ob sich die Zuordnungsbeziehung zwischen Labels und Features verändert hat“* (Zhang et al. 2017). Hierbei wird der Weight of Evidence (WoE) herangezogen, der misst, wie stark ein Feature die Zielvariable beeinflusst.

Jede Schicht evaluiert historische und aktuelle Werte (z. B. IV für Schicht II und WoE für Schicht III) und vergleicht diese mit festgelegten Schwellenwerten. Die Anwendung von Schwellenwerten wie t ermöglicht es, zwischen Drift und Rauschen zu differenzieren, wobei hohe Schwellenwerte zu weniger Fehlalarmen führen, aber auch die Sensitivität für kleinere Veränderungen verringern können.

Abbildung 6 zeigt eine schematische Darstellung der dreischichtigen Drift-Erkennung, wie in der Arbeit von Zhang et al. (2017) beschrieben.

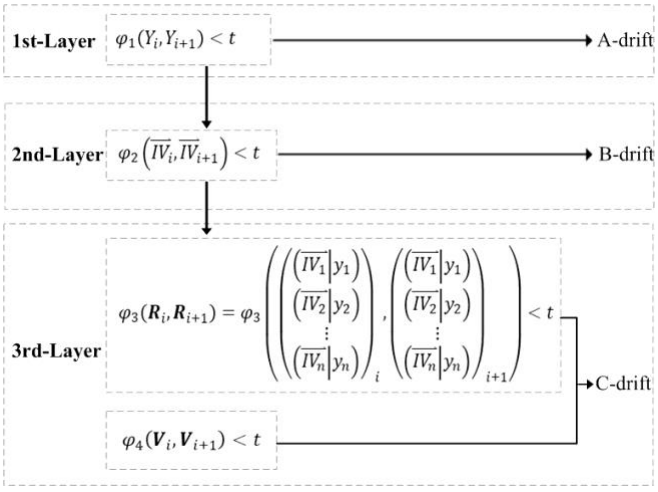


Abbildung 6: Das Drei-Schichten-Modell zur Erkennung von Konzeptdrift

Quelle: Zhang et al. (2017)

5.1.2 Vorteile und Nachteile der dreischichtigen Drift-Erkennung

Im Vergleich zu anderen Drift-Detektionsmethoden bietet IV-Jac mehrere Vorteile: Es zeigt eine hohe Effektivität in hochdimensionalen Datenströmen, da die Nutzung des IV-Wertes eine effiziente Identifikation relevanter Merkmale ermöglicht. Zudem weist der Algorithmus eine geringere Fehlalarmrate auf, insbesondere in häufig driftenden Datenströmen, was seine Präzision unterstreicht. Darüber hinaus verfügt das Modell über eine klar definierte Schichtstruktur, die modular aufgebaut ist und an spezifische Anforderungen angepasst werden kann.

Das IV-Jac-Modell hat einige Nachteile, darunter einen höheren Zeitaufwand für die Drift-Detektion, da Berechnungen wie IV-Werte und Jaccard-Ähnlichkeiten erforderlich sind, insbesondere in der dritten Schicht. Die Genauigkeit hängt stark von der optimalen Wahl der Schwellenwerte (t) ab, was manuelle Feinabstimmung erfordern kann. Zudem bietet die dritte Schicht, die seltene Driftarten wie Entscheidungsgrenzen-Drift behandelt, in vielen Anwendungsfällen keinen signifikanten Vorteil und führt zu zusätzlichem Rechenaufwand.

Das Modell ist auf gelabelte Datenströme beschränkt und somit nicht direkt für unlabeled oder semi-supervised Szenarien geeignet. Bei stark verrauschten Daten kann die Unterscheidung zwischen echten Drifts und Rauschen schwierig sein, was die Präzision beeinträchtigen kann.

5.2 Hierarchische Mehrfachhypothesen- Driftdetektoren

„Hierarchische Mehrfachhypothesen-Driftdetektionsalgorithmen (Abbildung 7) arbeiten in zwei Ebenen: einer Warnebene, die potenzielle Modelldrift signalisiert, und einer Validierungsebene, die diese Signale überprüft“ (Lu et al., 2018).

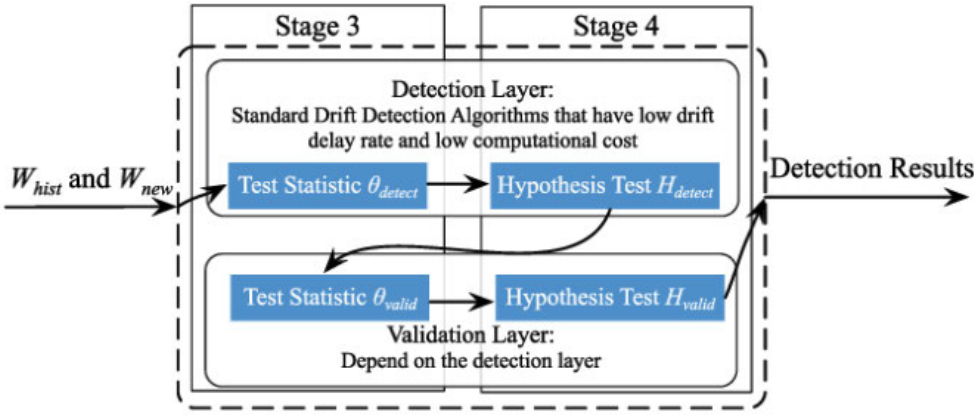


Abbildung 7: Hierarchische Mehrfachhypothesen - Driftdetektoren

Quelle: Lu et al. (2018)

5.2.1 Beispiel: Hierarchical Linear Four Rate (HLFR)

Ein bedeutendes Beispiel ist der Hierarchical Linear Four Rate (HLFR)-Algorithmus (Abbildung 8), der von Yu & Abraham (2017) vorgestellt wurde.

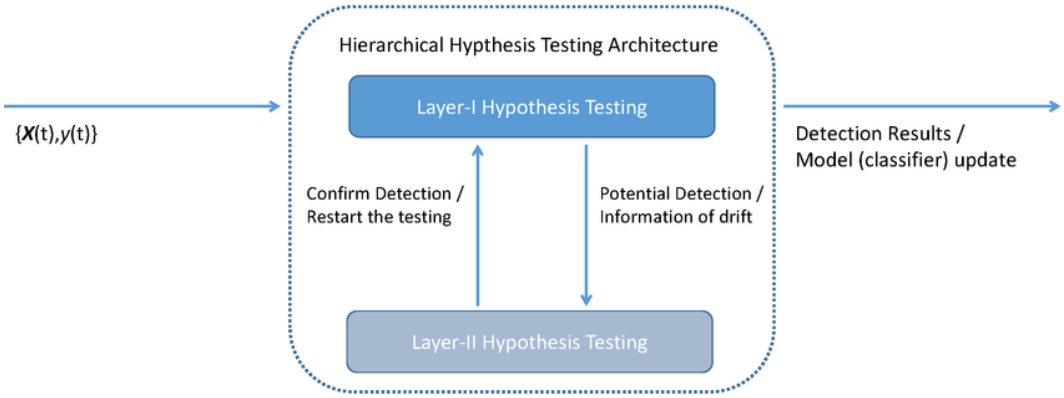


Abbildung 8: Die Architektur des hierarchischen Hypothesentests zur Erkennung von Konzeptdrift

Quelle: Yu & Abraham (2017)

In der ersten Ebene verwendet das System den Linear Four Rate (LFR)-Algorithmus, der Veränderungen in der Datenverteilung mithilfe von statistischen Tests auf Drift erkennt. Dabei werden vier wichtige Parameter der Konfusionsmatrix (True Positive Rate, True Negative Rate, Positive Predictive Value und Negative Predictive Value) überwacht, um Warnsignale für potenzielle Driftpunkte zu generieren.

Die zweite Schicht der HLFR verwendet Teststatistiken, die eng mit denen der Schicht I verwandt sind, um einen Permutationstest durchzuführen. Das Permutationstest nutzt den Null-Eins-Verlust als Bewertungsmaßstab.

Wenn eine Drift bestätigt wird, meldet die HLFR eine Detektion, andernfalls wird das Ergebnis der Detektion in Schicht I als falsch-positiv gewertet und der Test beginnt erneut, um die nächsten Daten zu bewerten.

5.2.2 Vorteile und Nachteile des HLFR

Ein zentrales Merkmal des HLFR-Algorithmus ist seine Unabhängigkeit vom zugrunde liegenden Klassifikator, wodurch er universell in verschiedenen Anwendungsszenarien eingesetzt werden kann. *„HLFR behandelt den zugrundeliegenden Klassifikator als Black Box und macht keinen Gebrauch von seinen inhärenten Eigenschaften“* (Yu & Abraham, 2017).

Zusammenfassend zeigt der HLFR-Algorithmus eine herausragende Leistung bei der Detektion verschiedener Arten von Konzeptdrift, einschließlich jener in unbalancierten Datenströmen. Er minimiert Falschalarme, garantiert jedoch gleichzeitig hohe Präzision und kurze Erkennungsverzögerungen. Dies macht ihn besonders wertvoll für Echtzeit-Anwendungen wie Spam-Filterung und Betrugserkennung, in denen sich Datenbeziehungen dynamisch verändern.

Ein Nachteil des HLFR-Algorithmus ist der erhöhte Rechenaufwand, der durch die zweistufige Architektur entsteht, insbesondere bei großen oder hochdimensionalen Datenströmen.

Die Durchführung von Permutationstests in der zweiten Ebene, um potenzielle Drifts zu validieren, kann zeit- und ressourcenintensiv sein. Dies könnte in Echtzeitanwendungen mit strengen Latenzanforderungen problematisch sein.

Zudem ist der Algorithmus zwar flexibel in Bezug auf den eingesetzten Klassifikator, jedoch kann die Auswahl geeigneter Schwellenwerte und Parameter, wie z. B. der Signifikanzniveaus, komplex sein und eine sorgfältige Feinabstimmung erfordern, um optimale Ergebnisse zu erzielen.

Schließlich ist der HLFR-Ansatz möglicherweise weniger effektiv in Szenarien mit sehr schnellen, abrupten Drifts, da die Validierung in der zweiten Ebene zusätzliche Verzögerungen verursachen kann.

5.3 Fazit der Mehrfachhypothesen-basierten Ansätze

Mehrfachhypothesen-basierte Ansätze bieten eine vielversprechende Methodik zur Erkennung von Konzeptdrift in Datenströmen, da sie die Stärken verschiedener Erkennungsmethoden kombinieren sowohl parallele als auch hierarchische Strategien umfassen.

Während parallele Ansätze wie der IV-Jac-Algorithmus durch eine modulare Struktur und hohe Präzision überzeugen, eignen sie sich besonders für komplexe, hochdimensionale Daten. Dennoch können sie durch ihren hohen Rechenaufwand und die Abhängigkeit von optimalen Schwellenwerten limitiert sein.

Hierarchische Ansätze, wie der HLFR-Algorithmus, zeichnen sich durch eine effektive Driftvalidierung und universelle Anwendbarkeit aus, was sie ideal für dynamische Echtzeitanwendungen macht. Allerdings bringen sie ebenfalls Herausforderungen mit sich, wie erhöhte Latenzzeiten und die Notwendigkeit einer sorgfältigen Parameterabstimmung.

Zusammenfassend lässt sich sagen, dass die Wahl des passenden Ansatzes stark von den spezifischen Anforderungen des Anwendungsfalls abhängt. Beide Ansätze zeigen, dass eine Kombination aus Flexibilität, Präzision und Robustheit entscheidend für eine effektive Driftdetektion ist, erfordern jedoch eine ausgewogene Abwägung zwischen Leistungsfähigkeit und Ressourcenaufwand.

6 Kontextbasierte Ansätze

„Kontextbasierte Ansätze zur Drifterkennung nutzen Informationen aus dem System und den Daten, um Veränderungen in den zugrunde liegenden Konzepten zu identifizieren“ (Bayram et al., 2022). Ziel dieser Methoden ist es, ein tieferes Verständnis für das Verhalten und die Evolution des Modells im Laufe der Zeit zu entwickeln. Sie analysieren nicht nur Datenmerkmale oder Leistungsmetriken, sondern berücksichtigen auch die Dynamik der Systemreaktionen auf neue Eingabedaten.

6.1 Beispiel: Evolutionäre Spiking Neural Network Drift Detection (eSNN-DD)

Ein bemerkenswertes Beispiel für einen kontextbasierten Ansatz ist die eSNN-DD-Methode (evolutionäre Spiking Neural Network Drift Detection), die von Lobo et al. (2018) vorgestellt wurde. Diese Methode basiert auf Evolving Spiking Neural Networks eSNNs (Schliebs & Kasabov, 2013) (siehe Abbildung 9).

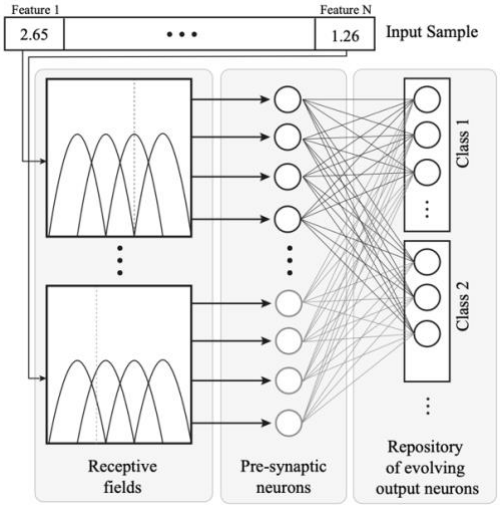


Abbildung 9: Schema eines eSNN

Quelle: Kasabov (2007)

Evolving Spiking Neural Networks (eSNNs) sind eine Klasse von Spiking Neural Networks, die Informationen durch zeitliche „Spikes“ verarbeiten und ihre Architektur dynamisch an neue Daten anpassen. Sie bestehen aus drei Schichten:

1. Eingabeneuronen, die Daten in Spike-Zeitreihen kodieren,
2. prä-synaptischen Neuronen und
3. einer sich entwickelnden Schicht von Ausgabeneuronen, die Muster in den Daten repräsentieren.

Die Lernmethode basiert auf dem „Leaky Integrate-and-Fire“-Modell, bei dem Neuronen feuern, sobald ihr postsynaptisches Potenzial einen Schwellenwert überschreitet. Neue Datenproben führen entweder zu neuronalen Verschmelzungen mit bestehenden Mustern oder zu neuen Neuronen, wenn sie neuartige Konzepte darstellen. Dadurch können eSNNs zeitliche Muster in Datenströmen erkennen und sich an nicht-stationäre Umgebungen anpassen, ohne ein separates Basismodell zu benötigen.

Die eSNN-DD-Methode des kontextbasierten Ansatzes nutzt dieses dynamische Modell, um Konzeptdrift zu erkennen, indem Änderungen in der Verschmelzungsdynamik der Neuronen überwacht werden. Wenn neue Daten auf ein Netzwerk treffen und die Muster nicht mehr mit den bestehenden Neuronen kompatibel sind, deutet dies auf einen Konzeptdrift hin. Nach einem Driftereignis finden zunächst weniger Verschmelzungen von Neuronen statt, bis das Netzwerk die neue Datenverteilung vollständig erlernt hat. Dies ermöglicht eine frühzeitige Identifikation von Drifts und eine entsprechende Anpassung des Modells.

6.2 Vorteile und Nachteile der eSNN-DD

Die Vorteile der eSNN-DD-Methode liegen in ihrer Fähigkeit, Konzeptdrifts ohne zusätzliche Basismodelle oder externe Leistungsmetriken zu erkennen. Sie nutzt ausschließlich interne architektonische Änderungen, was sie besonders effizient und ressourcenschonend für Szenarien mit beschränkten Rechenkapazitäten macht. Besonders hervorzuheben ist ihre Fähigkeit, Drift frühzeitig zu erkennen, da sie direkt auf die strukturellen Änderungen im Netzwerk reagiert.

Experimente mit synthetischen Datensätzen zeigen, dass eSNN-DD in Bezug auf wahre Positive, falsche Alarme und Distanz zum Driftereignis wettbewerbsfähig ist.

Ein Nachteil der eSNN-DD-Methode ist jedoch der erhöhte Rechenaufwand im Vergleich zu traditionelleren Ansätzen wie ADWIN oder HDDM, insbesondere bei großen Datenströmen.

Dennoch bietet die Methode eine wertvolle Grundlage für zukünftige Entwicklungen, wie z. B. die Einbeziehung von Online-Optimierungsmechanismen, um die Netzwerkparameter während des Betriebs dynamisch anzupassen.

Insgesamt bieten kontextbasierte Ansätze wie die eSNN-DD-Methode signifikante Vorteile gegenüber datenverteilungsbasierten und leistungsbasierten Methoden, da „sie eine frühzeitigere Erkennung von Drift ermöglichen“ (Lobo et al., 2018), indem sie die Entwicklung des Modells und dessen Reaktionen auf neue Daten analysieren. Dies reduziert das Risiko signifikanter Leistungseinbußen und ermöglicht eine dynamische Anpassung des Modells. Der Nachteil kontextbasierter Ansätze liegt jedoch in ihrem erhöhten Rechenaufwand und der komplexeren Modellstruktur, was ihre Anwendung in bestimmten praktischen Bereichen erschwert.

7 Experimentelle Untersuchung und Ergebnisse

Anschließend wurden in dieser Arbeit die zwei wichtigsten Gruppen von Ansätzen zur Drifterkennung, nämlich die datenverteilungsbasierten und leistungsbasierten Ansätze, durch eine Reihe gezielter Experimente untersucht.

Im Folgenden werden die Experimente beschrieben und die wichtigsten Ergebnisse zusammengefasst.

7.1 Datenverteilungsbasierte Ansätze

7.1.1 Experimenteller Aufbau und Ablauf

Das erste Experiment wurde durchgeführt, um die Sensitivität und Robustheit verschiedener datenverteilungsbasierter Tests zur Drifterkennung in univariaten Daten zu untersuchen. Hierfür wurden fünf statistische Tests verwendet: Hellinger-Distanz, Kullback-Leibler-Divergenz, Kolmogorov-Smirnov-Test, Wasserstein-Distanz und T-Test. Jeder Test wurde mit spezifischen Schwellenwerten konfiguriert, um eine ausgewogene Detektion zu ermöglichen, ohne unnötige Fehlalarme zu erzeugen.

Tabelle 1: Schwellenwerte für Tests

Quelle: Eigene Darstellung

Test	Schwellenwert	Begründung für Schwellenwert
Hellinger-Distanz	0.1	Höherer Schwellenwert erlaubt Erkennung subtiler Änderungen.
KL-Divergenz	0.15	Höherer Schwellenwert verhindert Überempfindlichkeit.
KS-Test	0.05	Standard-Signifikanzniveau für Unterschiede in kumulativen Verteilungen.
Wasserstein-Distanz	0.1	Moderater Schwellenwert für mittlere Änderungen.
T-Test	0.05	Standard-Signifikanzniveau erkennt nur wesentliche Unterschiede.

Unterschiedliche Schwellenwerte sind sinnvoll, da verschiedene statistische Tests unterschiedliche Sensitivitäten haben. Tests wie der Kolmogorov-Smirnov-Test oder der T-Test sind empfindlicher gegenüber kleineren Abweichungen und nutzen niedrigere Schwellenwerte, während Tests wie der Hellinger-Distanz oder die Wasserstein-Distanz strukturelle Unterschiede erfassen und benötigen daher oft höhere Schwellenwerte. Wichtig ist nur die Konsistenz der Ziele: Alle Tests sollten auf denselben

Referenz- und aktuellen Daten durchgeführt werden, um ein einheitliches Ziel – das Erkennen derselben Drift– zu gewährleisten.

Die Referenz- und aktuellen Daten basierten auf dem California-Housing-Datensatz, wobei die Variable „MedInc“ als univariate Testfeature verwendet wurde. Die Daten wurden in zwei Teile aufgeteilt: erste 10.000 Einträge als Referenzdaten und letzte 10.000 Einträge als aktuelle Daten.

Künstlicher Drift wurde in die aktuellen Daten mit einer selbst definierten Funktion eingeführt, um verschiedene Drift-Szenarien zu simulieren. Dabei wurde die Verteilung der Daten für jede Variable durch einen festen Wert verschoben, der wie folgt definiert ist

$$\big(\alpha + \text{mean}(feature)\big) * \text{perc} \tag{16}$$

Formel 16: Wert zur Erstellung der künstlichen Drift

Quelle: Eigene Darstellung

Durch die Verwendung des Mittelwerts jeder Variable wurde sichergestellt, dass die Verschiebung relativ zum Wertebereich der Merkmale erfolgte. Zusätzlich wurde ein kleiner „α“-Wert = 0.001 eingeführt, um die Verschiebung auch dann zu erzeugen, wenn der Mittelwert des Merkmals bei 0 liegt.

Der Drift wurde sowohl durch die Verschiebung der Datenverteilung (Drift-Größe) als auch durch die Veränderung eines Anteils der Daten (Drift-Ratio) variiert. Es wurden fünf verschiedene Driftgrößen (0.05, 0.1, 0.3, 0.7, 1.0) und fünf verschiedene Driftanteile (0.01, 0.05, 0.1, 0.2, 0.5) getestet. Die Werte und die Gründe für die Auswahl sind in der folgenden Tabelle zusammengefasst:

Tabelle 2: Auswahl der Driftgrößen

Quelle: Eigene Darstellung

Drift-Größe	Beschreibung
0.05	Kleiner Drift: Erfasst subtile Veränderungen in der Datenverteilung.
0.1	Moderater Drift: Testet mittlere Drifts, die häufig in realen Szenarien auftreten.
0.3	Bedeutender Drift: Stellt große Änderungen in der Verteilung dar, nützlich für die Robustheitsbewertung
0.7	Starker Drift: Simuliert extreme Drifts, um zu testen, ob die Erkennungsmethode weiterhin effektiv bleibt.
1.0	Vollständiger Drift: Stellt sicher, dass die Erkennungs-Pipeline funktioniert, wenn die Referenz- und aktuellen Daten vollständig unterschiedlich sind.

Tabelle 3: Auswahl der Driftanteile

Quelle: Eigene Darstellung

Drift Ratio	Beschreibung
0.01	Sehr kleiner Anteil: Testet die Sensitivität auf Drifts in kleinen Datensätzen, nützlich für die Identifizierung von Ausreißern.
0.05	Kleiner Anteil: Simuliert realistische Szenarien, in denen der Drift nur einen kleinen Teil des Datensatzes betrifft.
0.1	Moderater Anteil: Stellt Fälle dar, in denen der Drift einen signifikanten Teil des Datensatzes betrifft.
0.2	Großer Anteil: Evaluierung der Leistung, wenn ein bedeutender Teil der Daten von der Drift betroffen ist.
0.5	Hälfte des Datensatzes: Testet die Grenzen der Methode bei weit verbreiteten Änderungen im Datensatz.

Es wurden 25 verschiedene Kombinationen der Driftgrößen und -ratios untersucht, um unterschiedliche Drift-Szenarien zu simulieren, wie z. B. subtile, mittlere oder starke Drifts in kleinen oder großen Datensätzen.

Zur Auswertung der Ergebnisse wurden Zeitreihenplots sowie Histogramme der Verteilungen von Referenz- und aktuellen Daten erstellt. Die Ergebnisse der Tests, die die Abweichungen zwischen den Verteilungen der Referenz- und aktuellen Daten erkennen konnten, wurden mit Drift-Scores und einer Kennzeichnung, ob eine Drift erkannt wurde (True/False), zusammengefasst.

Im nächsten Abschnitt werden die Ergebnisse und Diskussionen für diese Experimente dargestellt.

7.1.2 Ergebnisse und Diskussion

Tabelle 4: Ergebnisse der Experimente mit datenverteilungsbasierten Ansätzen

Quelle: Eigene Darstellung

Testbedingungen	Hellinger-Distanz	KL-Divergenz	KS-Test	Wasserstein-Distanz (normed)	T-Test
5.0% segment moved by 1.0%	✗	✗	✓	✓	✓

5.0% segment moved by 5.0%	✗	✗	✓	✓	✓
5.0% segment moved by 10.0%	✗	✗	✓	✓	✓
5.0% segment moved by 20.0%	✗	✗	✓	✓	✓
5.0% segment moved by 50.0%	✓	✗	✓	✓	✓
10.0% segment moved by 1.0%	✓	✗	✓	✓	✓
10.0% segment moved by 5.0%	✓	✗	✓	✓	✓
10.0% segment moved by 10.0%	✓	✗	✓	✓	✓
10.0% segment moved by 20.0%	✓	✗	✓	✓	✓
10.0% segment moved by 50.0%	✓	✗	✓	✓	✓
30.0% segment moved by 1.0%	✓	✗	✓	✓	✓
30.0% segment moved by 5.0%	✓	✗	✓	✓	✓
30.0% segment moved by 10.0%	✓	✗	✓	✓	✓
30.0% segment moved by 20.0%	✓	✓	✓	✓	✓
30.0% segment moved by 50.0%	✓	✓	✓	✓	✓
70.0% segment moved by 1.0%	✓	✓	✓	✓	✓
70.0% segment moved by 5.0%	✓	✓	✓	✓	✓
70.0% segment moved by 10.0%	✓	✓	✓	✓	✓
70.0% segment moved by 20.0%	✓	✓	✓	✓	✓
70.0% segment moved by 50.0%	✓	✓	✓	✓	✓
100% segment moved by 1.0%	✓	✓	✓	✓	✓
100% segment moved by 5.0%	✓	✓	✓	✓	✓
100% segment moved by 10.0%	✓	✓	✓	✓	✓
100% segment moved by 20.0%	✓	✓	✓	✓	✓

100% segment moved by 50.0%	✓	✓	✓	✓	✓

Der detaillierte Vergleich der verschiedenen statistischen Tests zur Erkennung von Drift innerhalb einer Datenverteilung zeigte, dass insbesondere der Kolmogorov-Smirnov-Test, die Wasserstein-Distanz sowie der T-Test zu den empfindlichsten Verfahren zählen. Diese Methoden erwiesen sich als besonders leistungsfähig, da sie in sämtlichen untersuchten Szenarien eine Veränderung unabhängig von der konkreten Größe oder dem spezifischen Verhältnis der Drift zuverlässig identifizieren konnten. Dies verdeutlicht ihre hohe Sensitivität gegenüber Veränderungen in der Datenverteilung und macht sie besonders geeignet für Anwendungen, in denen bereits kleinste Abweichungen eine kritische Rolle spielen.

Die Hellinger-Distanz zeigte ebenfalls eine beachtliche Sensitivität, insbesondere bei signifikanten Driftgrößen. Bereits bei einer Verschiebung von lediglich 5 % der Daten konnte bei einer Driftgröße von 50 % eine erkennbare Abweichung festgestellt werden. Dennoch erwies sich dieses Verfahren als weniger empfindlich gegenüber kleineren strukturellen Veränderungen innerhalb der Datenverteilung, was seine Anwendung in Szenarien mit subtilen Driftphänomenen potenziell einschränkt.

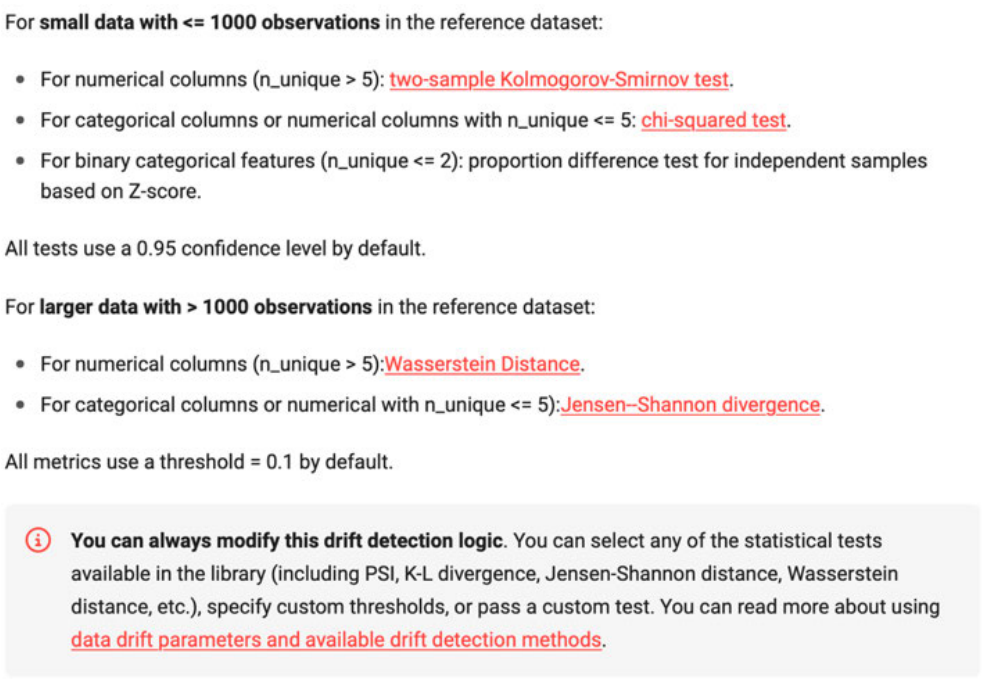
Die KL-Divergenz hingegen wies eine merklich geringere Sensitivität auf und erkannte eine Drift erst ab einer Verschiebung von mindestens 20 % der Daten, wobei die Driftgröße mindestens 30 % betragen musste. Dies könnte darauf zurückzuführen sein, dass die KL-Divergenz primär dazu neigt, große strukturelle Unterschiede zwischen Verteilungen hervorzuheben, anstatt feinkörnige Variationen zu detektieren. Folglich eignet sich dieses Verfahren besonders für Szenarien, in denen drastische, klar abgegrenzte Driftphänomene untersucht werden sollen, während seine Anwendung auf subtile und inkrementelle Veränderungen begrenzt ist.

Die durchgeführten Experimente verdeutlichen weiterhin, dass die Effektivität der jeweiligen statistischen Tests nicht nur von der absoluten Driftgröße, sondern auch von der Segmentgröße, also dem sogenannten Driftratio, signifikant beeinflusst wird. Dies legt nahe, dass die Wahl des geeigneten Testverfahrens stets in Abhängigkeit von den spezifischen Charakteristika der betrachteten Daten sowie dem jeweiligen Anwendungskontext erfolgen sollte.

Zusammenfassend lässt sich festhalten, dass insbesondere der Kolmogorov-Smirnov-Test, die Wasserstein-Distanz sowie der T-Test für Szenarien empfohlen werden können, in denen bereits kleinste Veränderungen in der Datenverteilung eine entscheidende Bedeutung haben. Ihre hohe Sensitivität gegenüber auch geringfügigen Driftphänomenen macht sie zu leistungsstarken Werkzeugen für anspruchsvolle Analyseaufgaben in diesem Bereich.

Für multivariate Daten bietet Evidently vordefinierte Berichte, die im „Forschungswerkstatt 1“-Paper repräsentiert wurden. Evidently wählt automatisch den geeignetsten Test für die gleichzeitige Analyse mehrerer Merkmale aus, lässt jedoch auch den Nutzer die Auswahl durch Parametereinstellungen

anpassen. Abbildung 10 ist ein Auszug aus der Dokumentation der Bibliothek Evidently zu diesem Thema.


For **small data with <= 1000 observations** in the reference dataset:

- For numerical columns (n_unique > 5): two-sample Kolmogorov-Smirnov test.
- For categorical columns or numerical columns with n_unique <= 5: chi-squared test.
- For binary categorical features (n_unique <= 2): proportion difference test for independent samples based on Z-score.

All tests use a 0.95 confidence level by default.

For **larger data with > 1000 observations** in the reference dataset:

- For numerical columns (n_unique > 5):Wasserstein Distance.
- For categorical columns or numerical with n_unique <= 5):Jensen–Shannon divergence.

All metrics use a threshold = 0.1 by default.

You can always modify this drift detection logic. You can select any of the statistical tests available in the library (including PSI, K-L divergence, Jensen-Shannon distance, Wasserstein distance, etc.), specify custom thresholds, or pass a custom test. You can read more about using data drift parameters and available drift detection methods.

Abbildung 10: Auszug aus der Dokumentation der Bibliothek Evidently

Quelle: https://docs.evidentlyai.com/reference/data-drift-algorithm

7.2 Leistungsbasierte Ansätze

7.2.1 Experimenteller Aufbau und Ablauf

Die Untersuchung der leistungsbasierten Ansätze wurde in zwei verschiedene Teil-Experimente unterteilt. Im ersten Experiment wurden Methoden der statistischen Prozesskontrolle und Fenstertechniken getestet, während im zweiten Experiment Ensemble-Lernmethoden verwendet wurden. Beide Experimente basierten auf synthetischen Datenströmen, die verschiedene Arten von Drift darstellten.

Experiment mit statistischer Prozesskontrolle und Fenstertechniken

Im ersten Experiment wurden sieben verschiedene Drift-Erkennungsverfahren in alphabetischer Reihenfolge getestet: ADWIN (Fenstertechniken), DDM (Statistische Prozesskontrolle), EDDM (Statistische Prozesskontrolle), HDDM_A (Statistische Prozesskontrolle), HDDM_W (Statistische Prozesskontrolle), KSWIN (Fenstertechniken) und PageHinkley (Statistische Prozesskontrolle), jeweils in Kombination mit einem Naive Bayes-Klassifikator.

Als Datensätze wurden vier verschiedene Generatoren verwendet, die unterschiedliche Driftarten simulieren. Diese Auswahl wurde durch die Experimente in Gonçalves et al. (2014) inspiriert.

1. Sine: „Abrupter Drift mit einer sanften, periodischen Änderung“, wobei die Daten durch wellenartige Muster mit plötzlichen Veränderungen geprägt sind (Gonçalves et al., 2014).
2. Hyperplane: „Gradualer Drift mit einer einfachen linearen Entscheidungsgrenze“, bei dem die Trennlinie der Klassen kontinuierlich verschoben wird (Gonçalves et al., 2014).
3. MIXED: „Gradualer Drift mit Rauschen und mehreren Merkmalen“, wobei komplexe, mehrdimensionale Verschiebungen und Störungen in den Daten auftreten (Gonçalves et al., 2014).
4. RandomRBF: Gradualer Drift mit nichtlinearen Übergängen, bei dem die Daten mit nichtlinearen Veränderungen konfrontiert werden.

Für jedes Experiment wurden 20.000 Samples simuliert, die in 200er-Chunks verarbeitet wurden. Der Naive Bayes-Klassifikator wurde auf jedem Chunk trainiert, während die Drift-Erkennungsverfahren die Driftphasen des Datenstroms überwachten. Dabei wurden die Genauigkeit, die Anzahl der Fehlalarme, die Anzahl der verpassten Drifts sowie die Rechenzeit für jedes Verfahren erfasst.

Experiment mit Ensemble-Methoden

Im zweiten Experiment kamen drei Ensemble-Methoden zum Einsatz: der Accuracy Weighted Ensemble, der Dynamic Weighted Majority und der Learn++.NSE, welche im Abschnitt 4.3 dieser Arbeit theoretisch erläutert wurden.

Diese Klassifikatoren wurden auf 3 der 4 Datensätze aus dem ersten Experiment angewendet: Sine, Hyperplane und RandomRBF, wobei MIXED aufgrund seiner mehr als zwei Merkmale ausgeschlossen wurde, da die Klassifikatoren mit Datensätzen dieser Art nicht kompatibel sind.

Für die Evaluierung dieses zweiten Experiments wurde der Prequential Evaluator verwendet, der die Klassifikatoren anhand von Genauigkeit, Präzision, Recall und F1-Score bewertet. Der besondere Aspekt des Prequential Evaluators ist, dass die Evaluierung sowohl während des Trainings als auch während der Testphase erfolgt. Dieser Ansatz wird daher als „Prequential Learning“ bezeichnet, da die Bewertung sowohl vor (pre) als auch während (sequential) der Verarbeitung der Daten durchgeführt wird.

7.2.2 Ergebnisse und Diskussion

Experiment mit statistischer Prozesskontrolle und Fenstertechniken

Im ersten Experiment wurden die Ergebnisse für die verschiedenen Detektoren auf vier Datensätzen mit abrupten und graduellen Modelldrift ausgewertet.

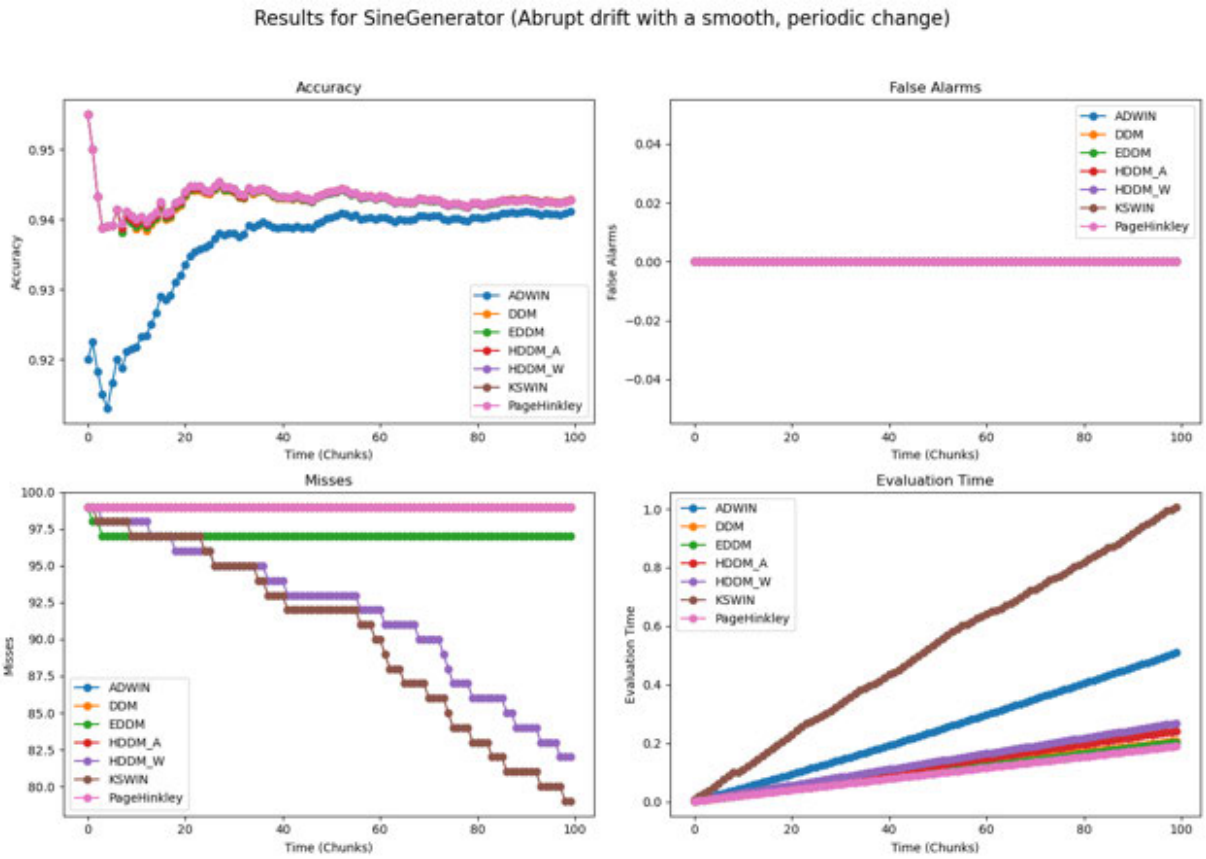


Abbildung 11: Ergebnisse der Experimente mit leistungsbasierten Ansätzen auf Sine

Quelle: Eigene Darstellung

Für den Sine-Datensatz, der durch abrupte, aber glatte und periodische Änderungen gekennzeichnet ist, erreichten alle Detektoren eine Genauigkeit von etwa 0,94. PageHinkley erzielte zu Beginn die höchste Präzision, zeigte jedoch im weiteren Verlauf eine abnehmende Genauigkeit.

Keiner der getesteten Detektoren generierte Fehlalarme, was ihre Widerstandsfähigkeit gegenüber Rauschen unterstreicht.

Hinsichtlich der verpassten Erkennungen lieferten HDDM_W und KSWIN die besten Ergebnisse, während PageHinkley und EDDM die höchste Anzahl nicht erkannter Drifts aufwiesen.

In Bezug auf die Laufzeitanalyse erwies sich KSWIN als der langsamste Detektor, gefolgt von ADWIN, dessen langsamere Anpassung an Drifts durch eine verbesserte Genauigkeit ausgeglichen wurde. Die übrigen Detektoren zeigten vergleichbare Rechenaufwände, ohne signifikante Unterschiede.

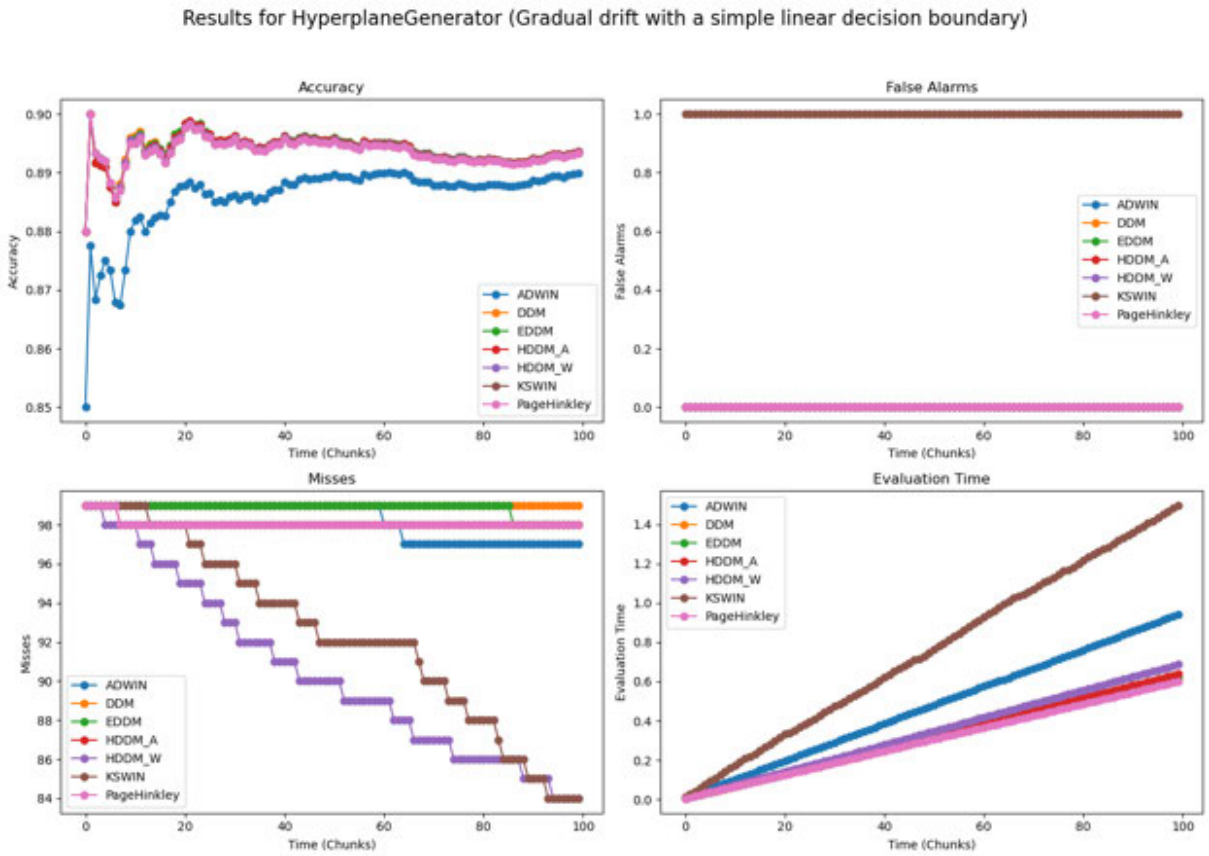


Abbildung 12: Ergebnisse der Experimente mit leistungsbasierten Ansätzen auf Hyperlane

Quelle: Eigene Darstellung

Beim Hyperplane, bei dem es sich um eine allmähliche Drift mit einer linearen Entscheidungsgrenze handelt, lagen alle Detektoren, mit Ausnahme von ADWIN, bei einer Genauigkeit von ungefähr 0.895. ADWIN hingegen erzielte zu Beginn eine Genauigkeit von 0.85 und konnte sich am Ende auf 0.885 steigern, was es zu dem Detektor mit der schlechtesten Performance machte. KSWIN hingegen erzeugte Fehlalarme, während die anderen Detektoren keine Fehlalarme registrierten.

In Bezug auf verpasste Erkennungen zeigten sich HDDM_W und KSWIN als die am besten adaptierenden Detektoren. Die anderen Detektoren passten sich deutlich langsamer an und zeigten teilweise nur sehr geringe Anpassungen. Die Auswertungszeit war ähnlich wie bei Sine, wobei KSWIN und ADWIN die langsamsten Detektoren waren. Die anderen Detektoren zeigten eine vergleichbare Geschwindigkeit. Insgesamt benötigten jedoch alle Detektoren bis zu 50 % mehr Zeit im Vergleich zu Sine.

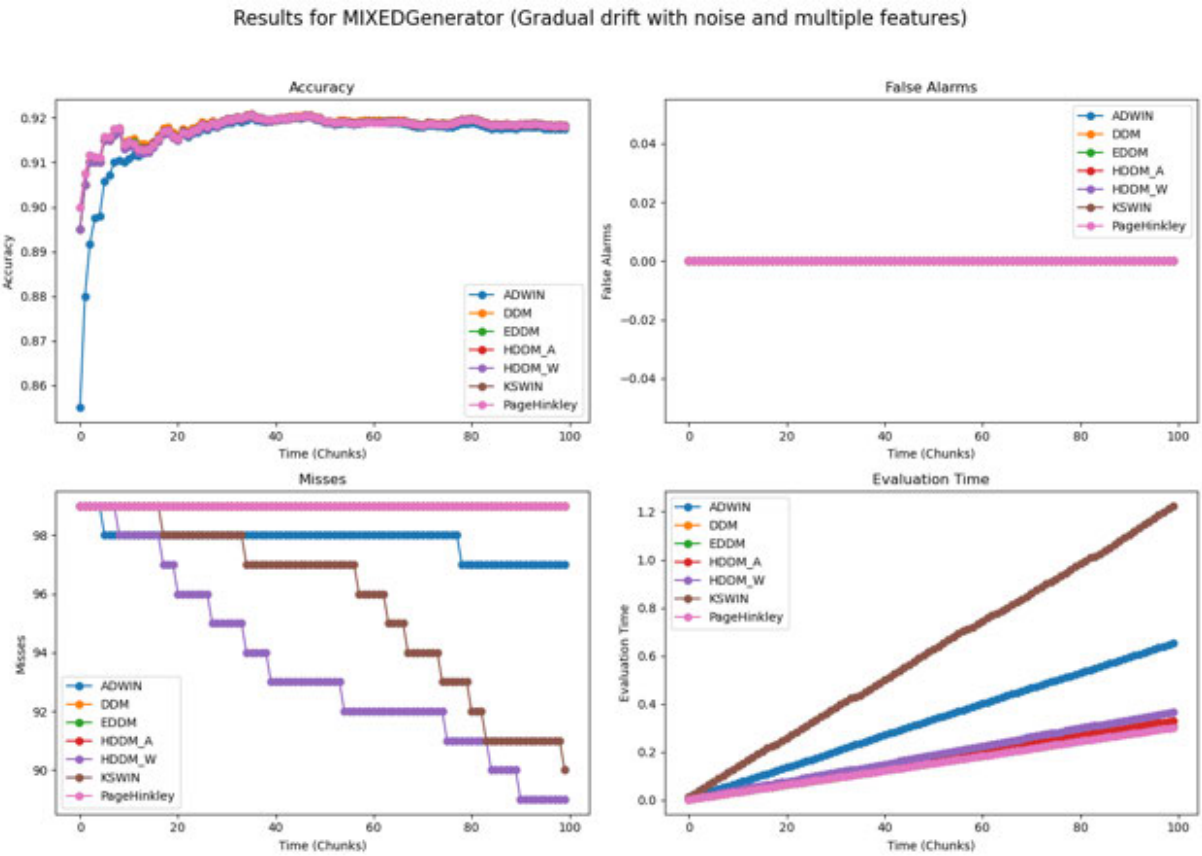


Abbildung 13: Ergebnisse der Experimente mit leistungsbasierten Ansätzen auf MIXED

Quelle: Eigene Darstellung

Beim MIXED-Szenario, das einen graduellen Drift mit Rauschen und mehreren Merkmalen beinhaltet, erzielten alle Detektoren Ergebnisse, die denen beim Sine-Test sehr ähnlich waren, mit einer Genauigkeit von etwa 0,92. Auch hier traten keinerlei Fehlalarme auf, und sowohl HDDM_W als auch KSWIN zeigten eine deutlich bessere Anpassungsfähigkeit im Vergleich zu den anderen Detektoren.

Wie bereits im Sine- und Hyperplane-Szenario benötigten KSWIN und ADWIN auch hier mehr Zeit für die Auswertung als die anderen Detektoren, während diese eine vergleichbare Geschwindigkeit aufwiesen.

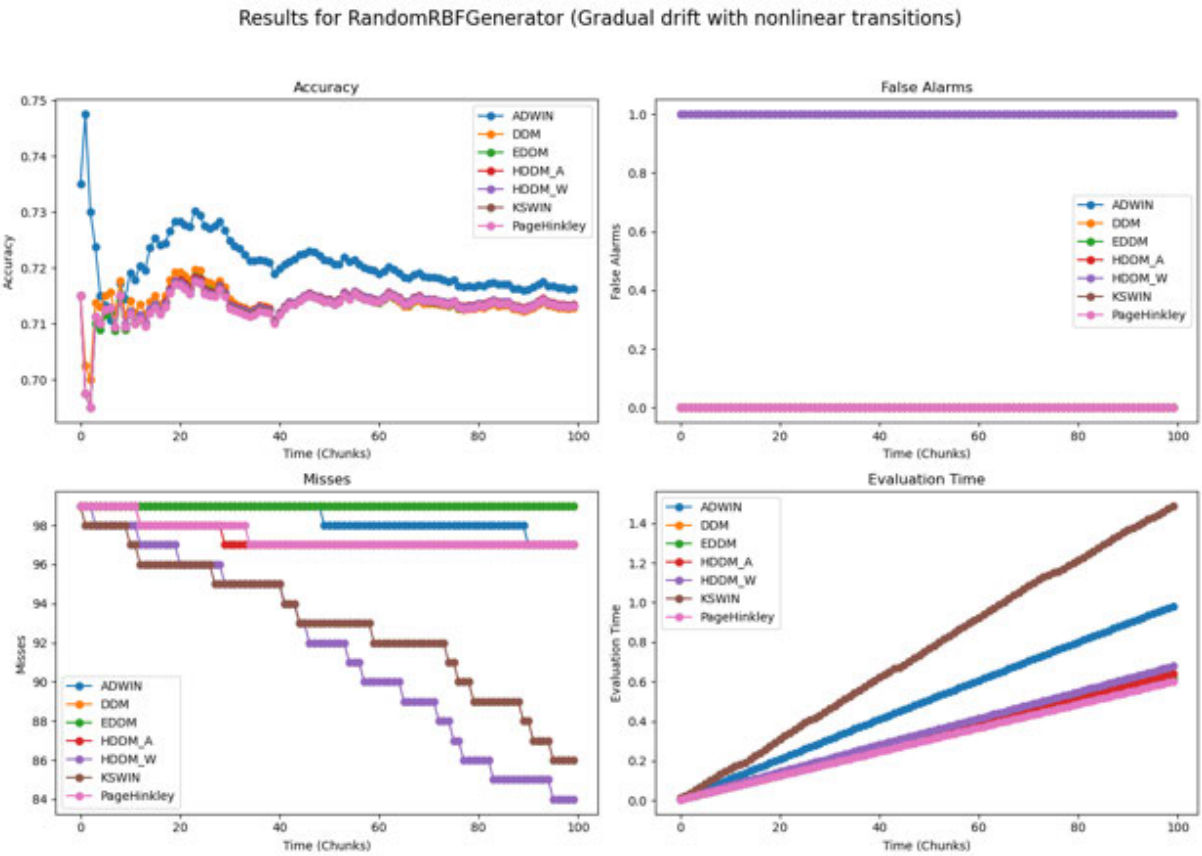


Abbildung 14: Ergebnisse der Experimente mit leistungsbasierten Ansätzen auf RandomRBF

Quelle: Eigene Darstellung

Beim RandomRBF-Szenario, das einen graduellen Drift mit nichtlinearen Übergängen umfasst, war die Leistung aller Detektoren deutlich schlechter als in den vorherigen Szenarien, mit einer Genauigkeit, die zwischen 0,69 und 0,75 lag. Alle Detektoren hatten Schwierigkeiten, sich an die nichtlineare Drift anzupassen, was sich negativ auf ihre Performance auswirkte. KSWIN schnitt dabei jedoch etwas besser ab als die anderen Detektoren, während HDDM_W aufgrund der Fehlalarme auch in diesem Fall weniger effektiv war und Schwierigkeiten hatte, die Drift zuverlässig zu erkennen.

Zusammenfassend lässt sich sagen, dass KSWIN und HDDM_W im Vergleich zu anderen Detektoren durch eine hohe Genauigkeit und eine geringe Anzahl verpasster Drifts überzeugen. Dennoch gibt es klare Unterschiede in der Verarbeitungszeit, die je nach Anwendungskontext berücksichtigt werden müssen:

- KSWIN erzielt in den meisten Testszenarien die höchste Genauigkeit, insbesondere bei abrupten und graduellen Drifts. Es erkennt auch subtile Veränderungen zuverlässig, was es zu einer ausgezeichneten Wahl für Szenarien macht, in denen Präzision im Vordergrund steht. Allerdings benötigt KSWIN mehr Rechenzeit, was es für Anwendungen mit höheren Leistungsanforderungen geeignet macht, bei denen der Gewinn an Präzision die längere Verarbeitungszeit rechtfertigt.

- HDDM_W liefert ebenfalls eine hohe Genauigkeit, liegt jedoch etwas hinter KSWIN. Dafür punktet es mit kürzerer Verarbeitungszeit und eignet sich somit besser für Echtzeitsysteme, bei denen Geschwindigkeit wichtiger ist und kleinere Genauigkeitseinbußen toleriert werden können. Bei subtilen Drifts kann HDDM_W jedoch weniger präzise sein, was in solchen Fällen berücksichtigt werden sollte.

Beide Detektoren haben also ihre Stärken, die je nach spezifischen Anforderungen der Anwendung abgewogen werden sollten.

Experiment mit Ensemble-Methoden

Im zweiten Experiment wurde die Leistung der Accuracy Weighted Ensemble (AWE), der Dynamic Weighted Majority Classifier (DWM) und der Learn++.NSE auf drei verschiedenen Datensätzen mit unterschiedlichen Driftarten untersucht.

Tabelle 5: Ergebnisse der Experimente mit Ensemble-Methoden (leistungsbasierte Ansätze)

Quelle: Eigene Darstellung

Datensatz	Klassifikator	Verarbeitungszeit (s)	Genauigkeit	Präzision	Recall	F1-Score
Sine	AWE	31.38	0.9730	0.9817	0.9592	0.9703
	DWM	15.60	0.9391	0.9503	0.9156	0.9326
	Learn++.NSE	125.78	0.9493	0.9446	0.9454	0.9450
Hyperplane	AWE	32.53	0.8781	0.8794	0.8773	0.8783
	DWM	16.29	0.8618	0.8603	0.8647	0.8625
	Learn++.NSE	90.08	0.8100	0.8103	0.8110	0.8106
RandomRBF	AWE	34.46	0.5496	0.5186	0.4945	0.5063
	DWM	18.27	0.5446	0.5130	0.4865	0.4994
	Learn++.NSE	96.87	0.5436	0.5113	0.5133	0.5123

Beim Sine-Szenario, das einen abrupten Drift mit glatten, periodischen Änderungen beinhaltet, erzielte der AWE exzellente Ergebnisse in den Bereichen Genauigkeit, Präzision, Recall und F1-Score. Der DWM zeigte solide, aber etwas schwächere Leistungen in denselben Metriken. Der Learn++.NSE-Klassifikator lieferte ebenfalls starke Ergebnisse, wobei die Metriken in Bezug auf Genauigkeit, Präzision, Recall und F1-Score leicht über denen des DWM lagen, jedoch hinter denen des AWE zurückblieben.

Im Hyperplane-Szenario, das einen graduellen Drift mit einer einfachen linearen Entscheidungsgrenze beschreibt, erzielte der DWM ähnliche Ergebnisse wie der AWE, mit stabilen Metriken. Der Learn++.NSE-Klassifikator erreichte jedoch etwas niedrigere Werte in Bezug auf die Metriken, sowohl im Vergleich zum DWM als auch zum AWE.

Beim RandomRBF, das einen graduellen Drift mit nichtlinearen Übergängen aufweist, hatten alle drei Klassifikatoren Schwierigkeiten, was die Herausforderungen bei der Erkennung von nichtlinearen Drifts verdeutlicht. Der Learn++.NSE-Klassifikator zeigte jedoch eine leicht bessere Leistung, besonders was Recall und F1-Score betrifft, verglichen mit den anderen beiden.

In Bezug auf den Zeitaufwand benötigte der DWM für alle getesteten Datensätze lediglich die Hälfte der Zeit im Vergleich zum AWE, was ihn deutlich schneller machte. Der Learn++.NSE-Klassifikator wiederum benötigte deutlich mehr Rechenzeit als sowohl der DWM als auch der AWE, um ähnliche Ergebnisse zu erzielen.

Zusammenfassend lässt sich sagen, dass der AWE in den meisten Fällen solide bis gute Ergebnisse lieferte, besonders bei abrupten Drifts, und dass seine Verarbeitungszeit zwischen dem DWM und Learn++.NSE lag. Learn++.NSE zeigte zwar eine höhere Leistung in bestimmten Metriken, benötigte aber auch viele Ressourcen und lieferte nur mittelmäßige Ergebnisse. Der DWM wiederum war am effizientesten, was die Verarbeitungszeit betraf, zeigte jedoch auch die schwächeren Leistungen in 2 von 3 getesteten Datensätzen. Alle drei Klassifikatoren hatten mit nichtlinearen Drifts zu kämpfen, was zeigt, dass die Wahl des besten Klassifikators stark von der Art des Drifts im jeweiligen Datensatz abhängt.

7.3 Fazit der Experimentellen Untersuchung

Die experimentelle Analyse der datenverteilungsbasierten und leistungsbasierten Ansätze zur Drifterkennung hat wertvolle Erkenntnisse über die Wirksamkeit dieser Methoden unter verschiedenen Bedingungen geliefert:

1. Datenverteilungsbasierte Ansätze, insbesondere der Kolmogorov-Smirnov-Test, die Wasserstein-Distanz und der T-Test, zeigten eine hohe Sensitivität gegenüber Veränderungen in den Datenverteilungen. Im Gegensatz dazu erwiesen sich die Hellinger-Distanz und die Kullback-Leibler-Divergenz in bestimmten Szenarien als weniger effektiv.

→ Diese Ergebnisse betonen die Bedeutung der Auswahl des richtigen Tests, je nach Größe des aufgetretenen Drifts. Besonders bei größeren Veränderungen in der Verteilung bieten einige Tests deutlich bessere Erkennungsfähigkeiten.

2. Leistungsbasierte Ansätze, die Fenstertechniken, statistische Prozesskontrolle und Ensemble-Lernmethoden umfassen, zeigten unterschiedliche Stärken:

a. Im Teil-Experiment mit Fenstertechniken und statistischer Prozesskontrolle erwiesen sich KSWIN und HDDM_W als die leistungsfähigsten Detektoren. Allerdings war KSWIN im Vergleich zu anderen Verfahren besonders langsam und leistungsschwach, was die Praktikabilität in realen Anwendungen einschränken kann.
b. Im Teil-Experiment mit Ensemble-Lernmethoden erzielte der Accuracy Weighted Ensemble (AWE) solide Ergebnisse, besonders bei abrupten Drifts, während der Dynamic Weighted Majority (DWM) die kürzeste Verarbeitungszeit benötigte, aber schwächere Leistungen zeigte. Der Learn++.NSE-Klassifikator benötigte mehr Zeit und lieferte insgesamt nur mittelmäßige Ergebnisse, was auf die Ressourcenkosten und die begrenzte Effektivität hinweist.
c. Alle getesteten leistungsbasierten Ansätze hatten jedoch Schwierigkeiten, mit nichtlinearen Drifts umzugehen, was die Komplexität dieser Art von Drift und die Notwendigkeit für spezialisierte Methoden zur Handhabung solcher Herausforderungen verdeutlicht.

→ Diese Ergebnisse verdeutlichen, dass die Wahl des geeigneten Ansatzes stark von der Art des Drifts abhängt. Während einige Methoden bei bestimmten Driftarten gut abschneiden, erfordert der Umgang mit nichtlinearen Drifts möglicherweise eine andere Herangehensweise.

Insgesamt liefern die Experimente wertvolle Erkenntnisse zur Auswahl geeigneter Verfahren zur Drifterkennung. Es ist entscheidend, einen Ansatz zu wählen, der sowohl Genauigkeit als auch Verarbeitungseffizienz berücksichtigt und in der Lage ist, verschiedene Driftgrößen und -arten effektiv zu identifizieren. Die Wahl des besten Ansatzes muss also an die spezifischen Anforderungen der Anwendung und die Art des auftretenden Drifts angepasst werden.

8 Zusammenfassung und Ausblick

In dieser Arbeit wurden verschiedene Ansätze zur Erkennung von Modelldrift sowie deren jeweilige Stärken und Schwächen umfassend untersucht. Dabei wurde ein besonderer Schwerpunkt auf datenverteilungsbasierte, leistungsbasierte sowie multihypothesen- und kontextbasierte Methoden gelegt, um deren Anwendbarkeit in unterschiedlichen Szenarien fundiert zu bewerten und praxisnahe Schlussfolgerungen zu ziehen.

Datenverteilungsbasierte Ansätze zeichnen sich durch eine hohe Genauigkeit bei der Identifikation von Driftzeitpunkten und -orten aus, indem sie systematisch Veränderungen in der Verteilung der

Eingangsdaten analysieren. Dadurch lassen sich potenzielle Drifts mit hoher Sensitivität detektieren. Dennoch können diese Methoden zu einer erhöhten Rate von Fehlalarmen führen, da nicht jede beobachtete Abweichung in den zugrunde liegenden Daten zwangsläufig eine relevante Modelldrift darstellt. Ein weiterer kritischer Punkt ist die Wahl geeigneter Distanzmaße und Teststatistiken, um signifikante Veränderungen zu identifizieren, ohne dass harmlose Schwankungen zu irreführenden Alarmauslösungen führen.

Im Gegensatz dazu bieten leistungsbasierte Ansätze eine robustere Möglichkeit zur Drift-Erkennung, indem sie sich direkt an der tatsächlichen Modellleistung orientieren. Diese Methoden bewerten Veränderungen auf Basis von Leistungsmetriken wie Genauigkeit, Präzision, Recall oder F1-Score. Sie sind besonders vorteilhaft, da sie nur dann Alarm schlagen, wenn sich eine reale Verschlechterung des Modells abzeichnet, wodurch unnötige Fehlalarme reduziert werden. Jedoch ist der Einsatz dieser Methoden mit bestimmten Herausforderungen verbunden: Einerseits erfordert die Evaluation eine kontinuierliche Verfügbarkeit wahrer Labels, was in vielen realen Anwendungsfällen, insbesondere bei Echtzeit- oder unüberwachten Lernsystemen, problematisch sein kann. Andererseits ist eine sorgfältige Auswahl geeigneter Metriken notwendig, um Drifts in verschiedenen Kontexten adäquat zu erfassen.

Multihypothesen-Ansätze bieten vielversprechende Methoden zur Erkennung von Modelldrift, insbesondere in komplexen und dynamischen Szenarien, in denen einfache Heuristiken nicht ausreichen. Parallele Ansätze wie der IV-Jac-Algorithmus zeichnen sich durch ihre modulare Struktur und hohe Präzision aus. Sie ermöglichen eine differenzierte Betrachtung von Hypothesen über potenzielle Driftmuster und können dadurch eine feinere Analyse der Driftquellen und -ursachen leisten. Allerdings sind sie oft rechenintensiv und setzen eine optimale Wahl der Schwellenwerte voraus, um eine Balance zwischen Sensitivität und Robustheit zu gewährleisten. Hierarchische Ansätze wie der HLFR-Algorithmus stellen eine weitere Möglichkeit dar, um Modelldrift effizient zu erkennen und zu validieren. Diese Methoden bauen auf einer gestaffelten Analyse auf, bei der zunächst grobe Driftindikatoren verwendet werden, bevor detailliertere Tests durchgeführt werden. Dadurch kann die Berechnungslast reduziert werden, während gleichzeitig eine hohe Genauigkeit gewahrt bleibt. Dennoch bringen solche Verfahren erhöhten Bedarf an Parameterabstimmung mit sich und sind oft mit höheren Latenzzeiten verbunden. Beide Methoden kombinieren Flexibilität und Robustheit, setzen jedoch einen erheblichen Ressourcenaufwand voraus, insbesondere im Hinblick auf Speicher- und Rechenkapazitäten.

Kontextbasierte Ansätze wie die eSNN-DD-Methode setzen an der Analyse des Modells selbst sowie seiner Reaktionen auf neue Daten an, um frühzeitig potenzielle Drifts zu erkennen. Diese Methode bietet den Vorteil, dass sie eine proaktive Drift-Erkennung ermöglicht und somit das Risiko signifikanter Leistungseinbußen minimiert. Indem sie Muster in der Modellaktivierung und Entscheidungsfindung untersuchen, können sie auch subtilere Drifts erfassen, die durch klassische Methoden womöglich übersehen würden. Allerdings sind kontextbasierte Verfahren oft durch ihre komplexe Modellstruktur

und den erhöhten Rechenaufwand eingeschränkt, da sie detaillierte Analysen von Aktivierungsmustern und Modellinternas erfordern. Zudem sind sie in vielen Fällen stark von der spezifischen Architektur des Modells abhängig, was ihre Generalisierbarkeit einschränken kann.

Die durchgeführten Experimente verdeutlichen, dass die Wahl des geeigneten Erkennungsansatzes maßgeblich von der Art und dem Ausmaß der Drift, den spezifischen Anforderungen an die Effizienz sowie den zugrunde liegenden Daten abhängt. Kein einzelner Ansatz erweist sich als universell optimal, was das sogenannte „No Free Lunch"-Theorem (Ho & Pepyne, 2002) im Kontext der Modelldrift-Erkennung erneut unterstreicht. Die Ergebnisse dieser Arbeit legen nahe, dass hybride Methoden, welche die Stärken mehrerer Ansätze kombinieren, eine vielversprechende Richtung für weiterführende Forschung darstellen könnten.

Zukünftige Forschungsarbeiten sollten daher verstärkt darauf abzielen, die Entwicklung und Erweiterung multihypothesen- und kontextbasierter Ansätze voranzutreiben, insbesondere im Hinblick auf nicht-lineare und hochdimensionale Driftszenarien. Darüber hinaus könnte die Integration verschiedener Strategien, etwa durch hybride Ansätze, sowohl die Genauigkeit als auch die Effizienz der Drift-Erkennung erheblich verbessern. Durch die geschickte Kombination datenverteilungsbasierter, leistungsbasierter und multihypothesen-Methoden könnte es möglich sein, eine widerstandsfähigere und anpassungsfähigere Drift-Erkennung zu realisieren. Solche kombinierten Ansätze könnten nicht nur theoretische Erkenntnisse weiter vertiefen, sondern auch dazu beitragen, die Lücke zwischen Forschung und praktischen Anwendungen im Bereich der Drift-Erkennung erfolgreich zu schließen. Insbesondere im industriellen Umfeld, wo Modelldrift oft schwerwiegende Konsequenzen nach sich ziehen kann, wären solche innovativen Strategien von großer Relevanz.

Literaturverzeichnis

Baena-Garcıa, M., del Campo-Ávila, J., Fidalgo, R., Bifet, A., Gavalda, R., & Morales-Bueno, R. (2006, September). Early drift detection method. In Fourth international workshop on knowledge discovery from data streams (Vol. 6, pp. 77-86).

Basseville, M. (1993). Detection of Abrupt Changes: Theory and Application. Prentice-Hall google schola, 2, 3-11.

Bayram, F., Ahmed, B. S. & Kassler, A. (2022). From concept drift to model degradation: An overview on performance-aware drift detectors. *Knowledge-Based Systems*, *245*, 108632. https://doi.org/10.1016/j.knosys.2022.108632

Bifet, A., & Gavalda, R. (2007, April). Learning from time-changing data with adaptive windowing. In Proceedings of the 2007 SIAM international conference on data mining (pp. 443-448). Society for Industrial and Applied Mathematics.

Buhrmester, V., Münch, D., & Arens, M. (2021). Analysis of explainers of black box deep neural networks for computer vision: A survey. Machine Learning and Knowledge Extraction, 3(4), 966-989.

Frias-Blanco, I., del Campo-Ávila, J., Ramos-Jimenez, G., Morales-Bueno, R., Ortiz-Diaz, A., & Caballero-Mota, Y. (2014). Online and non-parametric drift detection methods based on Hoeffding's bounds. IEEE Transactions on Knowledge and Data Engineering, 27(3), 810-823.

Gama, J., Medas, P., Castillo, G. & Rodrigues, P. P. (2004). Learning with Drift Detection. In Lecture notes in computer science (S. 286–295). https://doi.org/10.1007/978-3-540-28645-5_29

Goldenberg, I. & Webb, G. I. (2018). Survey of distance measures for quantifying concept drift and shift in numeric data. *Knowledge And Information Systems*, *60*(2), 591–615. https://doi.org/10.1007/s10115-018-1257-z

Gonçalves, P. M., De Carvalho Santos, S. G., Barros, R. S. & Vieira, D. C. (2014). A comparative study on concept drift detectors. *Expert Systems with Applications*, *41*(18), 8144–8156. https://doi.org/10.1016/j.eswa.2014.07.019

Ho, Y. C., & Pepyne, D. L. (2002). Simple explanation of the no-free-lunch theorem and its implications. Journal of optimization theory and applications, 115, 549-570.

Hoeffding, W. (1963). Probability Inequalities for Sums of Bounded Random Variables. Journal Of The American Statistical Association, 58(301), 13. https://doi.org/10.2307/2282952

Huang, D. T. J., Koh, Y. S., Dobbie, G., & Pears, R. (2014, December). Detecting volatility shift in data streams. In 2014 IEEE International Conference on Data Mining (pp. 863-868). IEEE.

Kasabov, N. K. (2007). Evolving connectionist systems: the knowledge engineering approach. Springer Science & Business Media.

Klinkenberg, R. (2004). Learning drifting concepts: Example selection vs. example weighting. Intelligent Data Analysis, 8(3), 281–300. https://doi.org/10.3233/ida-2004-8305

Kolter, J. Z., & Maloof, M. A. (2007). Dynamic weighted majority: An ensemble method for drifting concepts. The Journal of Machine Learning Research, 8, 2755-2790.

Lobo, J. L., Del Ser, J., Laña, I., Bilbao, M. N. & Kasabov, N. (2018). Drift Detection over Non-stationary Data Streams Using Evolving Spiking Neural Networks. In *Studies in computational intelligence* (S. 82–94). https://doi.org/10.1007/978-3-319-99626-4_8

Lu, J., Liu, A., Dong, F., Gu, F., Gama, J. & Zhang, G. (2018). Learning under Concept Drift: A Review. *IEEE Transactions on Knowledge and Data Engineering*, 1. https://doi.org/10.1109/tkde.2018.2876857

Mouss, H., Mouss, D., Mouss, N., & Sefouhi, L. (2004, July). Test of page-hinckley, an approach for fault detection in an agro-alimentary production system. In 2004 5th Asian control conference (IEEE Cat. No. 04EX904) (Vol. 2, pp. 815-818). IEEE.

Polikar, R., Upda, L., Upda, S. & Honavar, V. (2001). Learn++: an incremental learning algorithm for supervised neural networks. IEEE Transactions On Systems Man And Cybernetics Part C (Applications And Reviews), 31(4), 497–508. https://doi.org/10.1109/5326.983933

Rubner, Y., Tomasi, C. & Guibas, L. J. (2000). The Earth Mover's Distance as a Metric for Image Retrieval. International Journal Of Computer Vision, 40(2), 99–121. https://doi.org/10.1023/a:1026543900054

Schliebs, S., & Kasabov, N. (2013). Evolving spiking neural network—a survey. Evolving Systems, 4, 87-98.

Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H., & Zhao, B. Y. (2019, May). Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In 2019 IEEE symposium on security and privacy (SP) (pp. 707-723). IEEE.

Wang, H., Fan, W., Yu, P. S., & Han, J. (2003, August). Mining concept-drifting data streams using ensemble classifiers. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 226-235).

Woźniak, M., Ksieniewicz, P., Cyganek, B., & Walkowiak, K. (2016). Ensembles of heterogeneous concept drift detectors-experimental study. In Computer Information Systems and Industrial Management: 15th IFIP TC8 International Conference, CISIM 2016, Vilnius, Lithuania, September 14-16, 2016, Proceedings 15 (pp. 538-549). Springer International Publishing.

Yu, S. & Abraham, Z. (2017). Concept Drift Detection with Hierarchical Hypothesis Testing. In *Society for Industrial and Applied Mathematics eBooks* (S. 768–776). https://doi.org/10.1137/1.9781611974973.86

Zhang, Y., Chu, G., Li, P., Hu, X. & Wu, X. (2017). Three-layer concept drifting detection in text data streams. *Neurocomputing*, *260*, 393–403. https://doi.org/10.1016/j.neucom.2017.04.047

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

_____	_____	
Ort	Datum	Unterschrift im Original