

**MASTERARBEIT**

# **Ein Überblick über Algorithmen zur Erkennung von Drift**

---

vorgelegt am 31. März 2025  
Nhut-Hoa Huynh

Erstprüferin: Prof. Dr. Marina Tropmann-Frick

---

**HOCHSCHULE FÜR ANGEWANDTE  
WISSENSCHAFTEN HAMBURG**

Department Informatik  
Berliner Tor 7  
20099 Hamburg

## **Zusammenfassung**

Diese Arbeit analysiert verschiedene Ansätze zur Modelldrift-Erkennung, darunter datenverteilungsbasierte, leistungsbasierte, mehrfachhypothesebasierte und kontextbasierte Methoden. Die Algorithmen werden theoretisch erklärt und in Experimenten hinsichtlich ihrer Genauigkeit und Effizienz verglichen. Die Ergebnisse zeigen, dass keine universelle Lösung existiert; die Wahl der Methode hängt von Driftarten, -größen und Anwendungsanforderungen ab. Die Arbeit betont die Notwendigkeit, die Ansätze an verschiedene Szenarien anzupassen, um sowohl die Erkennungsgenauigkeit als auch die Verarbeitungseffizienz zu optimieren.

## **Abstract**

This thesis analyzes different approaches for model drift detection, including data distribution-based, performance-based, multiple hypothesis-based and context-based methods. The algorithms are explained theoretically and compared in experiments with respect to their accuracy and efficiency. The results show that no universal solution exists; the choice of method depends on drift types, sizes and application requirements. The work emphasizes the need to adapt the approaches to different scenarios to optimize both recognition accuracy and processing efficiency.

# Inhaltsverzeichnis

Abbildungsverzeichnis .....	V
Tabellenverzeichnis .....	VI
Formelverzeichnis .....	VII
1 Einleitung .....	8
1.1 Ziele dieser Arbeit .....	8
1.2 Gliederung dieser Arbeit .....	8
2 Grundlagen .....	8
2.1 Definition und Auswirkungen von Modelldriften .....	8
2.2 Arten und Ursachen von Modelldriften .....	9
2.2.1 Arten von Modelldriften .....	9
2.2.2 Ursachen von Modelldriften .....	11
2.3 Methoden zur Erkennung von Modelldriften .....	11
3 Datenverteilungsbasierte Ansätze .....	12
3.1 Univariate Daten .....	12
3.1.1 Maße der Verteilungsdissimilarität für univariate Daten .....	12
3.1.2 Distanzmaße zwischen Beobachtungen für univariate Daten .....	13
3.2 Multivariate Daten .....	14
3.2.1 Maße der Verteilungsdissimilarität für multivariate Daten .....	14
3.2.2 Distanzmaße zwischen Beobachtungen für multivariate Daten .....	15
3.3 Fazit der datenverteilungsbasierten Ansätze .....	15
4 Leistungsbasierte Ansätze .....	16
4.1 Statistische Prozesskontrolle .....	16
4.2 Fenstertechniken .....	17
4.3 Ensemble-Lernen .....	18
4.4 Fazit der leistungsbasierten Ansätze .....	19
5 Mehrfachhypothesen-basierte Ansätze .....	19
5.1 Parallele Mehrfachhypothesen - Driftdetektoren .....	20
5.2 Hierarchische Mehrfachhypothesen– Driftdetektoren .....	21

5.3	Fazit der Mehrfachhypothesen-basierten Ansätze .....	21
6	Kontextbasierte Ansätze .....	21
7	Experimentelle Untersuchung und Ergebnisse.....	22
7.1	Datenverteilungsbasierte Ansätze.....	22
7.1.1	Experimenteller Aufbau und Ablauf .....	22
7.1.2	Ergebnisse und Diskussion.....	24
7.2	Leistungsbasierte Ansätze .....	26
7.2.1	Experimenteller Aufbau und Ablauf .....	26
7.2.2	Ergebnisse und Diskussion.....	27
7.3	Fazit der Experimentellen Untersuchung .....	32
8	Zusammenfassung und Ausblick.....	33
	Literaturverzeichnis.....	34
	Eigenständigkeitserklärung .....	35

## Abbildungsverzeichnis

Abbildung 1: vier Driftmuster.....	10
Abbildung 2: Hierarchie der Drift-Erkennungen .....	11
Abbildung 3: Hierarchie der leistungsbasierten Ansätze zur Drifterkennung .....	16
Abbildung 4: Parallele Mehrfachhypothesen - Driftdetektoren.....	20
Abbildung 5: Hierarchische Mehrfachhypothesen - Driftdetektoren.....	21
Abbildung 6: Auszug aus der Dokumentation der Bibliothek Evidently .....	26
Abbildung 7: Ergebnisse der Experimente mit leistungsbasierten Ansätzen auf Sine .....	28
Abbildung 8: Ergebnisse der Experimente mit leistungsbasierten Ansätzen auf Hyperlane .....	29
Abbildung 9: Ergebnisse der Experimente mit leistungsbasierten Ansätzen auf MIXED .....	30
Abbildung 10: Ergebnisse der Experimente mit leistungsbasierten Ansätzen auf RandomRBF .....	31

## **Tabellenverzeichnis**

Tabelle 1: Auswahl der Driftgrößen.....	23
Tabelle 2: Auswahl der Driftanteile .....	23
Tabelle 3: Ergebnisse der Experimente mit datenverteilungsbasierten Ansätzen .....	24
Tabelle 4: Ergebnisse der Experimente mit Ensemble-Methoden (leistungsbasierte Ansätze).....	32

## Formelverzeichnis

Formel 1: Hellinger-Abstand .....	13
Formel 2: Kullback-Leibler-Divergenz.....	13
Formel 3: T-Test .....	14
Formel 4: T-Ratio.....	14
Formel 5: Standardabweichung des T-Ratios .....	14
Formel 6: Hotelling's $T^2$ .....	15
Formel 7: Kovarianzmatrix des Hotelling's $T^2$ .....	15
Formel 8: Standardabweichung einer Fehlklassifikation zum Zeitpunkt t .....	17
Formel 9: Hoeffding-Schranke der ADWIN.....	17
Formel 10: Wert zur Erstellung der künstlichen Drift .....	23

# **1 Einleitung**

## **1.1 Ziele dieser Arbeit**

Die vorliegende Masterarbeit hat das Ziel, einen Vergleich zwischen vier verschiedenen Ansätzen zur Erkennung von Modell-Drift zu ziehen: den datenverteilungsbasierten, leistungsbasierten, multiple Hypothesen-basierten sowie kontextualisierten Methoden. Die Arbeit untersucht, welcher dieser vier Ansätze in verschiedenen Anwendungen am effektivsten sein könnte, indem sie ihre Anwendbarkeit in unterschiedlichen Szenarien bewertet und einen Vergleich hinsichtlich ihrer Effizienz und Genauigkeit zieht. Das Ziel ist es, eine fundierte Entscheidungshilfe für die Wahl der am besten geeignete Methode zu liefern.

## **1.2 Gliederung dieser Arbeit**

Die Gliederung dieser Arbeit 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.

# **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, etwa jüngere Kunden, 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 1 dargestellt sind:

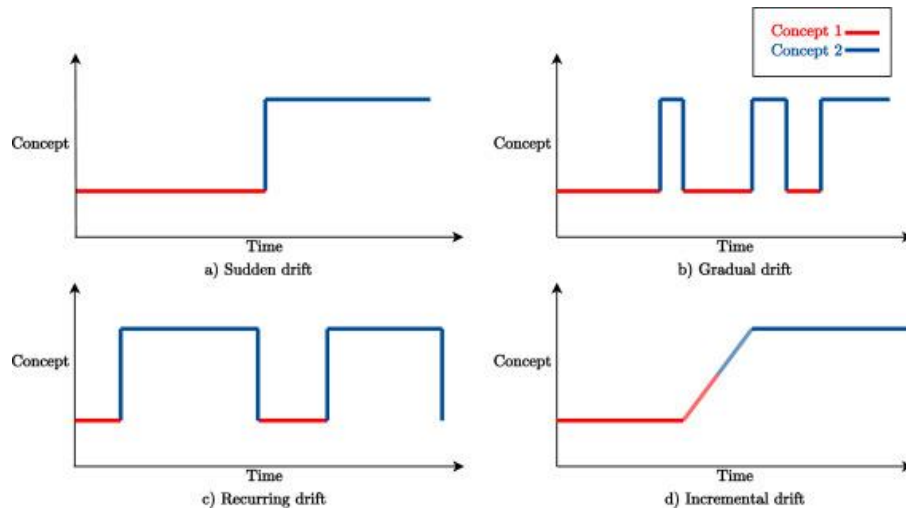


Abbildung 1: vier Driftmuster  
Quelle: Bayram et al. (2022)

### 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, etwa bei der Kovariatenverschiebung und der Verschiebung der A-priori-Wahrscheinlichkeit, können durch saisonale Schwankungen wie steigende Verkaufszahlen während der Weihnachtszeit oder Änderungen im menschlichen Verhalten, wie Konsum- oder Kaufverhalten, ausgelöst werden. Auch technologische Fortschritte, wie die Einführung neuer Datenquellen oder -formate, können die Struktur der Daten verändern und somit zu einer Modelldrift führen. Technologische Neuerungen, wie die Einführung neuer Geräte oder Betriebssysteme, können eine Kontextdrift hervorrufen. Ebenso können strategische Veränderungen, wie etwa eine Neuausrichtung auf eine andere Zielgruppe, den Kontext verändern. Anpassungen der Anforderungen an Produkte oder Dienstleistungen führen ebenfalls zu Kontextdrift, da sie die Rahmenbedingungen für die Nutzung des Modells verändern.

### 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“ (Bayram et al., 2022). 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 2 dargestellt.

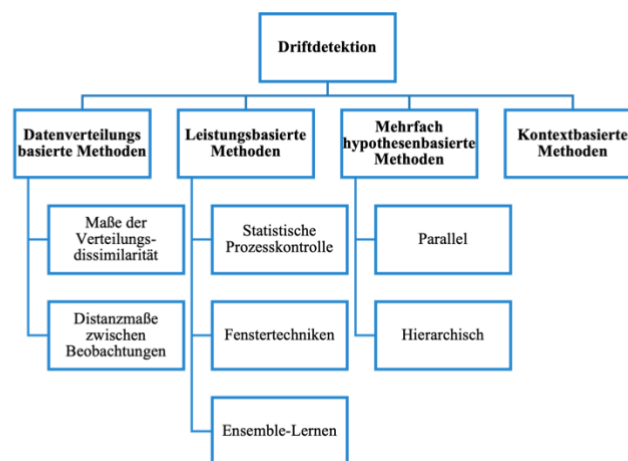


Abbildung 2: Hierarchie der Drift-Erkennungen  
Quelle: Eigene Darstellung

Die erste Kategorie konzentriert sich auf die Überwachung von Dateneigenschaften, um festzustellen, ob sich Verteilungen oder deskriptive Statistiken wie Minimum, Maximum, Median und Mittelwert signifikant verändert haben. 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. Obwohl es datenverteilungs- und leistungsbasierte Ansätze gibt, die am häufigsten eingesetzten Techniken zur Erkennung von Modelldrift darstellen, existieren auch komplexere Ansätze wie multiple Hypothesen-basierte und kontextbasierte Methoden. Nachfolgend werden die vier Methodengruppen näher erläutert.

### **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 und die Kullback-Leibler-Divergenz, 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**

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:

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

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.

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  $D_{KL}(Q || P)$ .

Neben der Hellinger-Distanz und der Kullback-Leibler-Divergenz gibt es weitere bekannte Methoden, um die Differenz oder Ähnlichkeit zwischen Verteilungen zu messen, z. B. die Kolmogorov-Smirnov-Statistik, die die maximale Differenz zwischen den empirischen Verteilungsfunktionen zweier Stichproben misst, oder die Wasserstein-Distanz (auch als „Earth-Mover’s Distance“ bekannt), die misst, wie viel Aufwand erforderlich ist, um eine Verteilung in eine andere zu transformieren.

### 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.

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 ( $\mu_0$ ) abweicht. Der Test berechnet den T-Wert,

der angibt, wie weit der beobachtete Mittelwert  $\bar{X}$  in Einheiten des Standardfehlers von  $\mu_0$  entfernt ist. Die Formel lautet:

$$T = \frac{\bar{X} - \mu_0}{S/\sqrt{n}} \quad (3)$$

Formel 3: T-Test

Quelle: Goldenberg & Webb (2018)

wobei  $S$  die Standardabweichung der Stichprobe und  $n$  die Stichprobengröße. Ist der T-Wert groß genug (über einem vordefinierten kritischen Wert), gibt es Hinweise darauf, dass der Stichprobenmittelwert signifikant von  $\mu_0$  abweicht.

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

$$T = \frac{\bar{X}_1 - \bar{X}_2}{S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \quad (4)$$

Formel 4: T-Ratio

Quelle: Goldenberg & Webb (2018)

wobei  $\bar{X}_1$  und  $\bar{X}_2$  die Mittelwerte der beiden Gruppen sind,  $n_1$  und  $n_2$  ihre Stichprobengrößen und  $S_p$  die gepoolte Standardabweichung, berechnet als

$$S_p = \sqrt{\frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2}} \quad (5)$$

Formel 5: Standardabweichung des T-Ratios

Quelle: Goldenberg & Webb (2018)

wobei  $S_1$  und  $S_2$  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.

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“.

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. Es wird geprüft, ob ein Punkt gut in eine bestehende Datenverteilung passt. Obwohl die Mahalanobis-Distanz kein direktes Maß für Drift ist (Goldenberg & Webb, 2018), kann sie als Grundlage für die Drifterkennung verwendet werden, indem geprüft wird, ob neue Datenpunkte noch in die ursprüngliche Verteilung passen. Eine größere Mahalanobis-Distanz signalisiert, dass ein Punkt weiter von der typischen Verteilung entfernt ist, was auf Ausreißer oder Drift in den Daten hindeuten kann.

### 3.2.2 Distanzmaße zwischen Beobachtungen für multivariate Daten

Ein bedeutendes Distanzmaß in dieser Kategorie ist der Hotelling's  $T^2$ -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:

$$T^2 = (\bar{X} - \bar{Y})' S^{-1} (\bar{X} - \bar{Y}) \quad (6)$$

Formel 6: Hotelling's  $T^2$

Quelle: Goldenberg & Webb (2018)

Dabei wird  $S$  wie folgt berechnet wird:

$$S = \frac{(n_1-1)S_1 + (n_2-1)S_2}{n_1 + n_2 - 2} \quad (7)$$

Formel 7: Kovarianzmatrix des Hotelling's  $T^2$

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, ist Hotelling's  $T^2$  robust gegenüber Normalitätsannahmen bei großen Stichproben und auch auf hochdimensionale Daten anwendbar.

### 3.3 Fazit der datenverteilungsbasierten Ansätze

„Datenverteilungsbasierte Algorithmen bieten 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 leistungsbasierten 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 3 dargestellt sind.

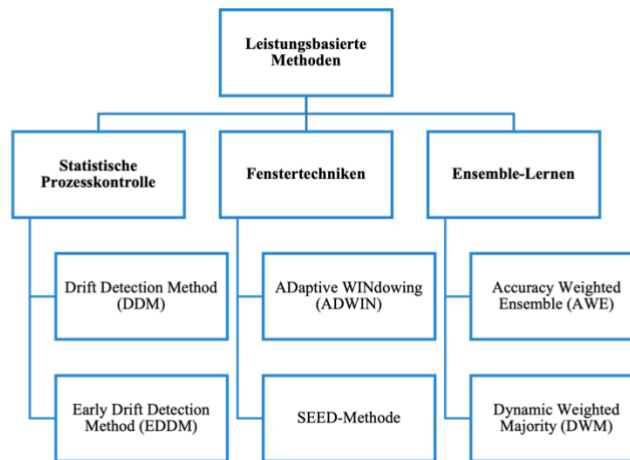


Abbildung 3: Hierarchie der leistungsbasierten Ansätze zur Drifterkennung  
Quelle: Eigene Darstellung

### 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).

Ein prominenter und weit verbreiteter Algorithmus in diesem Bereich ist die Drift Detection Method (DDM), 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:

$$s_t = \sqrt{\left(\frac{p_t(1-p_t)}{i}\right)} \quad (8)$$



Formel 8: 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 ( $2 \cdot s_{\min}$  und  $3 \cdot s_{\min}$ ) 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.

Um die Leistungsfähigkeit von DDM in verschiedenen Szenarien zu erhöhen, wurde zum Beispiel Early Drift Detection Method (EDDM) entwickelt. „EDDM erweitert DDM, indem es nicht nur die Fehlerrate überwacht, sondern auch den Abstand zwischen aufeinanderfolgenden Fehlklassifikationen“ (Bayram et al., 2022). „Dieser Ansatz eignet sich besonders für die Erkennung von allmählichen Drifts“ (Bayram et al., 2022), da er sensibler auf subtile Veränderungen in den Daten reagiert.

## 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).

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

$$\epsilon_{\text{cut}} = \sqrt{\frac{1}{2m} \ln \frac{4|W|}{\delta}} \quad (9)$$

Formel 9: Hoeffding-Schranke der ADWIN  
Quelle: Bayram et al. (2022)

Hierbei ist  $m$  der harmonische Mittelwert der beiden Fenster und  $|W|$  die Größe des Gesamtfensters.  $\delta$  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.

Ein weiteres Beispiel ist die SEED-Methode, 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  $\epsilon_{\text{cut}}$  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  $W_{\text{hist}}$  und  $W_{\text{new}}$  basiert, kombiniert SEED diese Korrektur, um Fehlalarme zu reduzieren und gleichzeitig die Empfindlichkeit zu erhöhen.

### 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.

Ein bekanntes Beispiel für einen ensemblebasierten Detektor ist der Accuracy Weighted Ensemble (AWE)-Algorithmus. „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 Leistungsver schlechterung 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.

Ein weiterer wichtiger Algorithmus in dieser Gruppe ist der Dynamic Weighted Majority (DWM)-Algorithmus. „DWM verwendet einen Gewichtungsmechanismus, der auf dem Weighted Majority Algorithmus (WMA) basiert. Jedes Modell im Ensemble erhält eine Gewichtung  $\beta$  ( $0 \leq \beta \leq 1$ ), 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 DWM und 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.

#### **4.4 Fazit der leistungsbasierten Ansätze**

Jede der drei Gruppen leistungsbasierter Ansätze hat ihre eigenen Stärken und Herausforderungen: Statistische Prozesskontrolle und Fenstertechniken überwachen Fehlerkennzahlen, um Modelldrift zu erkennen. Diese Methoden 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“ (Bayram et al., 2022), 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, 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 4) kombinieren die Ergebnisse verschiedener Drifterkennungsmethoden, um eine endgültige Entscheidung darüber zu treffen, ob ein Modelldrift vorliegt“ (Lu et al., 2018).

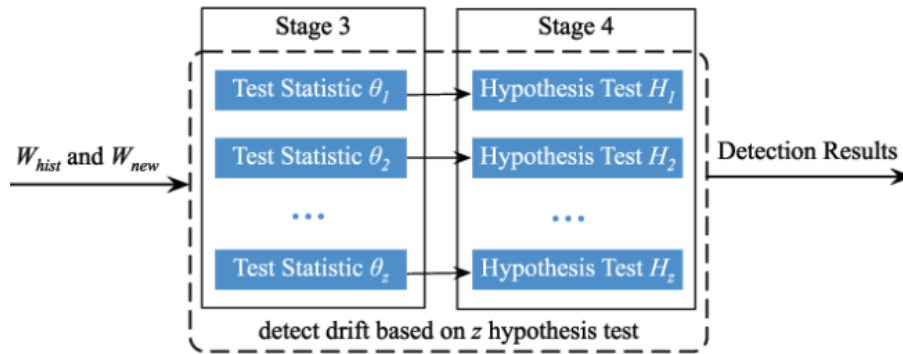


Abbildung 4: Parallele Mehrfachhypothesen - Driftdetektoren

Quelle: Lu et al. (2018)

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) ( $P_t(y)$ ) 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) ( $P_t(X)$ ) 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 ( $P_t(y | X)$ ) verändert hat“ (Zhang et al. 2017). Hierbei wird der Weight of Evidence (WoE) herangezogen, der misst, wie stark ein Feature die Zielvariable beeinflusst.

Der Algorithmus vergleicht die Werte von IV (für Schicht II) oder WoE (für Schicht III) aus historischen ( $(W_{hist})$ ) und aktuellen ( $(W_{new})$ ) Daten, um signifikante Veränderungen zu erkennen und damit eine Drift festzustellen.

## 5.2 Hierarchische Mehrfachhypothesen– Driftdetektoren

„Hierarchische Mehrfachhypothese-Drift-detecting-Algorithmen (Abbildung 5) arbeiten in zwei Ebenen: einer Warnebene, die potenzielle Modelldrift signalisiert, und einer Validierungsebene, die diese Signale überprüft“ (Lu et al., 2018).

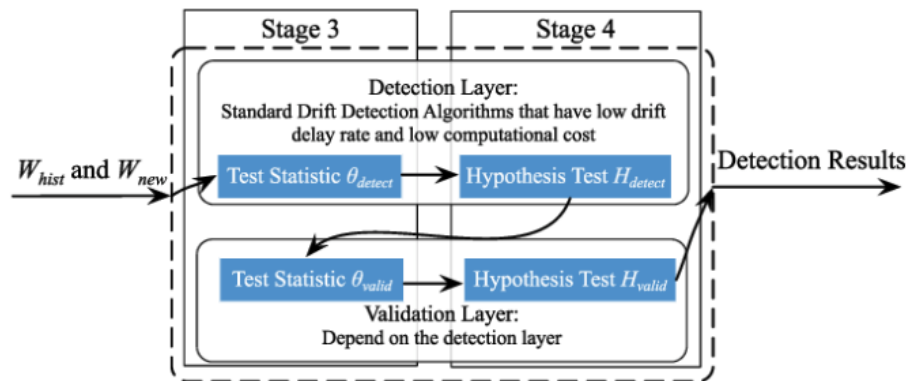


Abbildung 5: Hierarchische Mehrfachhypothesen - Driftdetektoren  
Quelle: Lu et al. (2018)

Ein bedeutendes Beispiel ist der Hierarchical Linear Four Rate (HLFR)-Algorithmus, der von Yu & Abraham (2017) vorgestellt wurde. Dabei verwendet die erste Ebene den Linear Four Rate (LFR)-Algorithmus, der Veränderungen in der Datenverteilung anhand statistischer Tests auf Drifts erkennt. In der zweiten Ebene kommt der Null-Eins-Verlust ( $E$ ), zum Einsatz, ein einfacher Maßstab, der misst, wie oft ein Modell falsche Vorhersagen macht. Dieser Verlust wird über Trainings- und Testdaten berechnet, und sobald  $E$  einen definierten Schwellenwert (z. B.  $h = 0,01$ ) überschreitet, wird ein Drift bestätigt. Das System löst dann eine Aktualisierung des Modells aus.

### 5.3 Fazit der Mehrfachhypothesen-basierten Ansätze

Mehrfachhypothese-basierte Ansätze kombinieren verschiedene Erkennungsstrategien und bieten eine höhere Genauigkeit sowie Anpassungsfähigkeit. Dies wird erreicht durch die systematische Validierung von Driftwarnungen, wie im HLFR-Modell, oder durch die Erkennung verschiedener Drifttypen in einer mehrschichtigen Struktur, wie im IV-Jac. Im Vergleich zu datenverteilungsbasierten Ansätzen, die schnellere Ergebnisse liefern können, oder leistungsbasierten Verfahren, die direkt auf Modellmetriken abzielen, bieten Mehrfachhypothese-Methoden eine robustere Lösung für komplexe Szenarien mit dynamischen Daten. Allerdings erfordern sie mehr Rechenressourcen und eine präzise Konfiguration, was ihre Anwendung in Echtzeit-Systemen erschwert.

## 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. 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.

Laut Lobo et al. (2018) verwendet die eSNN-DD-Methode Spiking Neural Networks (SNNs), die Informationen durch sogenannte “Spikes” (diskrete Ereignisse) verarbeiten. Ihre Aktivität entwickelt sich kontinuierlich im Zeitverlauf. Die Methode erkennt Modelldrift, indem sie die Evolution dieser Spike-Muster im Hinblick auf eingehende Daten überwacht. Wenn sich die Beziehung zwischen den Merkmalen ändert, verändert sich auch das Spike-Muster des Netzwerks, was als Hinweis auf Modelldrift dient. Diese Methode ist besonders wertvoll, da sie die Evolutionsdynamik der Daten berücksichtigt und somit in Echtzeit Drift erkennen kann, noch bevor signifikante Leistungseinbußen auftreten.

Kontextbasierte Ansätze wie die eSNN-DD-Methode bieten signifikante Vorteile gegenüber traditionellen Drifterkennungsansätzen. Anders als datenverteilungsbasierte und leistungsbasierte Methoden, die vor allem statistische Veränderungen oder Leistungsabfälle messen, „ermöglichen kontextbasierte Ansätze eine frühzeitige Erkennung von Drift“ (Lobo et al., 2018), indem sie die Entwicklung des Modells und dessen Reaktionen auf neue Daten untersuchen. Dies ermöglicht eine schnellere Identifikation von Modelldrift und eine dynamische Anpassung des Modells. Ein Nachteil kontextbasierter Ansätze, wie die Anwendung von Spiking Neural Networks, ist jedoch der erhöhte Rechenaufwand und die komplexere Modellstruktur, was ihre Nutzung in bestimmten praktischen Anwendungsbereichen 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 Fehllalarme zu erzeugen.

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

$$(\alpha + \text{mean}(\text{feature})) * \text{perc} \quad (10)$$

Formel 10: 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 „ $\alpha$ “-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 1: 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 2: 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 3: Ergebnisse der Experimente mit datenverteilungsbasierten Ansätzen  
Quelle: Eigene Darstellung

Testbedingungen	Hellinger-Distanz	KL-Divergenz	Kolmogorov-Smirnov-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%	✓	X	✓	✓	✓
10.0% segment moved by 10.0%	✓	X	✓	✓	✓
10.0% segment moved by 20.0%	✓	X	✓	✓	✓
10.0% segment moved by 50.0%	✓	X	✓	✓	✓
30.0% segment moved by 1.0%	✓	X	✓	✓	✓
30.0% segment moved by 5.0%	✓	X	✓	✓	✓
30.0% segment moved by 10.0%	✓	X	✓	✓	✓
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 Vergleich der verschiedenen statistischen Tests zeigte, dass der Kolmogorov-Smirnov-Test, die Wasserstein-Distanz und der T-Test die empfindlichsten Verfahren waren, da sie eine Veränderung in allen Szenarien unabhängig von der Größe oder dem Verhältnis der Drift erkannten. Die Hellinger-Distanz erkannte eine Drift bereits bei einer Verschiebung von 5 % der Daten bei einer Driftgröße von 50 %, war aber weniger empfindlich für kleinere Veränderungen. Die KL-Divergenz zeigte eine erkennbare Drift erst ab 20 % der Daten bei einer Driftgröße von 30 %. Dies könnte darauf zurückzuführen sein, dass die Stärke der Divergenz hauptsächlich in der Identifizierung großer struktureller Unterschiede zwischen den Verteilungen liegt.

Die Ergebnisse zeigen auch, dass die Effektivität der Tests sowohl von der Stärke der Abweichung (Driftgröße) als auch von der Segmentgröße (Driftratio) abhängt.

Zusammenfassend kann gesagt werden, dass für Szenarien, in denen kleine Änderungen in der Datenverteilung von Bedeutung sind, der Kolmogorov-Smirnov-Test, die Wasserstein-Distanz oder der T-Test empfohlen werden.

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 6 ist ein Auszug aus der Dokumentation der Bibliothek Evidently zu diesem Thema.

For **small data with  $\leq 1000$  observations** in the reference dataset:

- For numerical columns ( $n_{\text{unique}} > 5$ ): [two-sample Kolmogorov-Smirnov test](#).
- For categorical columns or numerical columns with  $n_{\text{unique}} \leq 5$ : [chi-squared test](#).
- For binary categorical features ( $n_{\text{unique}} \leq 2$ ): 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 ( $n_{\text{unique}} > 5$ ): [Wasserstein Distance](#).
- For categorical columns or numerical with  $n_{\text{unique}} \leq 5$ : [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 6: 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.

Im ersten Experiment wurden sieben verschiedene Drift-Erkennungsverfahren in alphabetischer Reihenfolge getestet: ADWIN (Fenster-Techniken), DDM (Statistische Prozesskontrolle), EDDM (Statistische Prozesskontrolle), HDDM\_A (Fenster-Techniken), HDDM\_W (Fenster-Techniken), KSWIN (Fenster-Techniken) und PageHinkley (Statistische Prozesskontrolle), jeweils in Kombination mit einem Naive Bayes-Klassifikator. Als Datensätze kamen vier unterschiedliche Generatoren zum Einsatz, die verschiedene Driftarten simulieren:

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 Fehllalarme, die Anzahl der verpassten Drifts sowie die Rechenzeit für jedes Verfahren erfasst.

Im zweiten Experiment kamen zwei Ensemble-Methoden zum Einsatz: der Accuracy Weighted Ensemble und der Dynamic Weighted Majority, 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 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**

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

Results for SineGenerator (Abrupt drift with a smooth, periodic change)

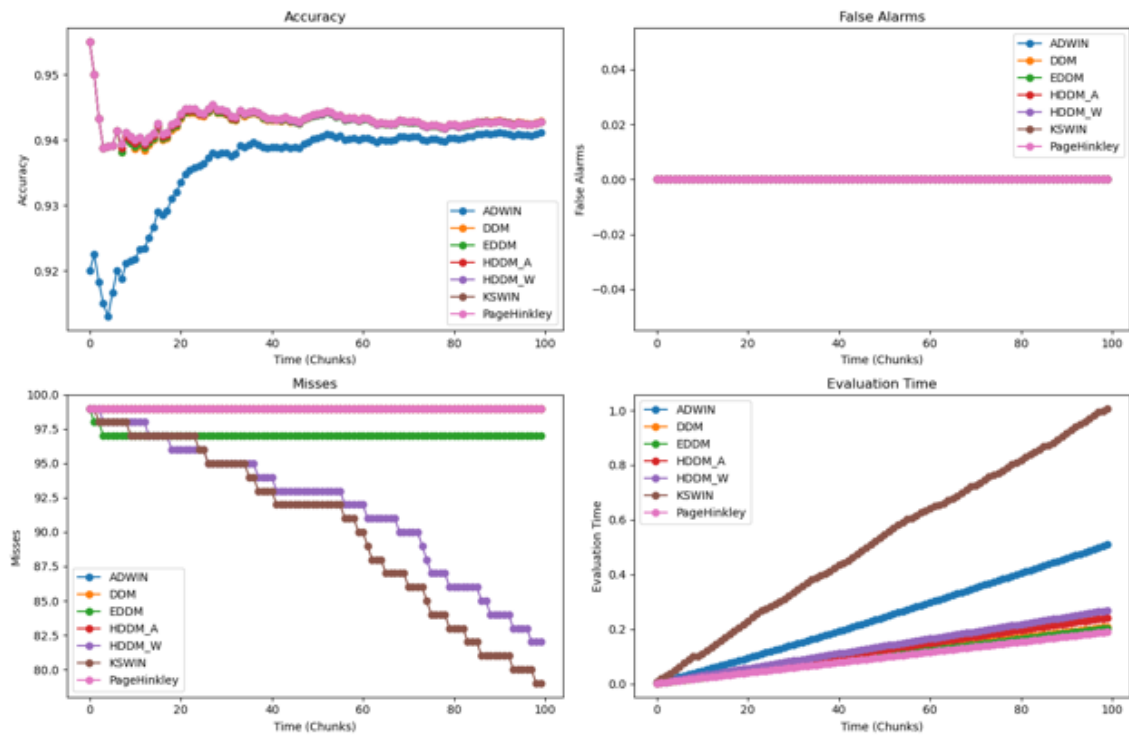


Abbildung 7: Ergebnisse der Experimente mit leistungsbasierten Ansätzen auf Sine  
Quelle: Eigene Darstellung

Beim Sine (abrupt mit glatten, periodischen Änderungen) erzielten alle Detektoren eine Genauigkeit von etwa 0.94, wobei PageHinkley anfangs die höchste Genauigkeit erreichte, diese jedoch später abnahm. Keiner der Detektoren registrierte Fehlalarme, was ihre Robustheit gegenüber Rauschen zeigte. In Bezug auf verpasste Erkennungen schnitten HDDM\_W und KSWIN am besten ab, während PageHinkley und EDDM die meisten verpassten Erkennungen aufwiesen. Bei der Auswertungszeit war KSWIN jedoch der langsamste Detektor, gefolgt von ADWIN, dessen etwas langsamere Anpassung an der Drift durch eine steigende Genauigkeit gerechtfertigt war. Die anderen Detektoren wiesen vergleichbare Rechenaufwände auf, wobei keine signifikanten Unterschiede zu verzeichnen waren.

Results for HyperplaneGenerator (Gradual drift with a simple linear decision boundary)

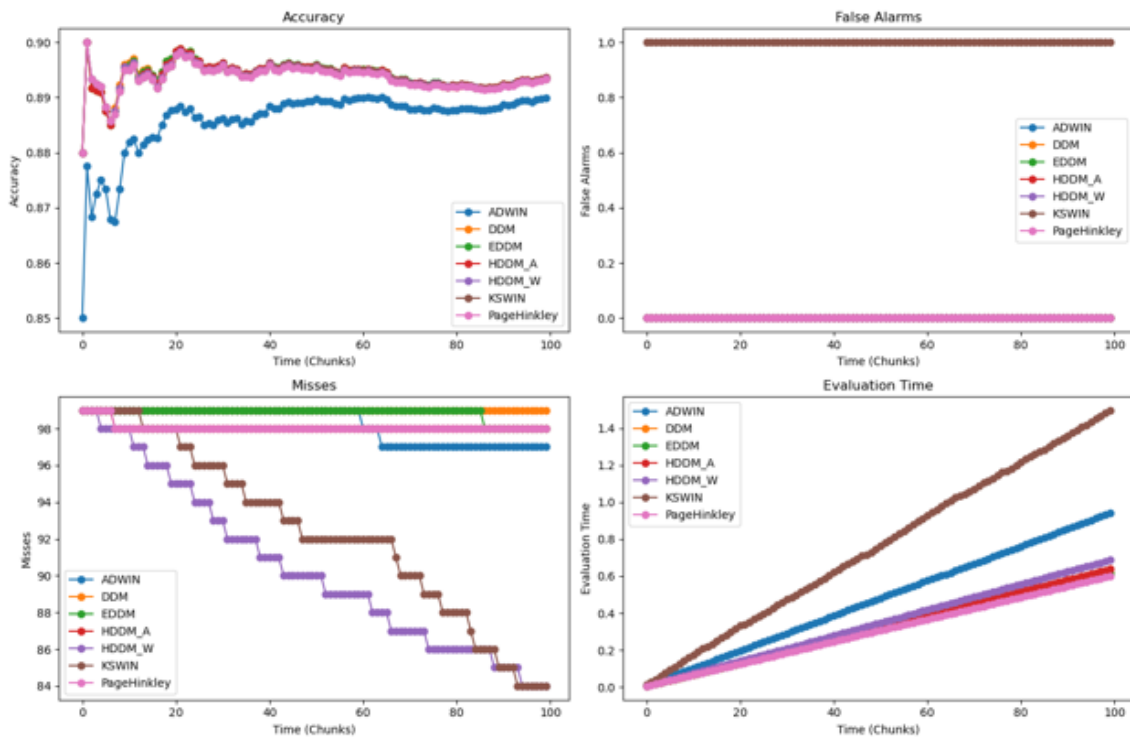


Abbildung 8: Ergebnisse der Experimente mit leistungsbasierten Ansätzen auf Hyperlane  
Quelle: Eigene Darstellung

Beim Hyperplane (gradual Drift mit einer linearen Entscheidungsgrenze) lagen alle Detektoren außer ADWIN bei einer Genauigkeit von etwa 0.895, während ADWIN mit einer Genauigkeit von 0.85 zu Beginn und 0.885 am Ende das schlechteste Ergebnis erzielte. KSWIN erzeugte Fehllalarme, während die anderen Detektoren keine Fehllalarme registrierten. In Bezug auf verpasste Erkennungen adaptieren wieder HDDM\_W und KSWIN besser, während die anderen Detektoren langsamer adaptierten und teils kaum Anpassungen zeigten. Die Auswertungszeit war ähnlich wie bei Sine, wobei KSWIN und ADWIN die langsamsten Detektoren waren und die anderen Detektoren eine vergleichbare Geschwindigkeit aufwiesen. Insgesamt benötigten jedoch alle Detektoren bis zu 50 % mehr Zeit als bei Sine.

Results for MIXEDGenerator (Gradual drift with noise and multiple features)

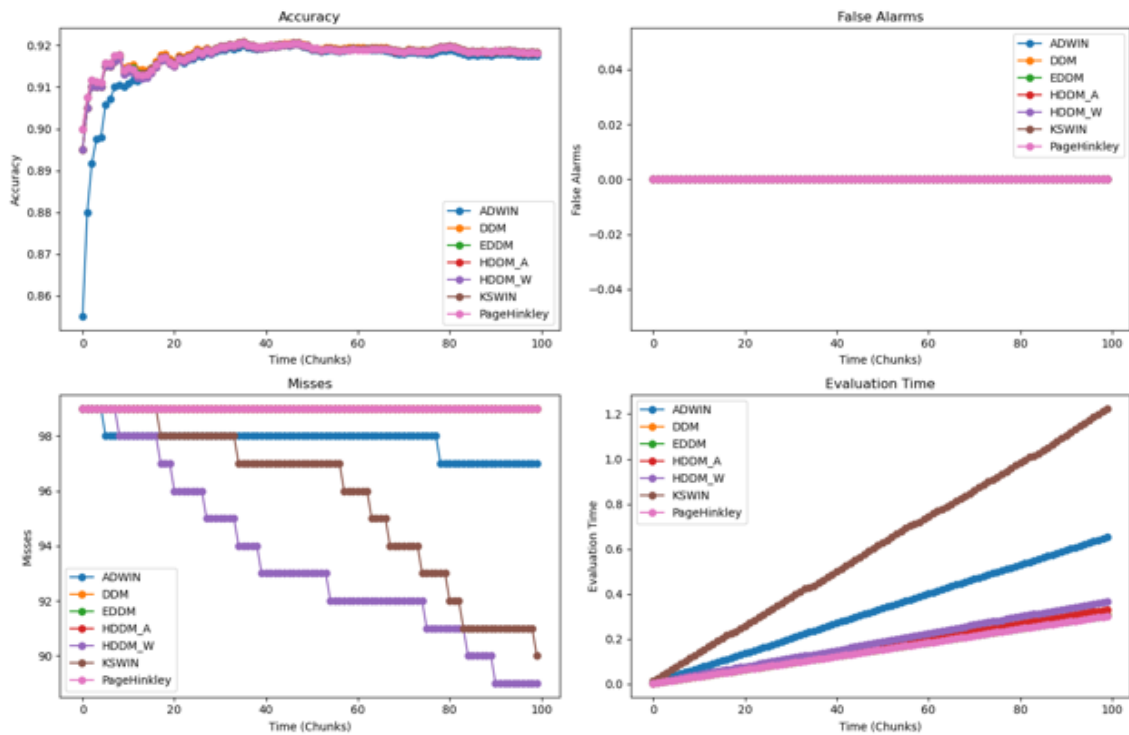


Abbildung 9: Ergebnisse der Experimente mit leistungsbasierten Ansätzen auf MIXED  
Quelle: Eigene Darstellung

Beim MIXED (gradual Drift mit Rauschen und mehreren Merkmalen) erzielten alle Detektoren ähnliche Ergebnisse wie beim Sine, mit einer Genauigkeit von 0,92. Es traten erneut keine Fehlalarme auf, und sowohl HDDM\_W als auch KSWIN zeigten eine bessere Anpassungsfähigkeit im Vergleich zu den anderen Detektoren. Wie zuvor benötigten KSWIN und ADWIN mehr Zeit für die Auswertung als die übrigen Detektoren.

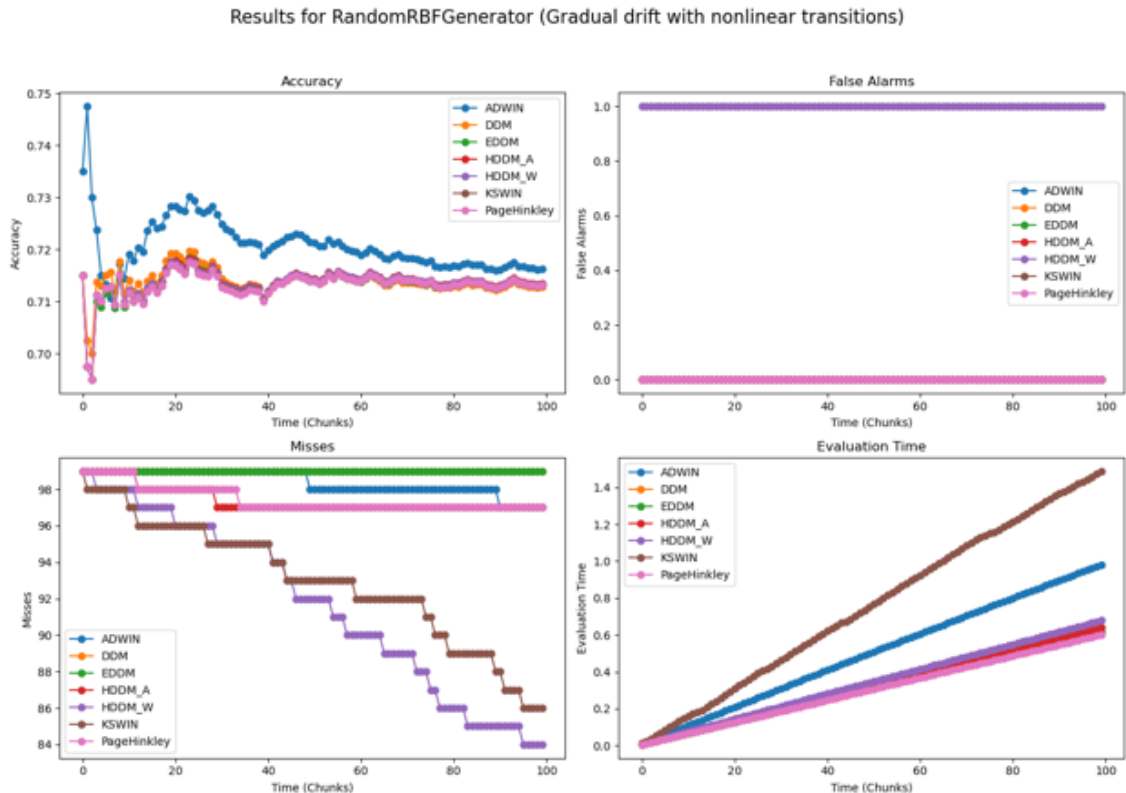


Abbildung 10: Ergebnisse der Experimente mit leistungs-basierten Ansätzen auf RandomRBF  
Quelle: Eigene Darstellung

Beim RandomRBF (gradual Drift mit nichtlinearen Übergängen) war die Leistung aller Detektoren deutlich schlechter, mit einer Genauigkeit zwischen 0,69 und 0,75. Alle Detektoren hatten Schwierigkeiten, sich an die nichtlineare Drift anzupassen. KSWIN schnitt jedoch etwas besser ab, während HDDM\_W aufgrund der Fehllarme auch in dieser Situation weniger effektiv war.

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. Gleichzeitig gibt es jedoch klare Unterschiede in der Verarbeitungszeit, die je nach Anwendungskontext abgewogen werden müssen:

- KSWIN erzielt in den meisten Testszenarien die höchste Genauigkeit, besonders bei abrupten und graduellen Drifts, erkennt auch subtile Veränderungen zuverlässig. Allerdings benötigt es mehr Rechenzeit und eignet sich daher für Anwendungen, bei denen Präzision Vorrang hat. Hier rechtfertigt die erhöhte Verarbeitungszeit den Gewinn an Präzision
- HDDM\_W liefert ebenfalls hohe Genauigkeit, liegt jedoch etwas hinter KSWIN, punktet aber mit kürzerer Verarbeitungszeit. Es kann bei subtilen Drifts weniger präzise sein, eignet sich jedoch besser für Echtzeitsysteme, bei denen Geschwindigkeit wichtiger ist und kleinere Genauigkeitseinbußen tolerierbar sind.

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

Tabelle 4: Ergebnisse der Experimente mit Ensemble-Methoden (leistungsbasierte Ansätze)  
Quelle: Eigene Darstellung

Datensatz	Klassifikator	Verarbeitungszeit (s)	Genauigkeit	Präzision	Recall	F1-Score
Sine	AWE	31.13	0.9730	0.9817	0.9592	0.9703
	DWM	15.53	0.9391	0.9503	0.9156	0.9326
Hyperplane	AWE	32.24	0.8781	0.8794	0.8773	0.8783
	DWM	16.15	0.8618	0.8603	0.8647	0.8625
RandomRBF	AWE	33.84	0.5496	0.5186	0.4945	0.5063
	DWM	18.17	0.5446	0.5130	0.4865	0.4994

Beim Sine (abrupte Drift mit glatten, periodischen Änderungen) erzielte der AWE exzellente Ergebnisse in Genauigkeit, Präzision, Recall und F1-Score, während der DWM solide, aber etwas schwächere Leistungen zeigte. Beim Hyperplane (graduelle Drift mit einer einfachen linearen Entscheidungsgrenze) erreichte der DWM ähnliche Ergebnisse wie der AWE mit stabilen Metriken. Beim RandomRBF (graduelle Drift mit nichtlinearen Übergängen) hatten beide Klassifikatoren Schwierigkeiten. In Bezug auf den Zeitaufwand benötigte der DWM nur halb so viel Zeit wie der AWE.

Zusammenfassend zeigte der AWE insgesamt bessere Leistungen, insbesondere bei abrupten Drifts, jedoch bei doppelt so hoher Verarbeitungszeit. Beide Klassifikatoren hatten außerdem Schwierigkeiten mit nichtlinearen Drifts, was zeigt, dass die Wahl eines geeigneten Klassifikators von der Art des Drifts im 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. Die datenverteilungsbasierten 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.



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

- Im Teil-Experiment mit Fenstertechniken und statistischer Prozesskontrolle erwiesen sich KSWIN und HDDM\_W als die leistungsfähigsten Detektoren. Hingegen war KSWIN im Vergleich zu anderen Verfahren besonders langsam und leistungsschwach.
- Im Teil-Experiment mit Ensemble-Lernmethoden zeigte der Accuracy Weighted Ensemble eine höhere Leistung als Dynamic Weighted Majority, benötigte jedoch doppelt so viel Zeit.
- Alle getesteten leistungsbasierten Ansätze hatten außerdem Schwierigkeiten, mit nichtlinearen Drifts umzugehen.

→ Diese Ergebnisse verdeutlichen, dass die Wahl des geeigneten Ansatzes stark von der Art des Drifts abhängt.

Die Experimente liefern insgesamt wertvolle Erkenntnisse zur Auswahl geeigneter Verfahren zur Drifterkennung, die sowohl Genauigkeit als auch Verarbeitungseffizienz berücksichtigen und in der Lage sind, verschiedene Driftgrößen und -arten effektiv zu identifizieren.

## **8 Zusammenfassung und Ausblick**

In dieser Arbeit wurden verschiedene Ansätze zur Erkennung von Modelldrift sowie deren Stärken und Schwächen untersucht. Datenverteilungsbasierte Ansätze sind genauer bei der Identifizierung von Driftzeitpunkten und -orten, können aber zu Fehlalarmen führen, da Änderungen in den Daten nicht immer Drifts verursachen. Leistungsbasierte Ansätze bieten dagegen robustere Lösungen mit Alarmen für die Modellleistung, erfordern aber die richtige Wahl der Metriken und das Vorhandensein wahrer Labels. Multihypothesen- und kontextbasierte Ansätze sind vielversprechend für komplexe Szenarien, erfordern aber mehr Rechenressourcen und komplexere Modelle.

Die Experimente zeigen, dass die Wahl des Erkennungsansatzes von der Art und Größe der Drift sowie den Effizienzanforderungen abhängt. Kein einzelner Ansatz ist für alle Szenarien gleichermaßen optimal, was das „No Free Lunch“-Prinzip (Bayram et al., 2022) bestätigt.

Zukünftige Forschung sollte sich auf die Erweiterung von hypothesen- und kontextbasierten Ansätzen sowie auf die Optimierung von Algorithmen für nichtlineare und hochdimensionale Drifts konzentrieren. Darüber hinaus könnte die Kombination verschiedener Strategien sowohl die Genauigkeit als auch die Effizienz verbessern und dazu beitragen, die Lücke zwischen Theorie und Praxis zu schließen.

## Literaturverzeichnis

- 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>
- 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>
- 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](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>
- 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>

## **Eigenständigkeitserklärung**

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit mit dem Titel:

---

selbständig und nur mit den angegebenen Hilfsmitteln verfasst habe. Alle Passagen, die ich wörtlich aus der Literatur oder aus anderen Quellen wie z. B. Internetseiten übernommen habe, habe ich deutlich als Zitat mit Angabe der Quelle kenntlich gemacht.

---

Datum

---

Unterschrift