HAW HAMBURG

BACHELORTHESIS
Jan von Appen

Konzeption und Evaluation von Verfahren für die Ausreißer-Erkennung / Outlier-Detection zur Identifikation von auffälligen Belegen in Buchungsjournalen aus Finanzbuchhaltungssystemen

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

Jan von Appen

Konzeption und Evaluation von Verfahren für die Ausreißer-Erkennung / Outlier-Detection zur Identifikation von auffälligen Belegen in Buchungsjournalen aus Finanzbuchhaltungssystemen

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang *Bachelor of Science Angewandte Informatik*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Martin Schultz
Zweitgutachter: Prof. Dr. Marina Tropmann-Frick

Eingereicht am: 15. Juni 2021

Jan von Appen

Thema der Arbeit

Konzeption und Evaluation von Verfahren für die Ausreißer-Erkennung / Outlier-Detection zur Identifikation von auffälligen Belegen in Buchungsjournalen aus Finanzbuchhaltungssystemen

Stichworte

Data Science, Ausreißer-Erkennung, Finanzbuchhaltung, Betrugserkennung, unüberwachtes Lernen, DBSCAN, OPTICS, LOF, ROCK, Autoencoder

Kurzzusammenfassung

Durch die stetige Automatisierung und Digitalisierung der Geschäftsprozesse im 21. Jahrhundert entstehen immer mehr Finanztransaktionen, die auf unbeabsichtigte Fehler oder Betrugsversuche analysiert werden müssen. Das bisherige Vorgehen ist die manuelle Überprüfung eines Ausschnittes aller Transaktionen oder die Analyse aller Transaktionen mit Hilfe vordefinierter statischer Regeln. In dieser Arbeit wird evaluiert, ob die unüberwachten Verfahren DBSCAN, OPTICS, LOF, ROCK und Autoencoder sinnvoll in der Wirtschaftsprüfung angewendet werden können. Hierzu wurden die Verfahren auf drei Konten erprobt, die vorher von Wirtschaftsprüfern analysiert wurden. In dem zweiten Teil der Arbeit wird ein Versuch unternommen, mit Hilfe einer verbesserten Datenvorverarbeitung, die erreichten Ergebnisse zu verbessern.

Jan von Appen

Title of Thesis

Design and evaluation of outlier detection methods for the identification of suspicious transactions in accounting journals from financial accounting systems

Keywords

data science, outlier-detection, financial accounting, fraud detection, unsupervised learning, DBSCAN, OPTICS, LOF, ROCK, Autoencoder

Abstract

The steady automation and digitalisation of business processes in the 21st century is creating more and more financial transactions that need to be analysed for unintentional errors or fraud attempts. The current approach is to manually review a subset of all transactions or to analyse all transactions using predefined static rules. This thesis evaluates if the unsupervised methods DBSCAN, OPTICS, LOF, ROCK and autoencoder can be applied in the auditing process. For this purpose, the procedures were tested on three accounts that were previously analysed by auditors. In the second part of the thesis an attempt is made to improve the achieved results by improving the data preprocessing.

Inhaltsverzeichnis

Abbildungsverzeichnis **vii**

Tabellenverzeichnis **viii**

Abkürzungen **x**

1 Einleitung **1**
1.1 Problemstellung und Motivation 1
1.2 Ziel der Arbeit 2
1.3 Aufbau der Arbeit 2

2 Grundlagen **4**
2.1 Ausreißer-Erkennung 4
2.2 Funktionsweise der Verfahren 6
2.2.1 DBSCAN 6
2.2.2 OPTICS 6
2.2.3 LOF 8
2.2.4 ROCK 9
2.2.5 Autoencoder 10

3 Datenvorverarbeitung **12**
3.1 Struktur und Aufbau des Datensatzes 12
3.2 Deskriptive Analyse des Datensatzes 14
3.3 Auswahl geeigneter Attribute 16
3.4 Verarbeitung kategorialer Attribute 17
3.5 Verarbeitung des metrischen Attributs 18
3.6 Ergebnis der Datenvorverarbeitung 19

4 Erprobung der Verfahren **22**
4.1 Vorgehensweise und Testaufbau 22

4.2 Erprobung von DBSCAN 25
4.3 Erprobung von OPTICS 27
4.4 Erprobung von LOF 32
4.5 Erprobung von ROCK 36
4.6 Erprobung von Autoencodern 39
4.7 Bewertung der Ergebnisse 43

5 Verbesserte Datenvorverarbeitung 44
5.1 Motivation 44
5.2 Erzeugung Features 44
5.3 Ergebnis der verbesserten Datenvorverarbeitung 46

6 Erprobung der Verfahren auf dem erweitertem Datensatz 49
6.1 Erprobung von DBSCAN 49
6.2 Erprobung von OPTICS 51
6.3 Erprobung von LOF 53
6.4 Erprobung von ROCK 55
6.5 Erprobung von Autoencodern 57
6.6 Bewertung der Ergebnisse 59

7 Fazit 60
7.1 Zusammenfassung 60
7.2 Mögliche zukünftige Verbesserungen und Ausblick 61

Literaturverzeichnis 63

A Anhang 66
A.1 Datensatzbeschreibung 66
A.2 Distanzen nach der ersten Datenvorverarbeitung 68
A.3 Distanzen nach der zweiten Datenvorverarbeitung 69

Selbstständigkeitserklärung 71

Abbildungsverzeichnis

2.1 Beispielhafte Visualisierung von OPTICS [ABpKS99] 8
2.2 Aufbau und Funktionsweise eines Autoencoder [SSB+18] 11

3.1 Überblick über die Verteilung des Attributs Betrag 15
3.2 Vergleich verschiedener Kodierungsansätz 18
3.3 Distanzverteilung in Umsatzerlöse EU 20
3.4 Distanzverteilung in Verbrauch Verpackungen 21
3.5 Distanzverteilung in Umsatzerlöse Inland 21

4.1 Erreichbarkeitsdistanzen von OPTICS auf dem Konto Umsatzerlöse EU . 29
4.2 Erreichbarkeitsdistanzen von OPTICS auf dem Konto Verbrauch Verpackungen 30
4.3 Erreichbarkeitsdistanzen von OPTICS auf dem Konto Umsatzerlöse Inland 31
4.4 Ausreißer Score von LOF auf dem Konto Umsatzerlöse EU 33
4.5 Ausreißer Score von LOF auf dem Konto Verbrauch Verpackungen 34
4.6 Ausreißer Score von LOF auf dem Konto Umsatzerlöse Inland 35
4.7 Entwicklung des Modells auf dem Konto Umsatzerlöse EU 40
4.8 Ausreißer Score von Autoencoder auf dem Konto Umsatzerlöse EU 40
4.9 Ausreißer Score von Autoencoder auf dem Konto Verbrauch Verpackungen 41
4.10 Ausreißer Score von Autoencoder auf dem Konto Umsatzerlöse Inland . . 42

5.1 Distanzverteilung in Umsatzerlöse EU 47
5.2 Distanzverteilung in Verbrauch Verpackungen 48
5.3 Distanzverteilung in Umsatzerlöse Inland 48

Tabellenverzeichnis

3.1 Überblick über die drei Konten 14

4.1 Ergebnisse von DBSCAN auf dem Konto Umsatzerlöse EU 26
4.2 Ergebnisse von DBSCAN auf dem Konto Verbrauch Verpackungen 26
4.3 Ergebnisse von DBSCAN auf dem Konto Umsatzerlöse Inland 27
4.4 Ergebnisse von OPTICS auf dem Konto Umsatzerlöse EU 29
4.5 Ergebnisse von OPTICS auf dem Konto Verbrauch Verpackungen 30
4.6 Ergebnisse von OPTICS auf dem Konto Umsatzerlöse Inland 31
4.7 Ergebnisse von LOF auf dem Konto Umsatzerlöse EU 33
4.8 Ergebnisse von LOF auf dem Konto Verbrauch Verpackungen 34
4.9 Ergebnisse von LOF auf dem Konto Umsatzerlöse Inland 35
4.10 Ergebnisse von ROCK auf dem Konto Umsatzerlöse EU 37
4.11 Ergebnisse von ROCK auf dem Konto Verbrauch Verpackungen 38
4.12 Ergebnisse von ROCK auf dem Konto Umsatzerlöse Inland 38
4.13 Ergebnisse von Autoencoder auf dem Konto Umsatzerlöse EU 41
4.14 Ergebnisse von Autoencoder auf dem Konto Verbrauch Verpackungen . . . 42
4.15 Ergebnisse von Autoencoder auf dem Konto Umsatzerlöse Inland 43

5.1 Vorkommen von geraden Beträgen 46

6.1 Ergebnisse von DBSCAN auf dem Konto Umsatzerlöse EU 50
6.2 Ergebnisse von DBSCAN auf dem Konto Verbrauch Verpackungen 50
6.3 Ergebnisse von DBSCAN auf dem Konto Umsatzerlöse Inland 51
6.4 Ergebnisse von OPTICS auf dem Konto Umsatzerlöse EU 52
6.5 Ergebnisse von OPTICS auf dem Konto Verbrauch Verpackungen 52
6.6 Ergebnisse von OPTICS auf dem Konto Umsatzerlöse Inland 53
6.7 Ergebnisse von LOF auf dem Konto Umsatzerlöse EU 54
6.8 Ergebnisse von LOF auf dem Konto Verbrauch Verpackungen 54
6.9 Ergebnisse von LOF auf dem Konto Umsatzerlöse Inland 55

6.10 Ergebnisse von ROCK auf dem Konto Umsatzerlöse EU 56
6.11 Ergebnisse von ROCK auf dem Konto Verbrauch Verpackungen 56
6.12 Ergebnisse von ROCK auf dem Konto Umsatzerlöse Inland 57
6.13 Ergebnisse von Autoencoder auf dem Konto Umsatzerlöse EU 58
6.14 Ergebnisse von Autoencoder auf dem Konto Verbrauch Verpackungen . . . 58
6.15 Ergebnisse von Autoencoder auf dem Konto Umsatzerlöse Inland 59

A.1 Erläuterung der Attribute in dem Datensatz 67
A.2 Abstände zwischen den Datenpunkten nach der erste Datenvorverarbeitung 68
A.3 Abstände zwischen den Datenpunkten nach der zweiten Datenvorverarbeitung 70

Abkürzungen

CBLOF Cluster-Based Local Outlier Factor.

CSV Comma-separated values.

DBSCAN Density-Based Spatial Clustering of Applications with Noise.

ERP Enterprise-Resource-Planning.

HDBSCAN Hierarchical Density-Based Spatial Clustering of Applications with Noise.

LOF Local Outlier Factor.

OPTICS Ordering Points To Identify the Clustering Structure.

OPTICS-OF Ordering Points To Identify the Clustering Structure - Outlier Factors.

PCA Principal Component Analysis.

ROCK A Robust Clustering Algorithm for Categorical Attributes.

SAP Systemanalyse Programmentwicklung.

1 Einleitung

In diesem Kapitel werden die Probleme der Identifikation von auffälligen Belegen, die durch die steigende Komplexität und zunehmender Automatisierung von Geschäftsprozessen entstehen, beschrieben. Die Ziele dieser Arbeit werden im Abschnitt 1.2 definiert. Anschließend wird im Abschnitt 1.3 auf den Aufbau dieser Arbeit eingegangen.

1.1 Problemstellung und Motivation

Die Welt des 21. Jahrhunderts entwickelt sich stetig weiter und immer mehr Aspekte des Lebens werden digitalisiert und automatisiert. Eine ähnliche Entwicklung ist auch in den heutigen Unternehmen zu erkennen, wie sich beispielhaft an der Umsatzentwicklung von SAP erkennen lässt [SAP20]. SAP ist das dritt größte börsennotierte Unternehmen weltweit, das sich auf die Entwicklung von Software zur Abwicklung aller Geschäftsprozesse in einem Unternehmen spezialisiert hat.

Durch diese Automatisierung entstehen nicht nur immer mehr Transaktionen, sondern die einzelnen Transaktionen selbst wachsen in der Anzahl ihrer Attribute, um die stetige Entwicklung durch neue Gesetze und komplexer werdende Geschäftsprozesse abzubilden. Große Unternehmen sind verpflichtet ihre Jahresabschlüsse jährlich durch externe Wirtschaftsprüfer prüfen zu lassen [Fle18]. Dadurch besteht der dringende Bedarf alle Transaktionen automatisiert auf unbeabsichtigte Fehler oder Betrugsversuche hin zu analysieren.

Der bisherige Ansatz sieht vor, dass die Wirtschaftsprüfer nur eine Stichprobe alle Transaktionen analysieren [BAAG+18]. Hier werden die Transaktionen eines Geschäftsprozesses von ihrem Ursprung bis zu ihrer Beendigung verfolgt [KS16]. Dies ermöglicht den Wirtschaftsprüfern zwar ein tieferes Verständnis des Geschäftsprozesses, jedoch birgt

es das Risiko seltene auffällige Transaktionen zu übersehen. Besonders unter der Berücksichtigung, dass weniger als ein Prozent aller Transaktionen manuell geprüft werden [NLHL19].

Ein weiterer Ansatz ist es den gesamten Datensatz, anhand von vordefinierten Regeln und Mustern, zu analysieren [AKV17]. Durch das Definieren von Regeln besteht das Risiko, dass diese zu allgemein definiert sind und somit zu viele unauffällige Transaktionen als Ausreißer erkannt werden oder reale Ausreißer gar nicht erkannt werden. Ein weiteres Problem dieses Ansatzes ist, dass neuartige Anomalien nicht erkannt werden, da der Datenbestand nur nach bereits bekannten Mustern durchsucht wird. Besonders bei betrügerischen Absichten wird aktiv versucht unerkannt zu bleiben.

Durch die exponentiell wachsende Anzahl an Datenmengen steigt das Risiko mit Hilfe der bisherigen Ansätze einzelne auffällige Transaktionen zu übersehen [BAAG$^+$18]. Es besteht daher ein dringender Bedarf alle Transaktionen vollumfänglich auf bereits bekannte und neue Muster hin automatisiert zu überprüfen.

1.2 Ziel der Arbeit

Es soll mit dieser Arbeit überprüft werden, ob bekannte Verfahren aus den Bereichen Clustering und maschinellem Lernen in der Wirtschaftsprüfung sinnvoll angewendet werden können. Es werden fünf unüberwachte Verfahren DBSCAN, OPTICS, LOF, ROCK und Autoencoder evaluiert. Im Gegensatz zu anderen Arbeiten [SSB$^+$18] werden die Verfahren einzeln auf den drei zur Verfügung stehenden Konten getestet, da jedes Konto ein eigenes Verhalten aufweist. Zudem sollen die Auswirkungen verschiedener Ansätze der Datenvorverarbeitung auf die Qualität der Ausreißererkennung untersucht werden. Sowie die Wahl der Parameter für die einzelnen Verfahren möglichst nachvollziehbar erläutert werden.

1.3 Aufbau der Arbeit

Diese Arbeit gliedert sich in sechs Kapitel. In Kapitel 2 werden die Grundlagen der Ausreißererkennung erläutert. Anschließend werden die Funktionsweisen der Verfahren, die im Laufe der Arbeit erprobt werden, erläutert. Hier wird auf die benötigten Eingabe-Parameter der Verfahren und deren Auswirkungen eingegangen. In Kapitel 3 wird der

Datensatz zuerst vorgestellt und seine Struktur analysiert. Anschließend befasst sich das Kapitel mit der Auswahl geeigneter Attribute für die Datenvorverarbeitung sowie mit der Umwandlung der ausgewählten Attribute in eine Form, mit der die Verfahren arbeiten können. Abschließend wird das Resultat der Datenvorverarbeitung beschrieben. In Kapitel 4 wird die Vorgehensweise und der Testaufbau vorgestellt. Alle in Kapitel 2 vorgestellten Verfahren werden mit der Datenvorverarbeitung aus Kapitel 3 erprobt und deren erreichte Leistung bewertet. Hier wird auch die Wahl der Parameter genauer erläutert. Kapitel 5 handelt von einer alternativen Datenvorverarbeitung, bei der zusätzliche Informationen aus dem Datensatz extrahiert werden und einer anderen Kodierung für ordinale Attribute und Listen verwendet wird. Es folgt, wie in Kapitel 3, ein Überblick über das Resultat der Datenvorverarbeitung. In Kapitel 6 werden dieselben Verfahren mit der Datenvorverarbeitung aus Kapitel 5 erprobt. Alle erreichten Leistungen innerhalb der Arbeit werden in Kapitel 7 zusammengefasst. Es folgt ein Ausblick auf mögliche zukünftige Verbesserungen.

2 Grundlagen

In diesem Kapitel wird zuerst das Problem der Ausreißererkennung im Abschnitt 2.1 erklärt. Alle Verfahren werden im Abschnitt 2.2 vorgestellt und deren Funktionsweise erläutert.

2.1 Ausreißer-Erkennung

Ausreißererkennung (Outlier Detection) hat das Ziel Objekte aus der Menge aller Objekte zu identifizieren, die sich von den anderen Objekten deutlich unterscheiden. Typische Anwendungsfälle sind die Betrugserkennung im Finanzsektor [ST20, CSAT17, SSB+18], Erkennung von Marktnischen, medizinische Überwachung und Identifikation von Netzwerkanomalien [SMG19]. Ausreißererkennung wird auch in der Datenbereinigung eingesetzt, um stark abweichende Objekte zu entfernen, die sonst negativen Einfluss auf die folgenden Analysen gehabt hätten.

Ein Objekt wird als Ausreißer bezeichnet, wenn es vom bekannten Verhalten des Datensatzes abweicht, sich in einem Wertebereich befindet, der außerhalb der erwarteten Werte liegt oder keine Ähnlichkeiten mit allen anderen Objekten besitzt [SMG19]. Eine oft zitierte Definition von Ausreißern lautet wie folgt:

> „An observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism“ - D. M. Hawkins, 1980 [Haw80]

Ausreißer können durch verschiedene Mechanismen entstehen. Es kann sich dabei um wirkliche Fehler handeln. Diese können mitunter durch falsche Eingabe von einem Menschen, Messfehler von Sensoren oder durch Fehler in der Handhabung des Datensatzes

entstehen. Es ist aber auch möglich, dass es sich bei einem Ausreißer nicht um einen Fehler, sondern um ein neues Verhalten aus der realen Welt handelt, dass in dem Datensatz noch nicht aufgetaucht ist.

Im Allgemeinen können Datensätze in zwei Gruppen unterschieden werden. Es gibt zum einen univariate (eindimensionale) oder multivariate (mehrdimensionale) Datensätze. Eindimensionale Datensätze sind meist einfacher zu analysieren, da diese gut für einen Menschen visualisiert werden können. Mit Hilfe eines Boxplots können hier stark abweichende Objekte einfach identifiziert werden. Anders ist es bei Datensätze mit einer großen Anzahl an Dimensionen. Für einen Menschen ist es schwer möglich eine große Anzahl Dimensionen gleichzeitig zu betrachten und dabei alle Zusammenhänge zwischen den Dimensionen zu erkennen. Dieses Problem wird meist umgangen, indem nur eine Auswahl von Features betrachtet werden oder indem die Anzahl der Dimensionen reduziert wird. Ein möglicher Ansatz hierfür ist die PCA. Hierbei verändert sich jedoch die Bedeutung der einzelnen Dimensionen und ein Teil der Informationen geht im Laufe des Prozesses verloren.

Ausreißererkennung wird für gewöhnlich als binäres Klassifikationsproblem betrachtet [SMG19]. Entweder wird ein Objekt zu der Gruppe von unauffällig Datenpunkten (Inliner) oder zur Gruppe der auffälligen Datenpunkte (Outlier) zugeordnet. Die Verfahren selbst können in drei Kategorien unterteilt werden [SMG19]. Zum einen gibt es überwachte Verfahren. Hierbei wird zu jedem Datenpunkt die Information mitgegeben zu welcher Gruppe dieser gehört. Bei one-class classification wird dem Verfahren nur ein Datensatz aus unauffälligen Datenpunkten übergeben. Das Verfahren lernt so nur mit den korrekten Daten. Bei der späteren Analyse von neuen Daten wird überprüft, ob diese den bereits gelernten Daten ähneln oder als Ausreißer zu bezeichnen sind. Diese beiden Ansätze bringen eine Reihe von Problemen mit sich. Zum einem muss vorher jeder Datenpunkt einer der beiden Gruppe zugeordnet werden. Diese Aufgabe ist nicht nur zeitintensiv, sondern es bedarf auch qualifizierter Mitarbeiter, die sich in der entsprechenden Domain auskennen. Die spätere Qualität der Ergebnisse wird stark von der Qualität des Trainingsdatensatzes beeinflusst. Ein weiterer Nachteil ist, dass dem Verfahren bereits (implizit oder explizit) Regeln vorgegeben werden. Sollten neuartige Ausreißer auftreten, besteht das Risiko, dass diese nicht erkannt werden. Bei unüberwachten Verfahren wird der gesamte Datensatz, welcher aus auffälligen und unauffälligen Datenpunkten besteht, an das Verfahren übergeben. Das Verfahren muss dann eigenständig erkennen nach welchen Regeln die Objekte bewertet werden. Dies bringt zwei Vorteile mit sich. Auf der einen Seite muss kein

Trainingsdatensatz erstellt werden und auf der anderen Seite werden neuartige Ausreißer automatisch erkannt.

2.2 Funktionsweise der Verfahren

Im folgenden Abschnitt werden alle Verfahren, die im Rahmen dieser Arbeit analysiert werden, einmal bezüglich ihrer Funktionsweise vorgestellt. Anschließend werden die Parameter erläutert, sowie deren Bedeutung und Auswirkung auf die Qualität der Ergebnisse beleuchtet.

2.2.1 DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) ist ein dichtebasiertes Clustering-Verfahren, dass von Martin Ester, Hans-Peter Kriegel, Jörg Sander und Xiaowei Xu [EKSX96] entwickelt wurde.

DBSCAN benötigt die beiden Eingabe-Parameter ε und *minimum number of points* (*minPTS*). Hierbei bestimmt ε den maximalen Abstand zwischen zwei Punkten, um als erreichbar bezeichnet zu werden. *minPTS* bestimmt hingegen wie viele erreichbare Nachbarn ein Punkt mindestens haben muss, damit dieser als dicht bezeichnet wird. Ein Punkt wird als Kernobjekt bezeichnet, wenn er erreichbar und dicht ist und bildet mit allen erreichbaren Punkten ein Cluster. Datenpunkte, die weder erreichbar noch dicht sind, werden als Rauschpunkte bezeichnet.

2.2.2 OPTICS

OPTICS (Ordering Points To Identify the Clustering Structure) ist ein dichtebasiertes Clustering-Verfahren, dass als eine Erweiterung von DBSCAN [EKSX96] zu verstehen ist. Es wurde von Mihael Ankerst, Markus M. Breunig, Hans Peter Kriegel und Jörg Sander entwickelt [ABpKS99].

Ein Problem von hoch dimensionalen Datensätzen ist, dass diese eine ungleiche Verteilung der Datenpunkte aufweisen können und somit durch die Wahl globaler Dichte-Parametern die Cluster-Struktur nicht optimal erfasst werden kann [ABpKS99]. Zusätzlich ist ein Verfahren wie DBSCAN sehr empfindlich im Bezug auf die Eingabe-Parameter. Deshalb

haben bereits kleine Veränderungen der Parameter große Auswirkungen auf die Qualität der Ergebnisse [EKSX96].

OPTICS löst diese Probleme, indem es nicht explizit eine Cluster-Struktur vorgibt, sondern alle Datenpunkte in einer linearen Liste sortiert. Hierbei werden Datenpunkte, die in einem dichteren Cluster sind, näher voneinander in die Liste einsortiert. Aus dieser Liste können dann alle nötigen Informationen, wie die Cluster-Struktur oder deren Form, extrahiert werden [EKSX96].

Ähnlich zu DBSCAN gibt es die beiden Parameter ε und *minPTS*. Diese haben jedoch eine leicht veränderte Bedeutung. Der Parameter ε dient hier als den maximalen Radius für eine ε-Umgebung. Durch die Reduzierung dieses Parameters kann die Komplexität des Verfahrens verringert werden. Wird dieser Parameter zu gering gewählt, hat dies eine negative Auswirkungen auf die Leistung des Verfahrens. Der Parameter *minPTS* gibt an, wie viele Datenpunkte mindestens in der ε-Umgebung liegen müssen, damit dieser Datenpunkt als Kernpunkt betrachtet wird.

Um die Datenpunkte korrekt in die Liste zu sortieren, benötigt OPTICS zwei weitere Informationen. Zum einen die Kerndistanz eines Datenpunkt p, welche definiert ist durch den kleinsten Wert ε', sodass die ε-Umgebung vom Punkt p mindestens *minPTS* Datenpunkte aufweist. Wenn nicht genügend Datenpunkte in der ε-Umgebung vorhanden sind, wird die Kerndistanz des Datenpunktes p auf *undefined* gesetzt. Des Weiteren wird die Erreichbarkeitsdistanz von einem Punkt p, ausgehend von einem anderen Punkt q, als der minimale Radius definiert, durch den Punkt p dichte-erreichbar von q wird. Die Erreichbarkeitsdistanz wird berechnet aus dem Maximum der Kerndistanz von q und dem Abstand zwischen p und q. Sollte q kein Kernpunkt sein, ist die Erreichbarkeitsdistanz *undefined*.

OPTICS startet bei einem beliebig ausgewählten Punkt p. Anschließend wird die ε-Umgebung von p und die Kerndistanz von p ermittelt. Die Erreichbarkeitsdistanz wird auf *undefined* gesetzt. Wenn p ein Kernobjekt ist, werden alle Datenpunkte in der ε-Umgebung als nächstes berechnet, sonst wird ein weiterer beliebiger Punkt analysiert. Die Datenpunkte werden in Abhängigkeit der Zugehörigkeit zu einem Cluster und ihren benachbarten Punkten in die Liste einsortiert.

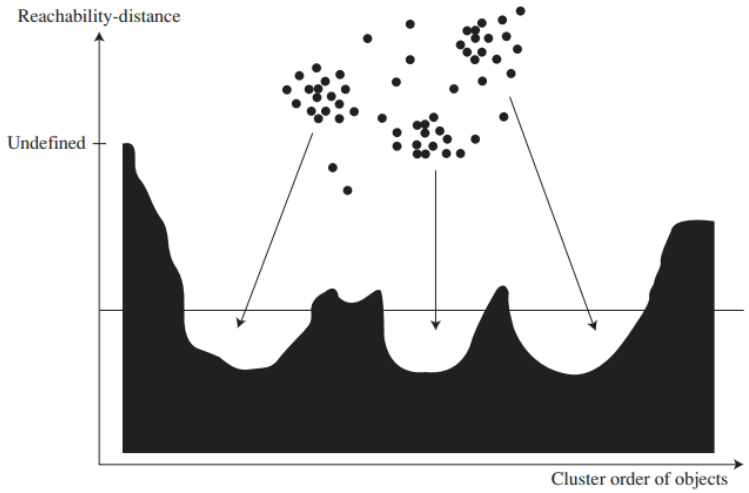

Abbildung 2.1: Beispielhafte Visualisierung von OPTICS [ABpKS99]

Das Ergebnis dieses Verfahrens ist beispielhaft in Abbildung 2.1 zu sehen. Die von OPTICS erzeugte Liste wird an der x-Achse dargestellt. Die y-Achse zeigt die Erreichbarkeitsdistanz des jeweiligen Datenpunktes. Die einzelnen Cluster erzeugen so Täler, während weit entfernte Datenpunkte, die potenzielle Ausreißer sind, als Hügel zu erkennen sind.

2.2.3 LOF

LOF (Local Outlier Factor) ist ein Verfahren von Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng und Jörg Sander [BKNS00]. Das Konzept des Verfahrens ist es, die lokale Dichte eines Punktes mit der Dichte seiner Nachbarn zu vergleichen. Das Ziel ist es keine binäre Unterscheidung zwischen einem Outlier und einem Inliner zu unternehmen. LOF berechnet für jeden Datenpunkt eine Bewertung, die angibt, wie auffällig dieser Datenpunkt ist. Dies hat den Vorteil, dass Datenpunkte mit höherer Bewertung zuerst analysiert werden können.

LOF benötigt als einzigen Parameter *minimum number of points* (*minPTS*). Dieser gibt an, wie viele der nächsten Nachbarn von Punkt p berücksichtigt werden, um die lokale Dichte des Punktes p zu berechnen.

Die k-Distanz für ein Punkt p ist die Distanz zu seinem k nächsten Nachbarn. Die k-Nachbarschaft von einem Punkt p beschreibt alle Punkte, die mit der k-Distanz erreichbar

sind. Dies können mehr als k Punkte sein, wenn es mehrere Punkte mit der gleichen k-Distanz gibt. Die Erreichbarkeitsdistanz von einem Punkt p von einem Punkt q ist definiert durch das Maximum aus der k-Distanz von Punkt q und der Distanz zwischen Punkt p und q.

Die lokale Erreichbarkeitsdichte ("local reachability density", "lrd") eines Punktes p berechnet sich aus dem Inversen der durchschnittlichen Erreichbarkeitsdistanz der k-Nachbarschaft. Die Formel lautet wie folgt:

$$lrd_{minPTS}(p) := 1/\left(\frac{\sum_{o \in N_{minPTS}(p)} reach - dist_{minPTS}(p,o)}{|N_{minPTS}(p)|}\right) \tag{2.1}$$

Daraus berechnet sich der lokale Ausreißer-Faktor ("local outlier factor", "lof") für den Punkt p. Hierzu werden die Verhältnisse der lokalen Erreichbarkeitsdichte in der k-Nachbarschaft zur eigenen Erreichbarkeitsdichte aufsummiert und durch die Anzahl der Punkte in der k-Nachbarschaft geteilt. Objekte, die in einem Cluster liegen, besitzen einen lokalen Ausreißer-Faktor von ungefähr 1. Je weiter ein Objekt von anderen Objekten entfernt ist, umso größer wird dieser Wert.

$$LOF_{minPTS}(p) := \frac{\sum_{o \in N_{minPTS}(p)} \frac{lrd_{minPTS}(o)}{lrd_{minPTS}(p)}}{|N_{minPTS}(p)|} \tag{2.2}$$

2.2.4 ROCK

ROCK (A Robust Clustering Algorithm for Categorical Attributes) ist ein von S. Guha, R. Rastogi und K. Shim [GRS99] entwickeltes hierarchisches Clustering-Verfahren. ROCK fügt einzelne Cluster auf Basis ihrer Gemeinsamkeiten (Links) zusammen. Die Links werden dabei durch eine Ähnlichkeitsfunktion berechnet. Die Ähnlichkeitsfunktion $sim(p_i, p_j)$ gibt an, wie ähnlich die beiden Punkten p_i und p_j sind. Hierbei können metrische Funktionen, wie die L1- oder L2-Norm verwendet werden. Um jedoch mehr Domänenwissen in den Clustering-Prozess zu bringen, ist es auch möglich eine nicht metrische Funktion, die von Experten erzeugt wurde, zu verwenden [GRS99].

Die verwendete Bibliothek pyclustering [Nov19] nutzt für die Ähnlichkeitsfunktion die L2-Norm (Euklidische Norm). Bei dieser Implementierung ist die Verwendung einer anderen Ähnlichkeitsfunktion nicht möglich. Zudem muss dem Verfahren der Parameter ϵ übergeben werden, welcher angibt, ab welchem Schwellenwert der L2-Norm zwei Punkte als ähnlich zu betrachten sind. Zusätzlich benötigt das Verfahren einen weiteren Parameter k. Dieser gibt an, in wie viele Cluster der eingegebene Datensatz zerlegt werden soll. Ein weiteres Problem tritt auf, wenn die einzelnen Cluster nicht sauber voneinander getrennt sind. Hier kann es passieren, dass große Cluster naheliegende kleinere Cluster schlucken, da die großen Cluster auf Grund der großen Anzahl an Datenpunkte mehr Links zu diesem Cluster besitzt [GRS99]. Kleinere Cluster mit weniger Links werden so nicht zusammengefügt, obwohl diese möglicherweise direkt nebeneinander liegen. Um dieses Problem zu lösen, wird eine Güte-Funktion definiert, die das Zusammenlegen von großen Clustern mit Hilfe eines Multiplikator θ bestraft.

Im ersten Schritt des Algorithmus wird jeder Datenpunkt in einem eigenen Cluster gelegt. Nun werden die Links zwischen allen Clustern berechnet. Auf Basis der Anzahl der Links zwischen zwei Clustern wird der Güte-Wert für alle Cluster-Paare berechnet. Anschließend wird das Paar mit dem größten Güte-Wert zu einem Cluster zusammengefügt. Dieser Schritt wird so lange wiederholt, bis nur noch k Cluster übrig sind.

2.2.5 Autoencoder

Autoencoder kommen aus dem Bereich des maschinellen Lernens. Sie sind eine spezielle Art von neuronalen Netzen. Ein Autoencoder besteht aus zwei separaten Netzwerken. Es gibt die Encoder-Seite, die lernt einen Datenpunkt in eine andere Repräsentation mit weniger Dimensionen zu reduzieren. Die Decoder-Seite hingegen wird trainiert aus dieser reduzierten Repräsentation wieder die ursprünglichen Daten zu erzeugen. Autoencoder wurden auch im Bereich der Bereinigung von Signalrauschen und bei der Erkennung von Ausreißern eingesetzt [HHWB02, SSB$^+$18, ST20].

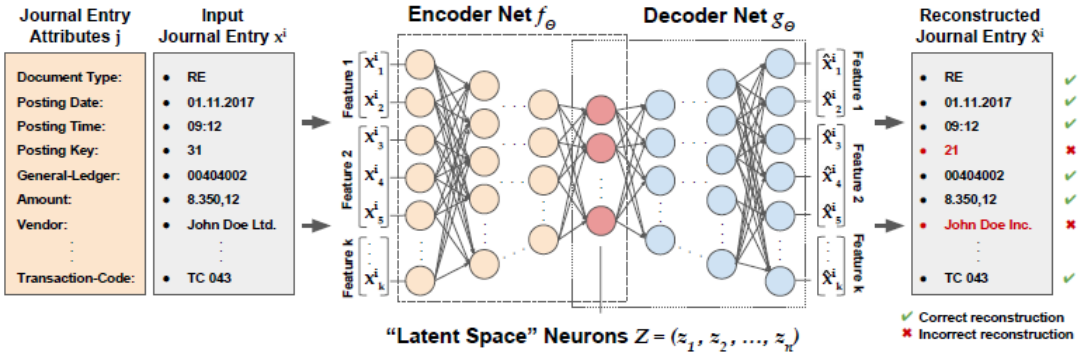

Abbildung 2.2: Aufbau und Funktionsweise eines Autoencoder [SSB+18]

Die Verwendung von Autoencoder in der Ausreißer-Erkennung wird anhand der Abbildung 2.2 erläutert. Jeder Datenpunkt wird nacheinander in den Autoencoder übergeben. Die Datenpunkte sind der Input für den Encoder (links) und werden dann durch die folgenden Ebenen immer weiter in der Dimension reduziert. Das Ergebnis (in der Abbildung rot) ist eine reduzierte Repräsentation des ursprünglichen Datenpunktes. Das Ergebnis vom Encoder ist der Input für den Decoder (rechts). Dieser versucht den Datenpunkt so wiederherzustellen, wie er in den Encoder gegeben wurde.

Der Autoencoder ist in diesem Fall ein unüberwachtes Verfahren und wird deshalb mit allen verfügbaren Datenpunkten trainiert. Da ein Großteil der Datenpunkte in dem Datensatz gültige Datenpunkte sind, welche meist festen Regeln folgen, lernt der Autoencoder diese Datenpunkte am besten zu reduzieren und zu expandieren. Wenn nun ein Datenpunkt in den Autoencoder gegeben wird, der nicht diesen Regeln folgt, so hat der Autoencoder Schwierigkeiten diesen Datenpunkt korrekt zu rekonstruieren, wie in Abbildung 2.2 verdeutlicht wird. Der Unterschied zwischen der Eingabe und der Rekonstruktion wird als Rekonstruktionsfehler betrachtet und berechnet sich aus der durchschnittlichen quadrierten Differenz jedes Attributes.

Zu den notwendigen Parametern für dieses Verfahren gehört die Struktur des Netzwerkes. Dazu zählen die Anzahl der Schichten, die Größe jeder Schicht und deren Aktivierungsfunktion. Ebenso muss definiert werden, für wie viele Epochen trainiert wird.

3 Datenvorverarbeitung

Das folgende Kapitel befasst sich mit der Datenvorverarbeitung des Datensatzes. In Abschnitt 3.1 wird der verwendete Datensatz im Bezug auf seiner Herkunft, seiner Attribute und seiner Struktur beschrieben. Darauffolgend wird in Abschnitt 3.2 der Datensatz analysiert. Auf Basis der Erkenntnisse aus dem Abschnitt 3.2 werden im Abschnitt 3.3 die Attribute ausgewählt, welche dann in den Abschnitten 3.4 und 3.5 verarbeitet werden. Abschließend wird im Abschnitt 3.6 das Resultat der Datenvorverarbeitung analysiert.

Die Verarbeitung der Daten ist ein wichtiger und notwendiger Schritt, da ein Großteil aller Attribute aktuell nur in Form von Zeichenketten vorliegen. Arithmetische Operationen auf Zeichenketten sind entweder nicht möglich oder nicht aussagekräftig. Die Differenz bzw. der Abstand zwischen zwei Zeichenketten kann nicht sinnvoll berechnet werden. Des Weiteren können die in Abschnitt 2.2 vorgestellten Verfahren nicht auf Zeichenketten arbeiten. Diese benötigen numerische Werte für die Berechnung. Das Ziel der Datenvorverarbeitung ist die sinnvolle Umwandlung der Zeichenketten in numerische Werte, ohne deren Bedeutung zu verlieren.

Die Datenvorverarbeitung wird auf jedem Konto einzeln und isoliert von den Anderen vorgenommen, da die Analyse im Abschnitt 3.2 ergeben hat, dass einige Ausprägungen der Attribute nur auf einem Konto vorkommen und das Konten unterschiedliche Charakteristiken aufweisen. Das Ziel ist eine bestmögliche Datenvorverarbeitung für jedes Konto und gleichzeitig unnötige Vergrößerungen der Dimensionen zu verhindern.

3.1 Struktur und Aufbau des Datensatzes

Bei dem in dieser Arbeit verwendete Datensatz handelt es sich um einen Datensatz, der aus dem SAP ERP System eines Unternehmens entnommen wurde und umfasst alle Buchungseinträge aus einem Fiskaljahr.

Der gesamte Datensatz liegt in Form einer CSV-Datei vor. Jede Zeile innerhalb dieser Datei repräsentiert eine Belegposition eines Beleges. Weder die Konten noch die Belege werden direkt beschrieben, diese ergeben sich indirekt aus den Informationen der Belegpositionen. Jede Belegposition wird dabei mit Hilfe des Attributs *DocNo* einem Beleg und durch das Attribut *accountno* bzw. *accountname* einem Konto zugeordnet. Die Position einer Belegposition auf einem Beleg wird durch das Attribut *DocPos* beschrieben. Eine Auflistung aller Attribute mit ihrer Bedeutung ist im Anhang A.1 zu finden.

Aufgrund von Datenschutzbestimmungen wurde ein großer Teil aller Attribute durch eine Einweg-Hashfunktion anonymisiert. Von der Anonymisierung sind folgende Attribute nicht betroffen: *UserGroup, RevenueIndicator, AmountLocalCurrency, DebitCreditIndicator, Currency, CreationDateDayOfWeek, CreationDate, DocDate, PostingDate, DocPos, AccountingPeriod, AccountingYear*. Obwohl diese Attribute anonymisiert wurden, ist es sinnvoll diese zu verwenden, da weniger der Name eines Kontos oder eines Benutzers aussagekräftig ist, sondern nur die Informationen welche Buchungspositionen miteinander in Verbindung stehen. Aus nicht gehashten Attributen könnten fachbezogene Informationen extrahiert werden, wenn Expertenwissen vorhanden wäre. Dies ist hier aber nicht der Fall.

Des Weiteren weist der Datensatz einige Attribute auf, die als Ausprägungen Listen beinhalten. Dies sind die folgenden Attribute *debit_ accountno_ list, debit_ accountname_ list, credit_ accountno_ list, credit_ accountname_ list, CredDebNumberList, creditornamelist, debitornameList, DebCredCountryList*. Die Entstehung der Listen wird im Folgenden am Beispiel von *debit_ accountno_ list* erläutert. Zu einem Beleg wurden alle Kontonummern der Belegposition extrahiert, die auf der Soll-Seite (Debit-Seite) des Belegs gebucht wurden. Diese Menge wurden dann von Duplikaten bereinigt und anschließend bei allen Belegpositionen als zusätzliches Attribut *debit_ accountno_ list* in Form einer Liste gespeichert. Aus der Liste lassen sich trotzdem die einzelnen anonymisierte Kontonummern extrahieren.

Zusätzlich zum eigentlichen Datensatz wurden drei Konten von Wirtschaftsprüfern analysiert. Diese Konten wurden ausgewählt, da diese ein einheitliches Buchungsmuster aufweisen und hauptsächlich von standardisierten und automatisierten Geschäftsprozessen verwendet werden [ST20]. Zu jedem Konto wurden alle Belege markiert, die, nach Meinung der Wirtschaftsprüfer, auffällig sind.

3.2 Deskriptive Analyse des Datensatzes

Der gesamte Datensatz besteht aus 302.365 Belegpositionen und 72.917 Belegen, die sich auf 321 Konten aufteilen. Zu jeder Belegposition gibt es 42 Attribute.

Kontoname (DE)	Kontoname (EN)	Belege	Positionen	Auffällige Positionen
Umsatzerlöse EU	Revenue Foreign	651	651	26
Verbrauch Verpackungen	Expenses	777	778	27
Umsatzerlöse Inland	Revenue Domestic	6643	6643	241

Tabelle 3.1: Überblick über die drei Konten

In Tabelle 3.1 sind die drei Konten zu sehen, die bereits in Abschnitt 3.1 beschrieben wurden. Es ist zu erkennen, wie viele Belege, Belegpositionen und auffällige Belegpositionen auf einem Konto vorhanden sind. Bei genauerer Betrachtung fällt auf, dass es auf dem Konto Verbrauch Verpackung einen Beleg gibt, der zwei Buchungen auf demselben Konto aufweist. Dies ist problematisch, da die Wirtschaftsprüfer nur die auffälligen Belege und nicht die Belegpositionen markiert haben. Da nicht bestimmt werden kann, ob nur eine oder beide Belegpositionen auffällig sind, wurden beide Belegpositionen als Ausreißer behandelt.

Als erster Schritt wurde der Datensatz auf fehlende Werten hin untersucht. Fehlende Werte können durch korrumpierte Datensätze oder durch Fehler beim Aufzeichnen der Datenpunkte entstehen. Dieses Problem wird meist gelöst, indem der Datenpunkt (die Zeile) oder das Feature (die Spalte) gelöscht werden. Im Datensatz ließen sich bei den Attributen *taxcode*, *DocNotes_Detailed*, *DocPosNotes_Detailed*, *CredDebNumberList*, *creditornamelist*, *debitornameList*, *DebCredCountryList*, *recurringDocNo*, *reversalDocNo* fehlende Werte finden. In Absprache mit einem Experten konnte ermittelt werden, dass diese Werte auf die Anwendungsdomäne zurückzuführen sind und somit gültige Werte darstellen. Eine Bereinigung dieser Datenpunkte war deshalb nicht sinnvoll, da sonst die Bedeutung dieser Attribute verfälscht werden würde.

Entgegen dem üblichen Vorgang konnte der Datensatz nicht auf Plausibilität überprüft werden, da zum einen viele Attribute anonymisiert wurden und zum anderen, das notwendige Wissen über die dazugehörenden Geschäftsprozesse fehlte.

Im Folgenden werden die einzelnen Attribute und deren Ausprägungen analysiert. Hierbei beschränkt sich die Analyse ausschließlich auf die drei Konten aus Tabelle 3.1. Da für die restlichen Konten keine Daten der Wirtschaftsprüfer vorliegen, werden diese von nun an nicht weiter betrachtet.

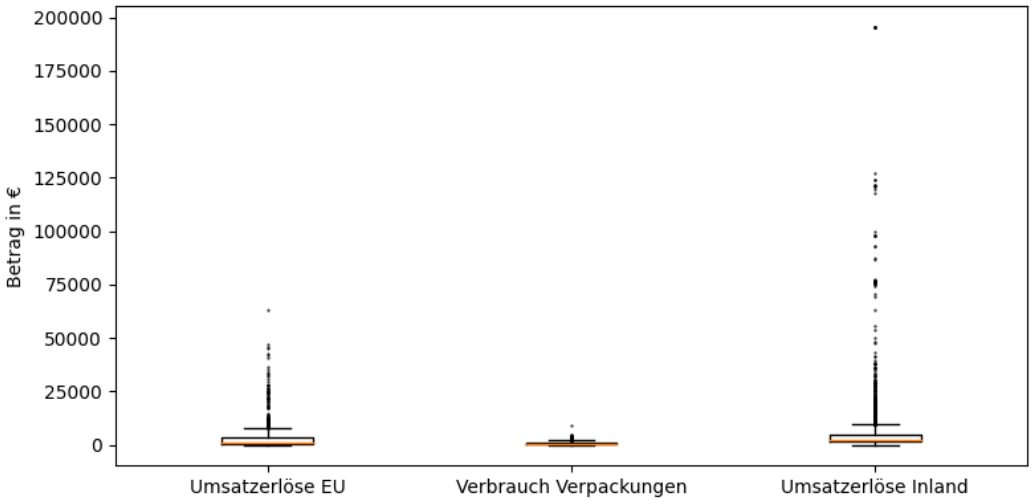

Abbildung 3.1: Überblick über die Verteilung des Attributs Betrag

Als erstes wird das Attribut *AmountLocalCurrency* betrachtet. Dieses Attribut ist besonders relevant, da es das einzige metrische Attribut in dem Datensatz ist und da es eine große Bedeutung in der Wirtschaftsprüfung besitzt. In Abbildung 3.1 werden die Höhe der Buchungspositionen durch drei Boxplots dargestellt. Es ist zu erkennen, dass der Betrag auf den drei Konten völlig unterschiedliche Charakteristiken aufweist. Auf dem Konto *Umsatzerlöse EU* liegt die Höhe der Transaktion zwischen 6,27€ und 63.235,72€ mit einer durchschnittlichen Höhe von 3.914,29€. Das Konto *Verbrauch Verpackungen* weist viel kleinere Transaktionen zwischen 0,05€ und 9.112,45€ auf. Die durchschnittliche Transaktionshöhe liegt hier bei 779,38€. Das Konto *Umsatzerlöse Inland* weist ein ganz anders Verhaltensmuster auf. Die größte Transaktion umfasst hier 195.624,00€ und auch die durchschnittliche Transaktion liegt mit 4.435,59€ wesentlich höher.

Eine Korrelationsanalyse hat ergeben, dass einige Attribute eine starke Korrelation zueinander aufweisen. Hierbei handelt es sich um Attribute, die sowohl als ID, als auch als Langname vorhanden sind. Als Beispiel sind die Attribute *accountno* und *accountname* zu nennen. Diese repräsentieren denselben Sachverhalt. Ein Sonderfall bilden die beiden Attribute *DocPosNotes* und *DocPosNotes_ Detailed*. *DocPosNotes_ Detailed* enthält

die gehashte Anmerkung zur Belegposition. Dadurch kommt aber jede Ausprägung, mit Ausnahme der leeren Ausprägung, nur genau ein einziges Mal vor. Das binäre Attribut *DocPosNotes* gibt dabei nur an, ob eine Anmerkung vorhanden ist oder nicht. Die Attribute *DocNotes* und *DocNotes_ Detailed* weisen dasselbe Verhalten auf. Das Attribut *reversalDocNo* besteht zu 97,65 Prozent aller Fälle aus leeren Werten. In den restlichen Fällen kommt jede Ausprägung nur einmal vor.

Des Weiteren gibt es diverse Attribute, die auf allen drei Konten nur eine Ausprägung besitzen. Dazu gehören *AccountingYear, accounttype, Currency, RevenueIndicator, UserGroup, DocNotes, DocNotes_ Detailed, recurringDocNo*. Zusätzlich gibt es Attribute, die auf einem Konto verschiedene Ausprägungen aufweisen, jedoch auf mindestens einem anderen Konto nur eine Ausprägung besitzen. Die betreffenden Attribute lauten *OneDateOutsideAccountYear, creationBeforPosting, taxcode, userid, tcodename, reversalDocNo, debitornameList*.

3.3 Auswahl geeigneter Attribute

Im ersten Schritt werden alle Attribute verworfen, die auf allen drei Konten nur eine Ausprägung besitzen, da diese keine Information tragen und nur dafür sorgen würden, dass sich die Anzahl der Dimensionen erhöht. Attribute, die bei der Betrachtung eines einzelnen Kontos nur eine Ausprägung aufweisen, werden für dieses Konto verworfen. Diese werden für die anderen Konten trotzdem verwendet. Wenn zwei Attribute denselben Sachverhalt repräsentieren, wird einer der beiden Attribute verworfen. Welches davon verworfen wird, ist auf Grund der Substituierbarkeit nicht relevant. Wenn beide Attribute verwendet werden würden, hätte dies zu Folge, dass sich die Distanzen zwischen zwei verschiedenen Ausprägungen verdoppelt. Dies ist besonders für distanzbasierte Verfahren problematisch, da diesen Attributen eine höhere Gewichtung zugewiesen werden würde, ohne, dass dies durch die Domaine gerechtfertigt ist. Bei den Attributen, dessen Ausprägungen nur einmal vorkommen, gibt es bereits ein binäres Attribut, das aussagt, ob ein Wert vorhanden ist oder nicht. Das eigentliche Attribut kann verworfen werden, da durch das einmalige auftauchen einer Ausprägung keine Ähnlichkeit zwischen den Belegpositionen bestimmt werden kann. Es erfolgt die Reduktion auf das binäre Attribut. Alle binären Attribute, die nicht bereits auf Grund der oben genannten Fälle entfernt wurden, bleiben bestehen.

Der Datensatz enthält die beiden Attribute *DocNo* und *Accountno*. Diese dienen dazu die Belege zu Identifizieren oder eine Belegposition einem Konto zuzuordnen. Da ein Beleg auf einem Konto nur einmal vorkommt, mit Ausnahme eines Sonderfalls vgl. Abschnitt 3.2, bietet es keinen Mehrwert. Ebenso mit dem Attribut *Accountno*, da es definitionsbedingt nur eine Ausprägung pro Konto annimmt. Es ist somit redundant und brauch nicht weiter berücksichtigt werden.

Das Attribut *DocPos* wird für die weitere Datenvorverarbeitung nicht weiter berücksichtigt, da im Gespräch mit einem Experten dieses Attribut für nicht relevant empfunden wurde. Die Position einer Belegposition auf einem Beleg ist zufällig bzw. hat keine Aussagekraft über die Belegpositionen selbst. Zudem werden die Attribute mit Datumsangabe *PostingDate*, *DocDate* und *CreationDate* nicht verwendet, da diese bereits in die Attribute *CreationDateDayOfWeek*, *DatesEqual*, *OneDateOutsideAccountYear*, *postingCloseToFiscalYearEnd* und *creationBeforPosting* umgewandelt wurden.

3.4 Verarbeitung kategorialer Attribute

Für die Kodierung kategorialer Attribute gibt es verschiedene Ansätze. Eine Möglichkeit ist die Label-Kodierung oder auch ordinal encoding genannt. Hierbei wird jeder der k Ausprägung eine natürliche ganze Zahl von *0* bis *k-1* zugeordnet. Diese werden dann in einer Dimension abgebildet. Das Problem bei dieser Kodierung ist, dass den Attributen A, B, C eine Ordnung gegeben wurde und so die Distanz zwischen A und B kleiner ist als die Distanz zwischen A und C. Die drei Datenpunkte sind in Abbildung 3.2 dargestellt. Dies würde besonders bei distanzbasierten Verfahren erhebliche Nachteile bringen.

Eine Alternative ist die Dummy Kodierung. Hierbei werden k Attribute in *k-1* Dimensionen aufgeteilt. Jede Dimension beschreibt so, ob diese Ausprägung vorhanden ist oder nicht. Die Dimension der k-ten Ausprägung wird dabei weggelassen, um Redundanzen in den Daten zu reduzieren. Wenn jede andere Dimension angibt, dass es diese Ausprägung nicht ist, so ist daraus zu schließen, dass es die k-te Ausprägung sein muss. Der Abstand aller Ausprägungen, mit Ausnahme der k-ten Ausprägung ist nun $\sqrt{2}$. Der Abstand der k-ten Ausprägung zu allen anderen ist jedoch 1.

Eine weitere Möglichkeit ist die One-Hot Kodierung. Diese ist ähnlich zur Dummy Kodierung, jedoch werden hier Redundanzen und eine weitere Dimension in Kauf genommen.

Dies sorgt für mehr Dimensionen, jedoch wird verhindert, dass eine Ausprägung näher an anderen liegt. Der Abstand zwischen allen Ausprägungen ist somit immer $\sqrt{2}$.

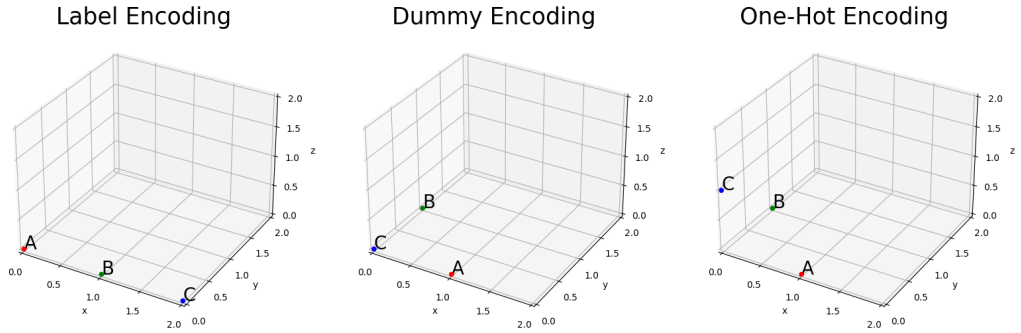

Abbildung 3.2: Vergleich verschiedener Kodierungsansätz

Auf Grund der Tatsache, dass ein Großteil der Verfahren die Distanzen zwischen Datenpunkten berücksichtigt, wird die One-Hot Kodierung verwendet. Diese Kodierung wird auch bei binären Attributen verwendet. Dies hat einen Anstieg der Dimensionen zur Folge und führt zu schlechterem Laufzeitverhalten der Verfahren, jedoch wird sichergestellt, dass die Ausprägungen aller Attribute die gleiche Distanz zueinander aufweisen. Alle Attribute, die aus Listen bestehen, werden wie die kategorialen Attribute behandelt und somit auch One-Hot kodiert.

3.5 Verarbeitung des metrischen Attributs

AmountLocalCurrency ist das einzige metrische Attribut in dem Datensatz und besitzt in der Wirtschaftsprüfung eine besondere Bedeutung. Im vorherigen Abschnitt 3.2 wurde erkannt, dass dieses Attribut auf den einzelnen Konten ein völlig unterschiedliches Verhalten aufweist (vgl. Abbildung 3.1).

Für die Umwandlung des Attributes *AmountLocalCurrency* besteht die Möglichkeit den Wertebereich zu normalisieren. Hierbei wird das Minimum als 0,0 und das Maximum als 1,0 definiert und alle weiteren Beträge ordnen sich in diesem Bereich ein. Für distanzbasierte Verfahren ist dieser Ansatz jedoch nicht optimal, da der Abstand zwischen den Ausprägungen höchstens 1,0 und häufig sogar noch niedriger ist. Der Einfluss dieses Attributes ist somit verschwindend gering, da ein Unterschied bei einem anderen Attribut schon eine zusätzliche Distanz von $\sqrt{2}$ erzeugt.

Als Alternative zur Normalisierung wird das metrische Attribut in ein kategoriales Attribut umgewandelt. Hierzu werden die einzelnen Ausprägungen in Gruppen eingeteilt. Als Grenze dient der Durchschnitt inklusive eines Vielfachen der Standardabweichung. Da das Erkennen von auffälligen Transaktionen mit höheren Beträgen wichtiger ist als kleinere Transaktionen zu erkennen, werden Beträge unter dem Durchschnitt in größere Gruppen eingeteilt. Der Fokus wird somit auf die größeren Beträge gerichtet. Die untere Grenze wurde auf 0 und die obere Grenze auf ∞ gesetzt. Als Vielfache der Standardabweichung wurden -1, 0, 0,5, 1, 1,5, 2, 2,5, 3 und 3,5 verwendet. Anschließend wurden die Gruppen, wie in Abschnitt 3.4, One-Hot kodiert.

3.6 Ergebnis der Datenvorverarbeitung

Im Folgenden wird ein Überblick über das Ergebnis der Datenvorverarbeitung gegeben. Jedes Konto wird im Bezug der verwendeten Attribute, der entstandenen Dimensionen, sowie die Anzahl der Distanzen und deren Verteilung beschrieben. Diese Informationen werden benötigt, um die Parameter der Verfahren im Kapitel 4 optimal zu bestimmen. Die Verteilung aller Distanzen im Detail kann der Tabelle im Anhang A.2 entnommen werden.

Umsatzerlöse EU

Für das Konto *Umsatzerlöse EU* wurden 17 Attribute verarbeitet und es entstanden 119 Dimensionen. Bei der Berechnung der Distanzen zwischen allen Datenpunkten traten 16 verschiedene Werte auf. Die genaue Verteilung ist in Abbildung 3.3 zu erkennen. Die Distanzen lagen zwischen 0,00 und 5,48, wobei der Mittelwert bei 3,40 und die Standardabweichung bei 0,61 liegt.

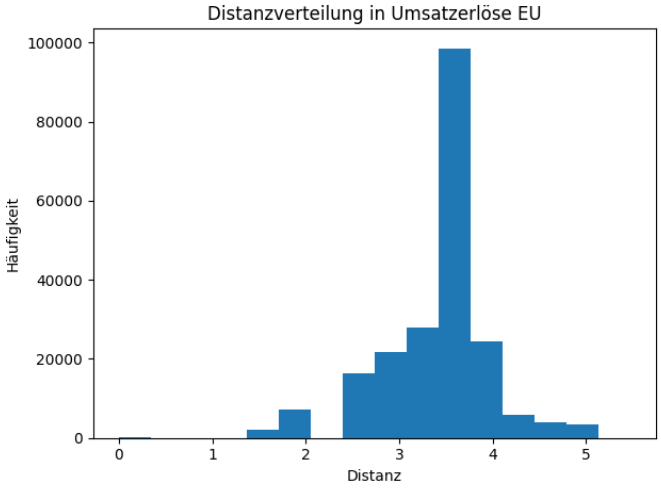

Abbildung 3.3: Distanzverteilung in Umsatzerlöse EU

Verbrauch Verpackungen

Für das Konto *Verbrauch Verpackungen* wurden 16 Attribute verarbeitet und es entstanden 72 Dimensionen. Bei der Berechnung der Distanzen zwischen allen Datenpunkten traten 17 verschiedene Werte auf. Die genaue Verteilung ist in Abbildung 3.4 zu erkennen. Die Distanzen lagen zwischen 0,00 und 5,66, wobei der Mittelwert bei 2,49 und die Standardabweichung bei 0,57 liegt. Auffällig ist hier, dass das Konto *Verbrauch Verpackungen* viel weniger Dimensionen als das Konto *Umsatzerlöse EU* besitzt, obwohl ungefähr gleich viele Transaktionen auf dem Konto stattgefunden haben. Aufgrund der geringen Anzahl an Dimensionen ist zu vermuten, dass dieses Konto viel strikteren Regeln folgt als die restlichen Konten.

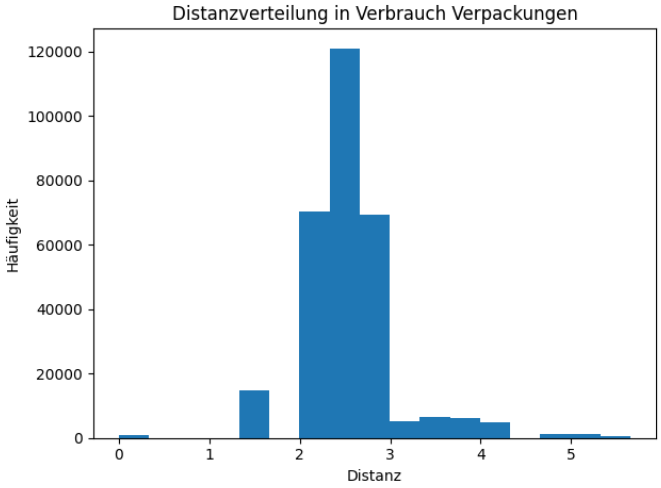

Abbildung 3.4: Distanzverteilung in Verbrauch Verpackungen

Umsatzerlöse Inland

18 Attribute wurden für das Konto *Einnahmen im Inland* verarbeitet. Jedoch hatten die einzelnen Attribute wesentlich mehr Ausprägungen. Dies, in Kombination mit fast zehn Mal so vielen Transaktionen, sorgt für insgesamt 251 Dimensionen. Es entstanden insgesamt 17 einzigartige Distanzen, die im Bereich 0,00 bis 5,66 liegen. Der Mittelwert liegt bei 3,47 und die Standardabweichung bei 0,53. Die genaue Verteilung ist in Abbildung 3.5 zu erkennen.

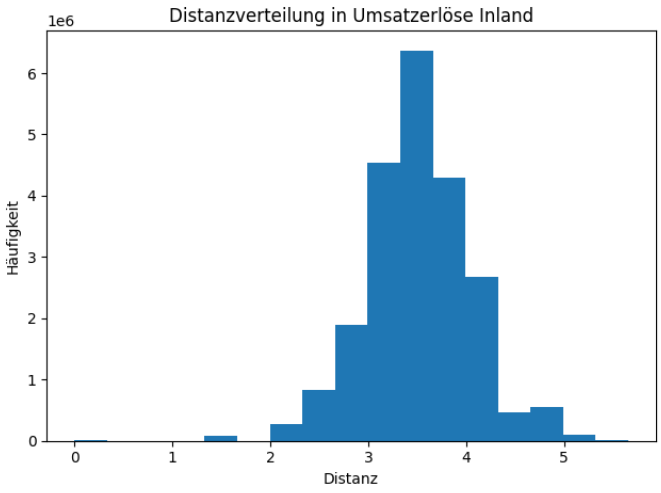

Abbildung 3.5: Distanzverteilung in Umsatzerlöse Inland

4 Erprobung der Verfahren

In diesem Kapitel wird zuerst im Abschnitt 4.1 das entwickelte Testsystem und die Metriken, anhand derer die Verfahren bewertet werden, beschrieben. In den Abschnitten 4.2, 4.3, 4.4, 4.5 und 4.6 werden die einzelnen Verfahren auf allen drei Konten erprobt. Zum Ende dieses Kapitel werden die erreichten Leistungen zusammengefasst.

4.1 Vorgehensweise und Testaufbau

Das Testsystem wurde in der Programmiersprache Python implementiert. Für jeden Teilbereich wurde ein eigenes Modul geschrieben. Es gibt je ein Modul für das Einlesen des Datensatzes und der Ausreißer, für die Analyse des Datensatzes, für die Datenvorverarbeitung, sowie für die Erprobung der Verfahren. Jedes Verfahren wird in einem eigenen Teilmodul angebunden und besitzt eine analoge Schnittstelle. Das Modul für die Datenvorverarbeitung ist modular aufgebaut, so kann jedes Attribut individuell vorverarbeitet werden. Dies hat die explorative Erprobung vereinfacht. Alle Ergebnisse laufen zentral in einem Modul zusammen und ermöglichen einen tieferen Einblick in die Ergebnisse der Verfahren.

Für die Erprobung von DBSCAN wurde die Software-Bibliothek Scikit-learn [PVG+11] verwendet. Alle Datenpunkte, die DBSCAN einem Cluster zuordnet, werden als unauffällige Datenpunkte markiert. Rauschpunkte werden als Ausreißer markiert.

Für OPTICS wird ebenfalls die Software-Bibliothek Scikit-learn [PVG+11] verwendet. Um OPTICS für die Ausreißer-Erkennung zu verwenden, wird ein weiterer Parameter *maxDistance* eingeführt. Alle Datenpunkte, die eine Erreichbarkeitsdistanz von mindestens *maxDistance* aufweisen, werden als Ausreißer betrachtet, da diese am weitesten von den gebildeten Clustern entfernt sind.

Die Bibliothek PyOD [ZNL19] enthält eine Implementierung für LOF, die für die Analyse verwendet wurde. Um die Datenpunkte binär als auffällig oder unauffällig zu klassifizieren, wird ein weiterer Parameter *contamination* eingeführt. Dieser gibt an, wie groß der zu erwartende Anteil aller Ausreißer am gesamten Datensatz ist. Hierbei werden die *n_ samples * contamination* Datenpunkte mit dem größten LOF-Wert als Ausreißer markiert.

Da ROCK ähnliche Datenpunkte in einen Cluster legt, sammeln sich die unauffälligen Datenpunkten in einzelnen großen Clustern. Die auffälligen Datenpunkten, die weniger Ähnlichkeiten mit den restlichen Datenpunkten besitzen, sammeln sich entweder allein in einem Cluster oder bilden selbst sehr kleinere Cluster. Mithilfe eines weiteren Parameters *sizethreshold* werden nun alle Datenpunkte eines Clusters, der kleiner als der Schwellwert ist, als Ausreißer betrachtet. Für ROCK wird die Implementierung aus der pyclustering Bibliothek [Nov19] verwendet.

Eine Implementierung des Autoencoders ist ebenfalls in der PyOD Bibliothek [ZNL19] zu finden. Analog zu LOF wird hier auch ein Parameter *contamination* eingeführt. Hier werden die *n_ samples * contamination* Datenpunkte mit dem größten Rekonstruktionsfehler als Ausreißer markiert.

Zuerst wird für jedes Verfahren eine Erwartung definiert, diese ergibt sich aus der Funktionsweise des Verfahrens, bisherigen Anwendungen der Verfahren und der Analyse des Datensatzes nach der Datenvorverarbeitung aus Abschnitt 3.2. Hier wird auch auf die Vor- und Nachteile der einzelnen Verfahren eingegangen.

Anschließend wird die Wahl der Parameter beschrieben. Hiermit soll verhindert werden, dass die Parameter willkürlich und ohne nachvollziehbare Grundlage ausgewählt werden. Für die Wahl der Parameter wird sowohl die Funktionsweise der Verfahren aus Abschnitt 2.2, als auch die Analyse der einzelnen Konten aus Abschnitt 3.2 verwendet. Sollte ein Parameter nicht eindeutig hergeleitet werden, so wird versucht den möglichen Wertebereich einzugrenzen.

Die Verfahren geben für eine Transaktion an, ob diese in die Gruppe der Outlier oder Inliner fällt. Der Wert True-Positives gibt an, wie viele Transaktionen korrekt als Ausreißer markiert wurden. Transaktionen, die fälschlicherweise als Ausreißer markiert wurden, werden als False-Positives gezählt. True-Negatives sind die Anzahl der Transaktionen, die richtig als Inliner markiert wurden und False-Negatives sind Transaktionen, die zwar

Ausreißer sind, jedoch vom Verfahren nicht als solche erkannt wurden. Hieraus lassen sich weitere Metriken berechnen, um die Leistung eines Verfahrens bewerten zu können.

Die Genauigkeit oder auch Accuracy gibt an, wie groß der Anteil der korrekt eingeordneten Transaktionen an der Gesamtmenge aller Transaktionen ist. Die Berechnung erfolgt dabei wie folgt:

$$Accuracy = \frac{TruePositives + TrueNegatives}{TruePositives + FalsePositives + TrueNegatives + FalseNegatives} \tag{4.1}$$

Die Trefferquote oder auch Recall gibt an, wie groß der Anteil der korrekt erkannten Ausreißer an der Anzahl aller zu erkennenden Ausreißer ist. Der Recall-Wert berechnet sich wie folgt:

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{4.2}$$

Der Anteil der korrekt erkannten Ausreißer an der Anzahl aller markierter Ausreißer wird als Genauigkeit oder Precision beschrieben und berechnet sich durch die folgende Formel:

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{4.3}$$

Das F-Maß bzw. F-Score kombiniert die Genauigkeit und die Trefferquote mit Hilfe des harmonischen Mittels. Beim F_1-Score werden beide Verhältniszahlen gleich gewichtet. Der F_1-Score berechnet sich dabei wie folgt:

$$F_1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4.4}$$

Zusätzlich zu diesen messbaren Metriken wird auch die Stabilität der Verfahren berücksichtigt. Als Stabilität wird die Empfindlichkeit eines Verfahrens im Bezug auf die Wahl der Parameter beschrieben. Verfahren mit einer hohen Stabilität erreichen trotz unterschiedlicher Parameterwahl gleiche Resultate.

4.2 Erprobung von DBSCAN

Erwartung

Bei DBSCAN handelt es sich zwar um ein dichtebasiertes Clustering-Verfahren, jedoch kann DBSCAN keine lokal unterschiedlichen Dichten erkennen. Ein Cluster kann aus sehr weit entfernten Datenpunkten bestehen, ein Anderer wiederum aus sehr nahen Datenpunkten. Da ϵ und *minPTS* global definiert werden, besteht die Gefahr, dass einige Cluster nicht korrekt erkannt werden. Es ist aber zu erwarten, dass DBSCAN einige weit entfernte Ausreißer erkennt.

Wahl der Parameter und Anpassungen

Da DBSCAN keine lokal unterschiedlichen Dichten erkennen kann, werden kleinere Werte für ϵ gewählt, um die dichtesten Cluster zu erkennen. Hierfür werden die Distanzen aus Tabelle A.2 genommen, die kleiner als der Mittelwert abzüglich der einfachen Standardabweichung sind. Um mögliche Fehler durch Fließkommaarithmetik zu verhindern, werden Werte zwischen diesen Distanzen verwendet. Daraus folgen die Werte 1,0, 1,7, 2,2 und 2,6 für ϵ auf dem Konto *Umsatzerlöse EU*. Nach demselben Verfahren ergeben sich die Grenzwerte für das Konto *Verbrauch Verpackung* von 1,0, 1,7 und 2,2 und für das Konto *Umsatzerlöse Inland* von 1,0, 1,7, 2,2 und 2,6. Für *minPTS* wird auf allen Konten 4 und 10 verwendet. Hiermit soll untersucht werden, ob DBSCAN bessere Leistungen erzielt, wenn nach kleineren oder größeren Clustern gesucht wird.

Erreichte Leistungen

Die erreichten Ergebnisse von DBSCAN auf dem Konto *Umsatzerlöse EU* finden sich in der Tabelle 4.1 wieder. Bei einem ϵ-Wert von 1,0 und einem *minPTS*-Wert von 4 bilden sich nur zwei Cluster, die die dichtesten neun Punkte beinhalten. Alle anderen Transaktionen wurden als Rauschen identifiziert. Bei einem *minPTS*-Wert von 10 bildetet sich kein einziger Cluster und alle Transaktionen wurden als Ausreißer markiert. Bei einem ϵ-Wert von 1,7 und 2,2 wurden mehr bedeutungsvollere Cluster gebildet und ergaben bessere Werte für die Accuracy und den F-Score. Die besten Ergebnisse wurden mit einem ϵ-Wert von 2,6 erreicht. Hier bildeten sich in beiden Fällen zwei Cluster.

ϵ	1,0		1,7		2,2		2,6	
minPTS	4	10	4	10	4	10	4	10
True-Negatives	9	0	509	326	606	586	616	616
True-Positives	26	26	23	26	18	19	17	17
False-Negatives	0	0	3	0	8	7	9	9
False-Positives	616	625	116	299	19	39	9	9
Accuracy	0,05	0,04	0,82	0,54	0,96	0,93	0,97	0,97
Recall	1,00	1,00	0,88	1,00	0,69	0,73	0,65	0,65
Precision	0,04	0,04	0,17	0,08	0,49	0,33	0,65	0,65
F-Score	0,08	0,08	0,28	0,15	0,57	0,45	0,65	0,65

Tabelle 4.1: Ergebnisse von DBSCAN auf dem Konto Umsatzerlöse EU

Die Tabelle 4.2 enthält die Ergebnisse für das Konto *Verbrauch Verpackungen*. Bei einem ϵ-Wert von 1,0 und einem *minPTS*-Wert von 4 bildeten sich 59 Cluster, jedoch wurden hier zu viele Transaktionen falsch als Ausreißer markiert. Die besten Ergebnisse wurden bei einem ϵ-Wert von 1,7 erreicht. Ein höherer Wert für *minPTS* sorgt für einen besseren Recall, jedoch auch für eine geringere Precision. Eine unerkannte auffällige Belegposition gehört zu dem Beleg, der zwei Belegpositionen auf dem Konto *Verbrauch Verpackungen* besitzt.

ϵ	1,0		1,7		2,2	
minPTS	4	10	4	10	4	10
True-Negatives	310	22	747	738	750	749
True-Positives	27	27	22	26	7	13
False-Negatives	0	0	5	1	20	14
False-Positives	441	729	4	13	1	2
Accuracy	0,43	0,06	0,99	0,98	0	0
Recall	1,00	1,00	0,81	0,96	0,26	0,48
Precision	0,06	0,04	0,85	0,67	0,88	0,87
F-Score	0,11	0,07	0,83	0,79	0,40	0,62

Tabelle 4.2: Ergebnisse von DBSCAN auf dem Konto Verbrauch Verpackungen

Ähnlich ist das Verhalten von DBSCAN auf dem Konto *Umsatzerlöse Inland*. Bei einem ϵ-Wert von 1,0 bildeten sich 330 bzw. 54 Cluster und es gibt viele False-Positives. Bei

einem ϵ-Wert von 1,7 verbesserte sich die Accuracy auf 90 bzw. 84 Prozent. Auch ein großer Teil der Ausreißer konnte erfolgreich erkannt werden. Wenn für ϵ 2,2 verwendet wird, steigt die Accuracy zwar weiter, jedoch werden höchsten die Hälfte aller Ausreißer erkannt. Dieser Trend setzt sich bei einem ϵ-Wert von 2,6 fort.

ϵ	1,0		1,7		2,2		2,6	
minPTS	4	10	4	10	4	10	4	10
True-Negatives	2135	678	5800	5319	6232	6082	6318	6283
True-Positives	203	229	126	183	82	131	24	76
False-Negatives	38	12	115	58	159	110	217	165
False-Positives	4212	5669	547	1028	115	265	29	64
Accuracy	0,35	0,14	0,90	0,84	0,96	0,94	0,96	0,97
Recall	0,84	0,95	0,52	0,76	0,34	0,54	0,10	0,32
Precision	0,05	0,04	0,19	0,15	0,42	0,33	0,45	0,54
F-Score	0,09	0,07	0,28	0,25	0,37	0,41	0,16	0,40

Tabelle 4.3: Ergebnisse von DBSCAN auf dem Konto Umsatzerlöse Inland

4.3 Erprobung von OPTICS

Erwartung

Da es sich bei OPTICS um ein Clustering-Verfahren handelt ist es nicht darauf optimiert Ausreißer zu finden. Trotzdem ist OPTICS in der Lage unterschiedliche lokale Dichten zu berücksichtigen, da die Umgebung jedes Datenpunktes analysiert wird. Ein mögliches Problem ist die global definierte maximale Erreichbarkeitsdistanz, da diese für alle Datenpunkte gilt. Es ist also möglich, dass ein Ausreißer eine höhere Erreichbarkeitsdistanz aufweist als die restlichen Datenpunkte in seinem Cluster, aber trotzdem im globalen Vergleich eine geringe Erreichbarkeitsdistanz aufweist. Diese Ausreißer können dann nicht erkannt werden.

Wahl der Parameter und Anpassungen

Der Parameter ε wird auf ∞ gesetzt, da ein zu geringerer Wert die Leistung von OPTICS nur negativ beeinflusst.

minPTS, also die Anzahl der Datenpunkte, die für die Berechnung der Erreichbarkeitsdistanz berücksichtigt werden, werden für die Konten *Umsatzerlöse EU* und *Verbrauch Verpackung* auf 10 gesetzt. Für das Konto *Umsatzerlöse Inland* wird der Parameter *minPTS* auf 40 gesetzt. Diese Wahl ist mit der unterschiedlichen Größe der einzelnen Konten zu erklären. Bei einem kleineren Wert würde sich eine kleine Menge von Ausreißer gegenseitig stützen. Ein größerer Wert sorgt dafür, dass Cluster aus unauffällige Datenpunkten, die kleiner als *minPTS* sind, sich nicht mehr gegenseitig stützen könnten und so fälschlicherweise als Ausreißer markiert werden.

Für die Wahl der maximalen Erreichbarkeitsdistanzen wird der Mittelwert, sowie die Standardabweichung jedes Kontos berücksichtigt. Im Folgenden wird die Wahl am Beispiel des Kontos *Umsatzerlöse EU* erläutert. Die in Abschnitt 3.6 berechneten Werte für den Mittelwert und die Standardabweichung sind 3,4 und 0,61. Nun werden die Distanzen aus der Tabelle A.2 entnommen, die in der Umgebung des Mittelwerts abzüglich der einfachen Standardabweichung liegen. In der Nähe von 2,79 liegen die Werte 1,41, 2,00, 2,45, 2,83 und 3,16. Diese Werte sind mögliche Kandidaten für die maximale Erreichbarkeitsdistanz, um jedoch Berechnungsfehler durch Fließkommaarithmetik zu verhindern, werden Distanzen zwischen diesen Werten verwendet. Daraus folgen die Werte 1,7, 2,2, 2,6 und 3,0 für die maximalen Erreichbarkeitsdistanzen auf dem Konto *Umsatzerlöse EU*. Nach demselben Verfahren ergeben sich die Grenzwerte für das Konto *Verbrauch Verpackung* von 1,0, 1,7, 2,2 und 2,6 und für das Konto *Umsatzerlöse Inland* von 2,2, 2,6, 3,0 und 3,2.

Erreichte Leistungen

Die erreichten Leistungen von OPTICS auf dem Konto *Umsatzerlöse EU* sind in Abbildung 4.1 und Tabelle 4.4 zu erkennen. In der Abbildung 4.1 werden die Erreichbarkeitsdistanzen der einzelnen Datenpunkte abgebildet. Die Datenpunkte werden hier in der Clusterreihenfolge aus OPTICS sortiert. Die rot markierten Balken zeigen, dass es sich bei diesem Datenpunkt um einen Ausreißer handelt, während grüne Balken einen Inliner darstellen. Es ist zu erkennen, dass zwei Drittel aller Ausreißer entweder am Anfang oder am Ende der Liste einsortiert wurden. Die restlichen Ausreißer liegen in der Mitte der Liste und sind ein Teil unauffälliger Cluster. Durch eine Reduzierung der maximalen Erreichbarkeitsdistanz auf 1,7 lassen sich zwar alle Ausreißer erkennen, jedoch nur durch eine starke Verschlechterung aller anderen Metriken. Der Unterschied zwischen 2,6 und 3,0 ist nur eine Verschlechterung des False-Positives-Wertes. Das beste Ergebnis wird mit einer Distanz von 3,0 erreicht, da hier alle Metriken den maximalen Wert aufweisen.

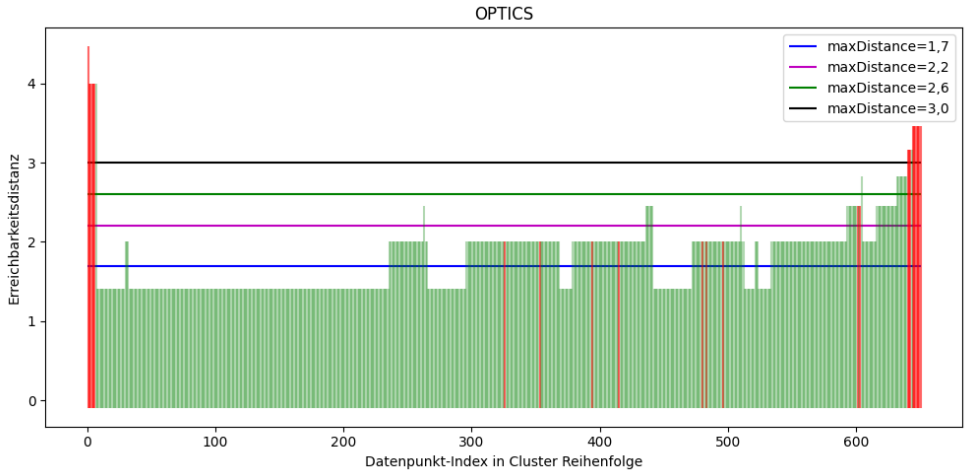

Abbildung 4.1: Erreichbarkeitsdistanzen von OPTICS auf dem Konto Umsatzerlöse EU

maxDistance	1,7	2,2	2,6	3,0
True-Negatives	313	581	614	623
True-Positives	26	19	17	17
False-Negatives	0	7	9	9
False-Positives	312	44	11	2
Accuracy	0,52	0,92	0,97	0,98
Recall	1,00	0,73	0,65	0,65
Precision	0,08	0,30	0,61	0,89
F-Score	0,14	0,43	0,63	0,76

Tabelle 4.4: Ergebnisse von OPTICS auf dem Konto Umsatzerlöse EU

In Abbildung 4.2 und Tabelle 4.5 werden die Ergebnisse auf dem Konto *Verbrauch Verpackungen* dargestellt. Aus der Abbildung 4.2 lässt sich eindeutig erkennen, dass fast alle Ausreißer am Anfang oder am Ende der Cluster-Reihenfolge liegen. Es gibt nur einen einzigen Ausreißer, der sich in der Mitte der Liste befindet. Genauere Untersuchungen haben ergeben, dass es sich bei diesem Datenpunkt um einen der beiden Transaktionen handelt, die auf demselben Beleg zu finden sind. Diese Anomalie wurde in Abschnitt 3.2 beschrieben. Mit einer Erreichbarkeitsdistanz von 2,2 und 2,6 ergibt sich der höchste Precision-Wert, jedoch wird bei 2,6 nur jeder vierte Ausreißer erkannt. Bei einer Distanz von 1,0 wurden viele unauffällige Transaktionen als Ausreißer markiert. Dies resultiert

in einer schlechten Accuracy, Precision und F-Score. Der beste F-Score wurden mit einer Erreichbarkeitsdistanz von 1,7 erreicht.

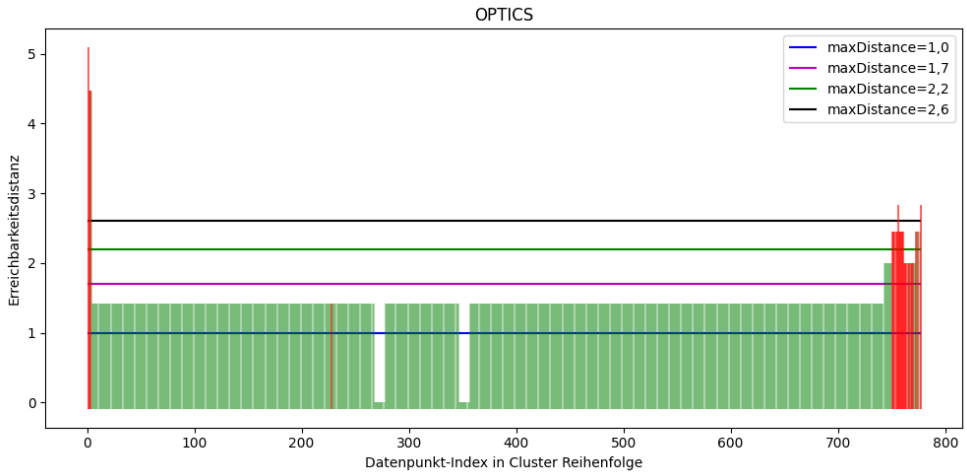

Abbildung 4.2: Erreichbarkeitsdistanzen von OPTICS auf dem Konto Verbrauch Verpackungen

maxDistance	1,0	1,7	2,2	2,6
True-Negatives	20	737	748	750
True-Positives	27	26	18	6
False-Negatives	0	1	9	21
False-Positives	731	14	3	1
Accuracy	0,06	0,98	0,98	0,97
Recall	1,00	0,96	0,67	0,22
Precision	0,04	0,65	0,86	0,86
F-Score	0,07	0,78	0,75	0,35

Tabelle 4.5: Ergebnisse von OPTICS auf dem Konto Verbrauch Verpackungen

Die Ergebnisse auf dem Konto *Umsatzerlöse Inland* werden in Abbildung 4.3 und Tabelle 4.6 dargestellt. Aus Abbildung 4.3 ist zu entnehmen, dass gerade einmal die Hälfte aller Ausreißer sich am Anfang oder am Ende der Liste befinden. Diese besitzen dafür jedoch eine sehr hohe Erreichbarkeitsdistanz und werden so mit einer maximalen Erreichbarkeitsdistanz von 3,0 bzw. 3,2 erkannt. Durch eine Reduzierung der maximalen Erreichbarkeitsdistanz auf 2,6 oder sogar auf 2,2 kann der Recall auf 0,55 bzw. 0,68

erhöht werden. Dies fällt jedoch sehr zu Lasten der Accuracy und Precision. Ein noch geringerer Grenzwert von 1,8 würde zwar noch mehr Ausreißer erkennen, jedoch würde dies das Problem der großen Anzahl an False-Positives nur vergrößern. Dieser Grenzwert wäre also nicht zielführend.

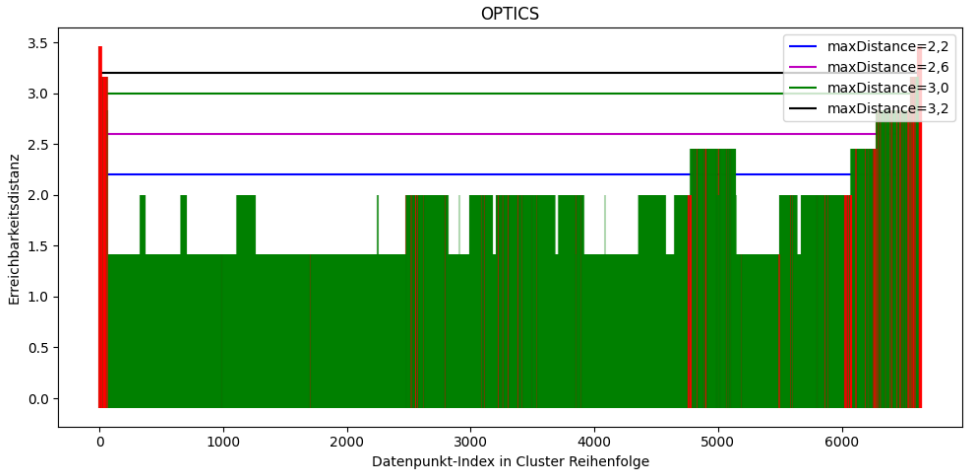

Abbildung 4.3: Erreichbarkeitsdistanzen von OPTICS auf dem Konto Umsatzerlöse Inland

maxDistance	2,2	2,6	3,0	3,2
True-Negatives	5529	6051	6300	6344
True-Positives	165	133	102	37
False-Negatives	76	108	139	204
False-Positives	818	296	47	3
Accuracy	0,86	0,94	0,97	0,97
Recall	0,68	0,55	0,42	0,15
Precision	0,17	0,31	0,68	0,92
F-Score	0,27	0,40	0,52	0,26

Tabelle 4.6: Ergebnisse von OPTICS auf dem Konto Umsatzerlöse Inland

4.4 Erprobung von LOF

Erwartung

Da LOF ein Verfahren ist, das für die Erkennung von Ausreißern entwickelt wurde und zudem Unterschiede in der lokalen Dichte berücksichtigt, wird erwartet, dass LOF solide Ergebnisse erzielt. Es werden jedoch keine perfekten Ergebnisse erwartet, da dieses Verfahren jedes Attribut mit einer gleichen Gewichtung berücksichtigt. Es ist aber zu erwarten, dass einige Attribute eine größere Bedeutung besitzen als Andere. Eine weitere Schwierigkeit ist die Bestimmung eines richtigen Grenzwertes für die *contamination*, da zu erwarten ist, dass dieser stark zwischen den Konten variiert.

Wahl der Parameter und Anpassungen

LOF benötigt für die Berechnung der Ausreißer-Scores nur den Parameter *minPTS*. Hier werden die Werte aus dem Abschnitt 4.3 verwendet. Da das Risiko besteht, dass sich Ausreißer gegenseitig stützen, werden auch größere Werte für *minPTS* erprobt. Für die Konten *Umsatzerlöse EU* und *Verbrauch Verpackung* wurden die drei Werte 10, 25 und 50 gewählt. Für das Konto *Umsatzerlöse Inland* wurden die Werte 40 und 500 verwendet.

Der Parameter *contamination* gibt das Verhältnis zwischen auffälligen und unauffälligen Datenpunkten an. Je größer dieses Verhältnis ist, umso eher wird ein Datenpunkt vom Verfahren als Ausreißer markiert. Eine höhere *contamination* sorgt für einen besseren Recall-Wert, jedoch verschlechter sich der Precision-Wert. Der ideale Wert muss anwendungsspezifisch entschieden werden. Es kann jedoch eine untere Grenze definiert werden. Aus dem Abschnitt 3.2 ist zu entnehmen, dass der Anteil der Ausreißer bei allen Konten knapp unter vier Prozent liegt. Deshalb liegt die untere Grenze bei einer *contamination* von 0,04 und wird dann in Schritten von 0,02 auf 0,10 erhöht.

Erreichte Leistungen

Die Ergebnisse von LOF auf dem Konto *Umsatzerlöse EU* sind in Tabelle 4.7 dargestellt. In der Abbildung 4.4 werden die Ausreißer Scores der einzelnen Datenpunkte dargestellt. Für die Erzeugung der Grafik wurde ein Wert von 50 für den Parameter *minPTS* verwendet. Die roten Punkte wurden von den Wirtschaftsprüfern als Ausreißer markiert und die grünen Punkte sind unauffällige Transaktionen. Aus der Tabelle 4.7 ist zu entnehmen, dass bei der ursprünglichen Wahl von 10 für *minPTS* unerwartet schlechte Ergebnisse erreicht werden. Durch eine Erhöhung auf 25 verbessern sich diese zwar, jedoch wird das

Maximum erst mit 50 erreicht. Größere Werte für *minPTS* führen weder zu einer Verbesserung noch zu einer Verschlechterung. Bei einem größeren Grad der Kontaminierung verbessert sich zwar der Recall leicht, jedoch verschlechtern sich dabei die restlichen Metriken. Um die letzten verbleibenden acht False-Negatives zu erkennen, müsste der Grad der Kontaminierung so weit angehoben werden, dass ungefähr die Hälfte aller Transaktionen als Ausreißer markiert werden.

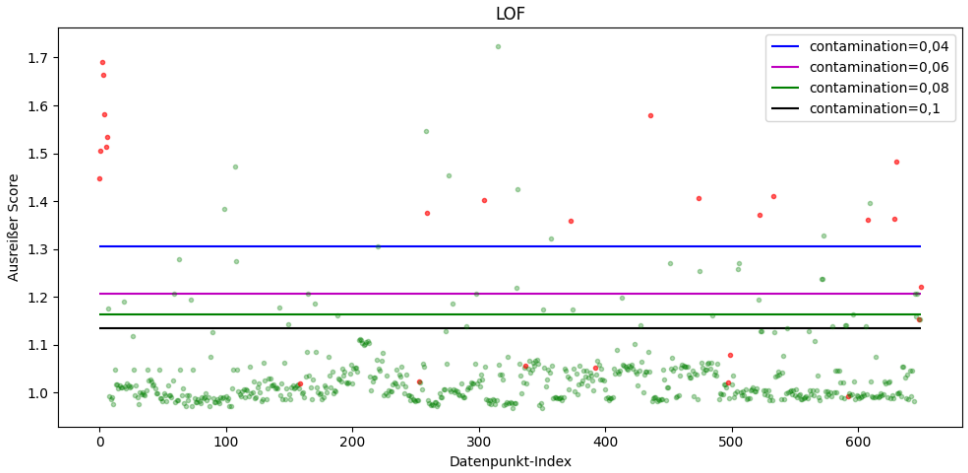

Abbildung 4.4: Ausreißer Score von LOF auf dem Konto Umsatzerlöse EU

minPTS	10	25	50			
contamination	0,04	0,04	0,04	0,06	0,08	0,10
True-Negatives	609	614	616	606	590	583
True-Positives	7	14	17	18	18	19
False-Negatives	19	12	9	8	8	7
False-Positives	16	11	9	19	35	42
Accuracy	0,95	0,96	0,97	0,96	0,93	0,92
Recall	0,27	0,54	0,65	0,69	0,69	0,73
Precision	0,30	0,56	0,65	0,49	0,34	0,31
F-Score	0,29	0,55	0,65	0,57	0,46	0,44

Tabelle 4.7: Ergebnisse von LOF auf dem Konto Umsatzerlöse EU

In der Tabelle 4.8 sind die Leistungen von LOF auf dem Konto *Verbrauch Verpackungen* notiert. Bei einem *minPTS*-Wert von 10 wurde kein Ausreißer korrekt erkannt. Es

wurden ausschließlich unauffällige Transaktionen als Ausreißer markiert. Bei einem *minPTS*-Wert von 25 verbessern sich die Ergebnisse, jedoch werden noch viele Ausreißer nicht korrekt erkannt. In der Grafik 4.5 werden die Ausreißer Scores bei einem *minPTS*-Wert von 50 dargestellt. Es ist eine klare visuelle Trennung zwischen den auffälligen und den unauffälligen Transaktionen zu erkennen. Eine einzige auffällige Transaktion liegt jedoch weit unterhalb dieser visuellen Trennlinie. Hierbei handelt es sich um eine der Transaktionen, die in Abschnitt 3.2 bereits aufgefallen sind. Es handelt sich hier um den Beleg, der zwei Belegpositionen auf demselben Konto besitzt. Um selbst diese Transaktion noch zu identifizieren, müsste der *contamination* Wert auf 0,13 erhöht werden.

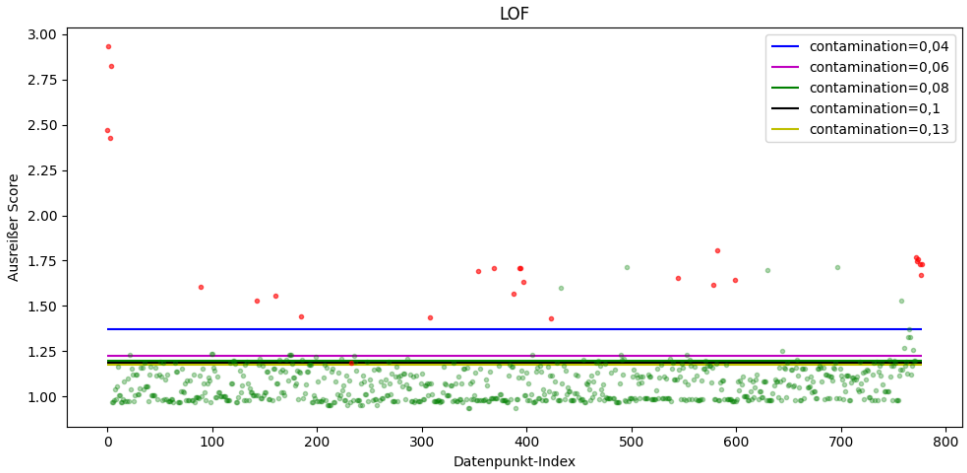

Abbildung 4.5: Ausreißer Score von LOF auf dem Konto Verbrauch Verpackungen

minPTS	10	25			50		
contamination	0,04	0,04	0,04	0,06	0,08	0,10	0,13
True-Negatives	728	744	745	732	720	706	679
True-Positives	0	15	26	26	26	26	27
False-Negatives	27	13	1	1	1	1	0
False-Positives	23	7	6	19	31	45	72
Accuracy	0,94	0,98	0,99	0,97	0,96	0,94	0,91
Recall	0,00	0,56	0,96	0,96	0,96	0,96	1,00
Precision	0,00	0,68	0,81	0,58	0,46	0,37	0,27
F-Score	0,00	0,61	0,88	0,72	0,62	0,53	0,43

Tabelle 4.8: Ergebnisse von LOF auf dem Konto Verbrauch Verpackungen

Die Ergebnisse von LOF auf dem Konto *Umsatzerlöse Inland* sind in Tabelle 4.9 abgebildet. Bei einem *minPTS*-Wert von 40 wurden nur 18 Prozent aller Ausreißer korrekt identifiziert. Gleichzeitig wurden 198 Transaktionen fälschlicherweise als Ausreißer markiert. Bei einem *minPTS*-Wert von 500 verbessern sich die Werte im Allgemeinen, wie in Abbildung 4.6 zu sehen ist. Trotzdem werden selbst bei einer *contamination* von 0,10 nur zwei Drittel aller Ausreißer erkannt. Zudem gibt es eine sehr große Anzahl an False-Positives und selbst durch eine weitere Erhöhung der *contamination* würden viele Ausreißer nicht erkannt werden, ohne die gesamte Accuracy des Verfahrens signifikant zu verschlechtern.

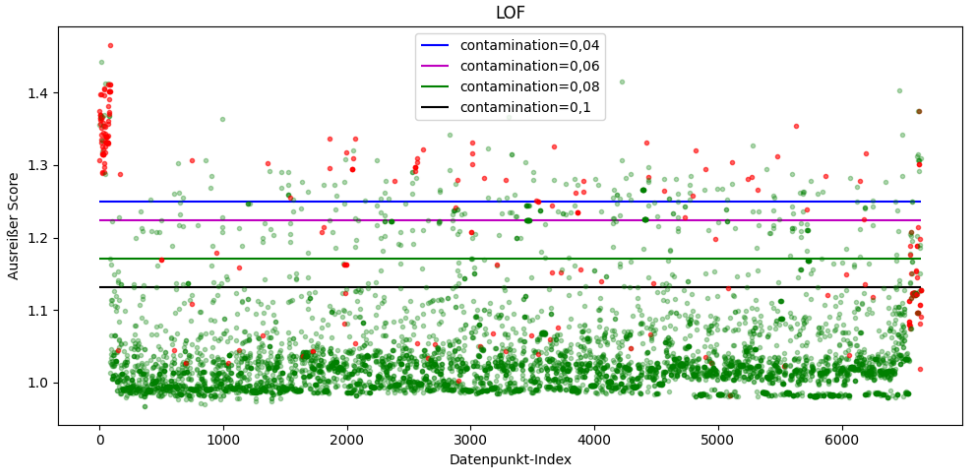

Abbildung 4.6: Ausreißer Score von LOF auf dem Konto Umsatzerlöse Inland

minPTS	40	500			
contamination	0,04	0,04	0,06	0,08	0,10
True-Negatives	6144	6208	6085	5966	5851
True-Positives	43	125	135	147	166
False-Negatives	198	116	106	94	75
False-Positives	203	139	262	381	496
Accuracy	0,94	0,96	0,94	0,93	0,91
Recall	0,18	0,52	0,56	0,61	0,69
Precision	0,17	0,47	0,34	0,28	0,25
F-Score	0,18	0,50	0,42	0,38	0,37

Tabelle 4.9: Ergebnisse von LOF auf dem Konto Umsatzerlöse Inland

4.5 Erprobung von ROCK

Erwartung

Bei ROCK handelt es sich um ein Verfahren, das speziell für kategorielle Datensätze entwickelt wurden [GRS99, SMG19]. Ein Problem von ROCK sind die vier Parameter, die, mit Ausnahme von *eps*, nur durch Exploration bestimmt werden können. Auch ist aus den Abschnitten 3.2 und 3.6 bekannt, dass die einzelnen Konten unterschiedliches Verhalten aufweisen. Eine Parameterkombination kann auf einem Konto gute Ergebnisse und auf einem anderen schlechte Ergebnisse erzielen. Ein anderes Problem, dass schon in [GRS99] beschrieben wurde, ist, dass größere Cluster kleinere Cluster absorbieren können. So ist es möglich, dass ein kleinerer Cluster aus auffällige Transaktionen in einem größeren Cluster verschwindet und so die Transaktionen nicht mehr identifiziert werden können.

Wahl der Parameter und Anpassungen

Da für die Berechnung der Links die L2-Norm verwendet wird, sind für die Wahl von ϵ nur Werte aus der Tabelle A.2 möglich. Für das Konto *Umsatzerlöse EU* ergeben sich so wieder die Werte 1,7, 2,2, 2,6 und 3,0. Für das Konto *Verbrauch Verpackung* lauten die Werte 1,0, 1,7, 2,2 und 2,6 und für das Konto *Umsatzerlöse Inland* 2,2, 2,6, 3,0 und 3,2.

Für den Parameter θ wurde zum einen der Standardwert von 0,5, als auch ein niedrigerer Wert von 0,25 erprobt. Hiermit soll untersucht werden, ob die Wahl des Parameters einen großen Einfluss auf das Endergebnis hat.

Der Parameter k, also die Anzahl an Clustern, die entstehen sollen, wurde für alle Konten auf 20 gesetzt. Experimente haben ergeben, dass bei kleineren Werten zu viele Cluster aus auffälligen Transaktionen in größere Cluster aufgehen und bei größeren Werten bilden sich noch viele Cluster aus unauffälligen Transaktionen.

Für *sizethreshold* wurde der Wert für die beiden Konten *Umsatzerlöse EU* und *Verbrauch Verpackung* auf 15 gesetzt. Für das Konto *Umsatzerlöse Inland* wurde dieser auf 30 gesetzt. Diese Wahl ist damit zu begründen, dass die ersten beiden Konten nur sehr wenige auffällige Transaktionen besitzen. Das Konto *Umsatzerlöse Inland* besitzt wesentlich mehr auffällige Transaktionen, die sich in größeren Clustern versammeln könnten.

Das Konto *Umsatzerlöse Inland* wurde zudem besonders behandelt. Aufgrund der großen Anzahl von Transaktionen auf diesem Konto und einem Worst-Case Laufzeitverhalten

von $O\left(n^2 + nm_m m_a + n^2 log\,(n)\right)$, musste der Datensatz für dieses Konto in drei Teil-Datensätze zerlegt werden. n beschreibt hier die Anzahl von Datenpunkten, m_m die maximale Anzahl von Nachbarn und m_a die durchschnittliche Anzahl an Nachbarn. Selbst bei einer Zerlegung benötigt die Berechnung eines Teils des Datensatzes mehr als 30 Minuten. Die dargestellten Ergebnisse sind die Summe aller drei Teilergebnisse.

Erreichte Leistungen

Auf dem Konto *Umsatzerlöse EU* wurden von ROCK bei einem ϵ-Wert von 1,7 in beiden Fällen 112 Cluster gebildet, also weit mehr, als eigentlich durch den Parameter k vorgegeben wurde. Dies widerspricht dem Prinzip von ROCK, daher muss es sich hierbei um eine Eigenheit der Implementierung handeln. Durch die große Anzahl an Clustern entstand eine große Menge kleiner Cluster, die als auffällig erkannt wurden und somit sehr viele False-Positives erzeugen. Ein Unterschied bei der Wahl von θ ist bei kleinen Distanzen nicht zu erkennen. Bei einem ϵ-Wert von 2,6 sorgt eine geringere Bestrafung große Cluster zu erzeugen dafür, dass mehr unauffällige Transaktionen in einem größeren Cluster gesammelt und so nicht als Ausreißer bewertet wurden. Anders sind die Auswirkungen bei einem ϵ-Wert von 3,0, hier verbessert sich der Recall zwar leicht, jedoch steigt auch die Anzahl an False-Positives. Die genauen Werte sind in Tabelle 4.10 aufgelistet.

ϵ	1,7		2,2		2,6		3,0	
θ	0,25	0,5	0,25	0,5	0,25	0,5	0,25	0,5
True-Negatives	481	481	592	592	612	616	613	606
True-Positives	22	22	19	19	17	17	18	19
False-Negatives	4	4	7	7	9	9	8	7
False-Positives	144	144	33	33	13	9	12	19
Accuracy	0,77	0,77	0,94	0,94	0,97	0,97	0,97	0,96
Recall	0,85	0,85	0,73	0,73	0,65	0,65	0,69	0,73
Precision	0,13	0,13	0,37	0,37	0,57	0,65	0,60	0,50
F-Score	0,23	0,23	0,49	0,49	0,61	0,65	0,64	0,59

Tabelle 4.10: Ergebnisse von ROCK auf dem Konto Umsatzerlöse EU

Die erreichten Ergebnisse von ROCK auf dem Konto *Verbrauch Verpackungen* werden in Tabelle 4.11 gezeigt. Hier tritt ein ähnliches Phänomen bei einem zu kleinen ϵ-Wert auf. Insgesamt bilden sich in beiden Fällen 403 Cluster und somit werden alle Transaktionen als Ausreißer markiert. Bei allen anderen Werten für ϵ gibt es jeweils eine nicht erkannte

auffällige Transaktion. Hierbei handelt es sich um die Transaktion, die bereits aus Abschnitt 3.2 bekannt ist. Eine Veränderung des θ-Wertes sorgt bei einem ϵ-Wert von 1,7 zu keiner Veränderung, bei 2,2 sorgt es für mehr False-Positives und bei einem ϵ-Wert von 2,6 für weniger False-Positives.

ϵ	1,0		1,7		2,2		2,6	
θ	0,25	0,5	0,25	0,5	0,25	0,5	0,25	0,5
True-Negatives	0	0	745	745	746	719	738	744
True-Positives	27	27	26	26	26	26	26	26
False-Negatives	0	0	1	1	1	1	1	1
False-Positives	751	751	6	6	5	32	13	7
Accuracy	0,03	0,03	0,99	0,99	0,99	0,96	0,98	0,99
Recall	1,00	1,00	0,96	0,96	0,96	0,96	0,96	0,96
Precision	0,03	0,03	0,81	0,81	0,84	0,45	0,67	0,79
F-Score	0,07	0,07	0,88	0,88	0,90	0,61	0,79	0,87

Tabelle 4.11: Ergebnisse von ROCK auf dem Konto Verbrauch Verpackungen

Die Tabelle 4.12 enthält die Ergebnisse von ROCK auf dem Konto *Umsatzerlöse Inland*. Es ist zu beobachten, dass für kleinere ϵ-Werte der θ Parameter keinen Einfluss auf die Ergebnisse hat. Anders ist es bei größeren ϵ-Werten. Hier fällt auf, dass ein Wert für θ in einem Fall für einen besseren Recall und in einem Anderen für einen schlechteren Recall sorgt. Durch einen θ-Wert von 0,5 wird jedoch in beiden Fällen der F-Score leicht verbessert.

ϵ	2,2		2,6		3,0		3,2	
θ	0,25	0,5	0,25	0,5	0,25	0,5	0,25	0,5
True-Negatives	5762	5762	6180	6180	6317	6290	6036	6149
True-Positives	163	163	117	117	73	83	148	131
False-Negatives	78	78	124	124	168	158	93	110
False-Positives	585	585	167	167	30	57	311	198
Accuracy	0,90	0,90	0,96	0,96	0,97	0,97	0,94	0,95
Recall	0,68	0,68	0,49	0,49	0,30	0,34	0,61	0,54
Precision	0,22	0,22	0,41	0,41	0,71	0,59	0,32	0,40
F-Score	0,33	0,33	0,45	0,45	0,42	0,44	0,42	0,46

Tabelle 4.12: Ergebnisse von ROCK auf dem Konto Umsatzerlöse Inland

4.6 Erprobung von Autoencodern

Erwartung

Die Erwartungen an den Autoencoder sind sehr hoch, da schon in anderen Arbeiten [ST20, SSB+18] sehr gute Ergebnisse mit Autoencodern erzielt wurden. Die hohen Erwartungen sind auch mit der Funktionsweise eines Autoencoders zu begründen. Dieser lernt selbstständig nach welchen Regeln die Transaktionen auf den Konten erfolgen, er lernt die Bedeutung und die Gewichtung einzelner Attribute, sowie den Zusammenhang zwischen den Attributen.

Wahl der Parameter und Anpassungen

Für den Autoencoder wurde ein Netzwerk aus mehreren vollständig verbundenen Ebenen (fully-connected hidden layers) verwendet. Der Aufbau ähnelt dabei der Struktur aus Abbildung 2.2. Für die hidden Layers wurde ReLU als Aktivierungsfunktion verwendet. Als Aktivierungsfunktion für den Output-Layer wurde die Sigmoid-Funktion verwendet. Dies ist notwendig, damit die Output-Neuronen wieder einen Wertebereich zwischen 0 und 1 annehmen. Als Batch-Größe wird die Anzahl der Transaktionen auf dem Konto verwendet. Aufgrund der geringen Anzahl an Dimensionen nach der Vorverarbeitung (vgl. 3.6) werden für alle Konten fünf Layer mit [16, 8, 4, 8, 16] Neuronen verwendet. Aufgrund der geringen Größe des Netzwerkes wird für nur 200 Epochen trainiert. Zusätzlich wird ein Dropout von 20 Prozent verwendet. Für die *contamination* wurden die Werte 0,04, 0,06, 0,08 und 0,10 verwendet.

Erreichte Leistungen

Die Abbildung 4.7 zeigt die Entwicklung der Lernkurve auf dem Konto *Umsatzerlöse EU* innerhalb von 200 Epochen. Der Autoencoder hat in den ersten 50 Epochen sehr schnell gelernt, jedoch zeigt sich nach der 150. Epoche keine nennenswerte weitere Verbesserung. Die Lernkurven für die Konten *Verbrauch Verpackung* und *Umsatzerlöse Inland* verlaufen analog.

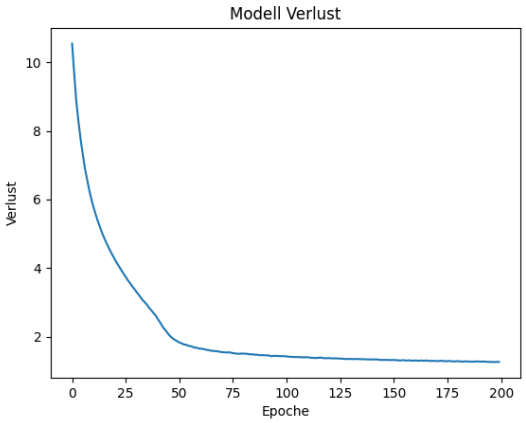

Abbildung 4.7: Entwicklung des Modells auf dem Konto Umsatzerlöse EU

Die Ergebnisse des Autoencoders sind in Tabelle 4.13 zu sehen. Die Ausreißer-Scores für die einzelnen Datenpunkte sind in Abbildung 4.8 dargestellt. Bei einer *contamination* von 0,04 ergibt sich zwar der beste F-Score, jedoch werden nur zwei Drittel aller Ausreißer erkannt. Eine Senkung der *contamination* auf 0,06 ermöglicht es zwar drei weitere Ausreißer zu erkennen, aber die restlichen sieben Ausreißer können auch bei einer weiteren Reduzierung der *contamination* nicht erkannt werden. Die weitere Reduzierung erzeugt mehr False-Positives und verschlechtert somit den F-Score. Alle diese sieben Transaktionen haben ein überdurchschnittlich hohes Transaktionsvolumen. Einer dieser Transaktionen besitzt das höchste Transaktionsvolumen auf diesem Konto.

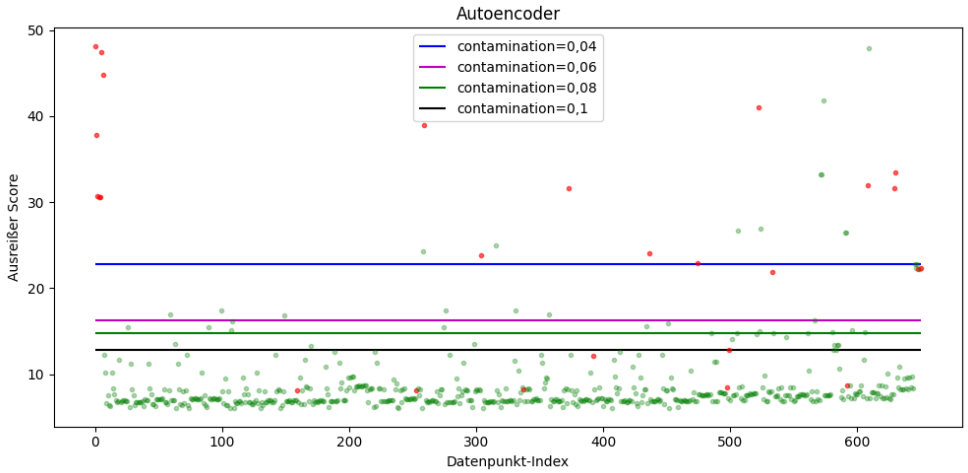

Abbildung 4.8: Ausreißer Score von Autoencoder auf dem Konto Umsatzerlöse EU

contamination	0,04	0,06	0,08	0,10
True-Negatives	615	605	592	580
True-Positives	16	19	19	19
False-Negatives	10	7	7	7
False-Positives	10	20	33	45
Accuracy	0,97	0,96	0,94	0,92
Recall	0,62	0,73	0,73	0,73
Precision	0,62	0,49	0,37	0,30
F-Score	0,62	0,58	0,49	0,42

Tabelle 4.13: Ergebnisse von Autoencoder auf dem Konto Umsatzerlöse EU

Die Resultate auf dem Konto *Verbrauch Verpackungen* sind in der Tabelle 4.14 zu finden. Die Ausreißer-Scores sind in Abbildung 4.9 dargestellt. Die auffälligen Buchungspositionen unterscheiden sich hier deutlich von den restlichen Buchungspositionen. Schon bei einer *contamination* von 0,04 werden 96 Prozent alle auffälligen Transaktionen erkannt. Bei dem einzigen False-Negative handelt es sich um dieselbe Buchungsposition, die bereits aus den vorherigen Abschnitten bekannt ist. Diese Position wird auch bei einer größeren *contamination* nicht erkannt.

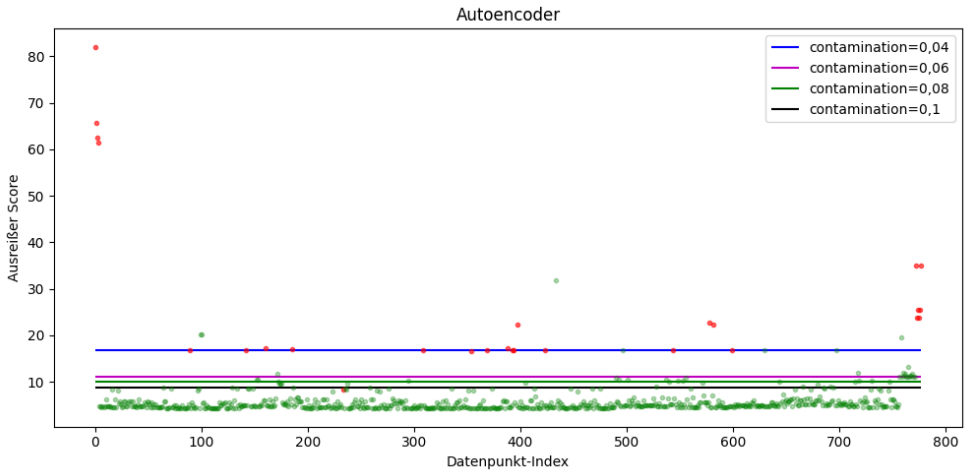

Abbildung 4.9: Ausreißer Score von Autoencoder auf dem Konto Verbrauch Verpackungen

contamination	0,04	0,06	0,08	0,10
True-Negatives	744	730	714	699
True-Positives	25	26	26	26
False-Negatives	2	1	1	1
False-Positives	7	21	37	52
Accuracy	0,99	0,97	0,95	0,93
Recall	0,93	0,96	0,96	0,96
Precision	0,78	0,55	0,41	0,33
F-Score	0,85	0,70	0,58	0,50

Tabelle 4.14: Ergebnisse von Autoencoder auf dem Konto Verbrauch Verpackungen

Die Tabelle 4.15 enthält die erreichten Ergebnisse auf dem Konto *Umsatzerlöse Inland*. Im Gegensatz zum Konto Verbrauch Verpackung ist in Abbildung 4.10 keine eindeutige visuelle Grenze zu erkennen. Viele der auffälligen Transaktionen haben einen ähnlichen Ausreißer-Score, wie die nicht auffälligen Transaktionen. Aus diesem Grund liegt der Recall zwischen 0,40 und 0,61. Der höchste F-Score wird bei einer *contamination* von 0,41 erreicht.

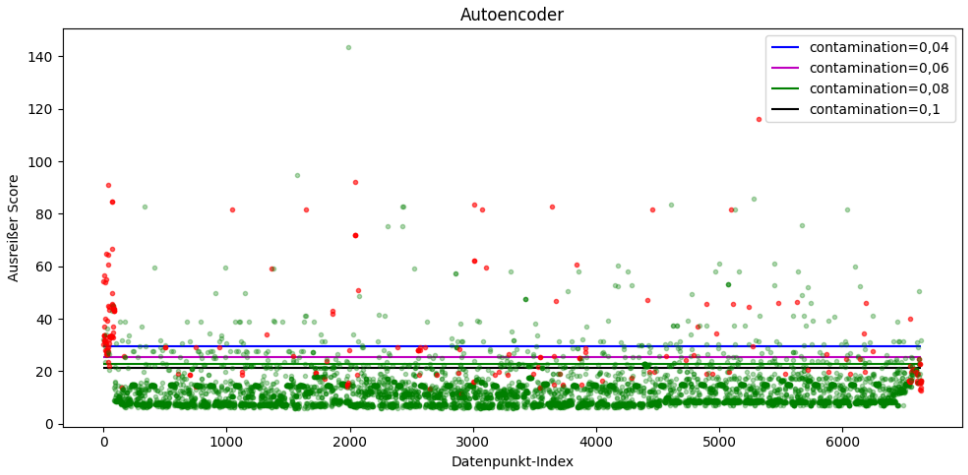

Abbildung 4.10: Ausreißer Score von Autoencoder auf dem Konto Umsatzerlöse Inland

contamination	0,04	0,06	0,08	0,10
True-Negatives	6177	6081	5960	5833
True-Positives	96	132	145	148
False-Negatives	145	109	96	93
False-Positives	170	266	387	514
Accuracy	0,95	0,94	0,93	0,91
Recall	0,40	0,55	0,60	0,61
Precision	0,36	0,33	0,27	0,22
F-Score	0,38	0,41	0,38	0,33

Tabelle 4.15: Ergebnisse von Autoencoder auf dem Konto Umsatzerlöse Inland

4.7 Bewertung der Ergebnisse

Die Untersuchungen der einzelnen Verfahren haben ergeben, dass alle Verfahren auf dem Konto *Verbrauch Verpackungen* die besten Ergebnisse erreicht haben. Jedes Verfahren hatte auf diesem Konto Schwierigkeiten die eine Belegposition zu erkennen, die in Abschnitt 3.2 beschrieben wurde. Alle Verfahren hatten Schwierigkeiten alle Ausreißer auf dem Konto *Umsatzerlöse Inland* korrekt zu erfassen. Es ist also zu vermuten, dass die Transaktionen auf diesem Konto sehr komplexen Mustern folgen und diese Regeln von den Verfahren nicht vollständig erkannt werden.

Der höchste F-Score von 0,76 auf dem Konto *Umsatzerlöse EU* wurde von OPTICS erreicht. Auf dem Konto *Verbrauch Verpackungen* wurde der höchste F-Score von 0,90 von ROCK erreicht. OPTICS erzielte auf dem Konto *Umsatzerlöse Inland* den höchsten F-Score von 0,52. Insgesamt ist aber zu sagen, dass alle Verfahren ähnliche Leistungen zeigen. Es kann nicht eindeutig ein Verfahren bestimmt werden, das auf allen Konten besser oder schlechter als andere Verfahren ist.

5 Verbesserte Datenvorverarbeitung

In diesem Kapitel wird zuerst im Abschnitt 5.1 erläutert, wo Probleme bei der Datenvorverarbeitung aus Kapitel 3 liegen und warum eine andere Datenvorverarbeitung sinnvoll ist. Im Abschnitt 5.2 werden dann die konkreten Maßnahmen beschrieben, um diese Probleme zu lösen. Im letzten Abschnitt 5.3 wird das Resultat der neuen Verarbeitung beschrieben.

5.1 Motivation

Eine erneute Betrachtung der Datenvorverarbeitung ist notwendig, da zum Einen einige Informationen bei der alten Datenvorverarbeitung in Kapitel 3 verloren gehen und zum Anderen da eine One-Hot Kodierung die Reihenfolge ordinaler Attribute nicht berücksichtigt. Als ordinale Attribute werden kategoriale Attribute bezeichnet, die in eine Ordnung gebracht werden können. Als Beispiel sind hier die Ausprägungen „sehr schlecht“, „schlecht“, „gut“ und „sehr gut“ zu nennen. Auch wenn es keine Werte zwischen „gut“ und „sehr gut“ gibt und arithmetische Operationen wenig sinnvoll sind, so haben diese doch eine klare Reihenfolge.

Ein weiteres Problem ist die aktuelle Kodierung der Listen. Als Beispiel dienen im Folgenden die drei Listen [A, B], [A, B, C] und [D, E]. Das Ergebnis der bisherigen Kodierung wäre dann [1, 0, 0], [0, 1, 0] und [0, 0, 1]. Laut dieser Kodierung sind alle drei Listen im gleichen Maße unterschiedlich, obwohl der Unterschied zwischen der ersten und zweiten Liste nur durch ein Element bestimmt wird.

5.2 Erzeugung Features

Zuerst wird das Problem mit den Listen behoben. Dafür wurde die Menge aller Einträge innerhalb der Listen berechnet und von Duplikaten befreit. Die Kardinalität dieser

Menge wird als Länge für das Array verwendet. In einem zweiten Schritt wird jeder Ausprägung dieser Menge ein Index zugeordnet. Als letztes werden die eigentlichen Listen umgewandelt, hierzu wird analysiert welche Einträge in der Liste stehen und diese in Index-Werte umgerechnet. Diese Index-Werte geben an, an welcher Stelle im Array eine 1 platziert werden muss. Diese Umwandlung wird nun im Folgenden an dem Beispiel aus Abschnitt 5.1 verdeutlicht. Aus den drei Listen [A, B], [A, B, C] und [D, E] wird die Menge aller Einträge berechnet. Diese lautet [A, B, C, D, E] und jedem Eintrag wird ein Index zugewiesen [A:0, B:1, C:2, D:3, E:4]. Abschließend werden die Listen kodiert. Aus der Liste [A, B] wird [1, 1, 0, 0, 0], aus [A, B, C] wird [1, 1, 1, 0, 0] und aus [D, E] wird [0, 0, 0, 1, 1]. Für die Verfahren ist es nun einfacher zu erkennen, dass die ersten beiden Listen mehr Gemeinsamkeiten untereinander aufweisen als zur letzten Liste.

Eine weitere Information, die verloren geht ist, ob es sich bei einem Wochentag um einen Arbeitstag oder um einen der beiden Tage am Wochenende handelt. Bei weiteren Untersuchungen ist aufgefallen, dass keine der drei Konten Buchungen beinhalten, die an einem Sonntag lagen. Zudem weist das Konto *Umsatzerlöse EU* auch keine Buchungen auf, die auf einem Samstag lagen. Für dieses Konto wäre diese Information nur redundant und kann weggelassen werden. Für die beiden anderen Konten wird ein binäres Attribut hinzugefügt, welches One-Hot kodiert wird.

Des Weiteren wird das Attribut *AmountLocalCurrency* erneut betrachtet. Im Allgemeinen gelten gerade Beträge als auffällig, da diese möglicherweise von einem Menschen eingetragen wurden. Auch diese Information geht bei der bisherigen Kodierung verloren. Um diese Information den Verfahren zur Verfügung zu stellen, wird ein weiteres Attribut *AmountIsEven* hinzugefügt. Es wird erfasst, ob das Attribut *AmountLocalCurrency* ungerade oder auf 1€, 10€ oder 100€ glatt ist. Die Tabelle 5.1 enthält eine Übersicht über das Vorkommen gerade Beträge auf den einzelnen Konten. Es ist zu erkennen, dass die Konten *Umsatzerlöse EU* und *Verbrauch Verpackung* keine Beträge besitzen, die glatt auf 100€ sind. Das Attribut wird nun in ein drei- bzw. vier-dimensionales Array umgewandelt. Hierbei ist zu ergänzen, dass ein Betrag, der zur Gruppe „100€ glatt“ gehört auch in die Gruppe „10€ glatt“ und „1€ glatt“ gehört und so an drei Stellen eine 1 besitzt.

	krumme Beträge	1€ glatt	10€ glatt	100€ glatt
Umsatzerlöse EU	593	57	1	0
Verbrauch Verpackung	768	9	1	0
Umsatzerlöse Inland	5742	695	200	6

Tabelle 5.1: Vorkommen von geraden Beträgen

Durch die Listen-Attribute werden zwar Informationen aus dem gesamten Beleg extrahiert, jedoch fehlt die Information, wie viele Belegpositionen auf einem Beleg sind. Dazu wird das neue Attribut *MaxDocPos* eingeführt. Auf den drei Konten befinden sich nur Belege mit 2, 3, 4, 6, 8, 10 oder 14 Buchungspositionen pro Beleg. Jeder dieser Werte bildet eine eigene Kategorie.

Auch ist es üblich Ausprägungen, die nur in fünf bis zehn Prozent aller Fälle vorkommen, in „minority class(es)“ zusammenzuführen [SMG19]. Der Gedanke hierbei ist, dass nur angegeben wird, ob es sich um eine seltene Ausprägung handelt, welche genau es ist, ist hierbei nicht relevant. Für das Konto *Umsatzerlöse EU* betrifft dies die Attribute *tcodename* und *userid*. Die Attribute *doctype*, *postingkey*, *userid*, *tcodename* und *DocOrigin* vom Konto *Verbrauch Verpackungen* werden auch nach diesem Prinzip umgewandelt. Auf dem Konto *Umsatzerlöse Inland* betrifft es die Attribute *taxcode*, *userid* und *tcodename*. Die einzige Ausnahme bildet das Attribut *AccountingPeriod*, da dies gleichmäßig verteilt ist und die Häufigkeiten zwischen 6,89 Prozent und 10,16 Prozent liegen.

Zuletzt werden alle ordinalen Attribute, also solche bei denen die Ausprägungen in eine Reihenfolge gebracht werden können, anders kodiert. Diese Kodierung ist zwar ähnlich wie die One-Hot Kodierung jedoch wird die 1 nicht an nur einer Stelle gesetzt, sondern auch bei allen vorherigen Ausprägungen aus der Reihenfolge. Dies betrifft die folgenden Attribute *AccountingPeriod*, *CreationDateDayOfWeek*, *AmountLocalCurrency*, *AmountIsEven*, *MaxDocPos*. Diese Änderung bringt die Ausprägungen, die sich ähnlicher sind, näher zusammen.

5.3 Ergebnis der verbesserten Datenvorverarbeitung

Ähnlich wie im Abschnitt 3.6 werden im Folgenden die Ergebnisse der zweiten Datenvorverarbeitung beschrieben. Die genaue Verteilung aller Distanzen auf den drei Konten kann der Tabelle im Anhang A.3 entnommen werden.

Umsatzerlöse EU

Insgesamt wurden 20 Attribute für das Konto *Umsatzerlöse EU* verarbeitet und es entstanden daraus 101 Dimensionen. Dies sind 18 Dimensionen weniger als nach der ersten Datenvorverarbeitung. Diese Reduktion ist durch die neue Kodierung der Listen, als auch durch die Bildung von minority-classes zu erklären. Durch die neue Kodierung der Listen und ordinaler Attribute stieg die Anzahl unterschiedlicher Distanzen auf 50. Die Abstände liegen nun zwischen 0,00 und 7,00 mit einem Mittelwert von 3,84 und einer Standardabweichung von 0,89. Die genaue Verteilung ist in Abbildung 5.1 zu sehen.

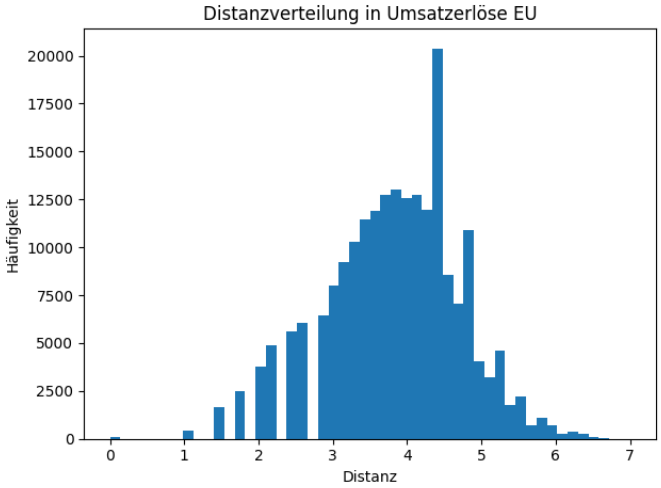

Abbildung 5.1: Distanzverteilung in Umsatzerlöse EU

Verbrauch Verpackungen

Für das Konto *Verbrauch Verpackung* wurden 21 Attribute verwendet. Die Anzahl der Dimensionen stieg von 72 auf 84. Auch auf diesem Konto stieg die Anzahl unterschiedlicher Distanzen auf 53. Die obere Grenze der Abstände liegt bei 7,34. Der neue Mittelwert liegt bei 2,90 und die neue Standardabweichung bei 0,85. Die Verteilung für das Konto *Verbrauch Verpackungen* ist in Abbildung 5.2 zu sehen.

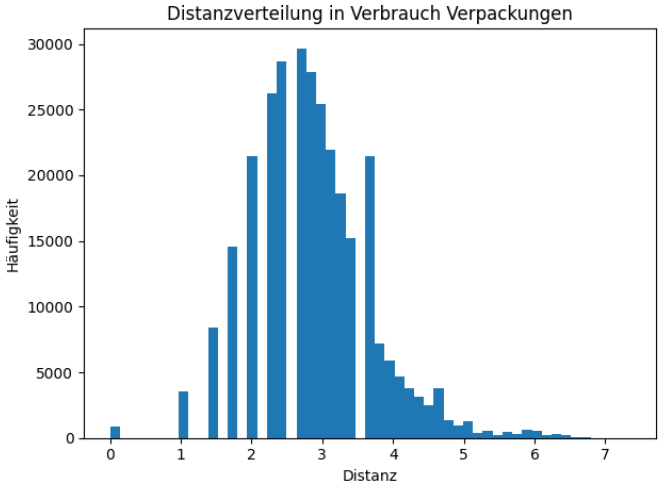

Abbildung 5.2: Distanzverteilung in Verbrauch Verpackungen

Umsatzerlöse Inland

Für das Konto *Umsatzerlöse Inland* wurden insgesamt 23 Attribute verarbeitet. Die Anzahl der Dimensionen wurde von 251 auf 199 reduziert. Die Anzahl der unterschiedlichen Distanzen stieg von 17 auf 62. Das neue Maximum liegt bei 7,81, der neue Mittelwert bei 4,47 und die neue Standardabweichung bei 0,95. Die genaue Verteilung ist in Abbildung 5.3 zu sehen.

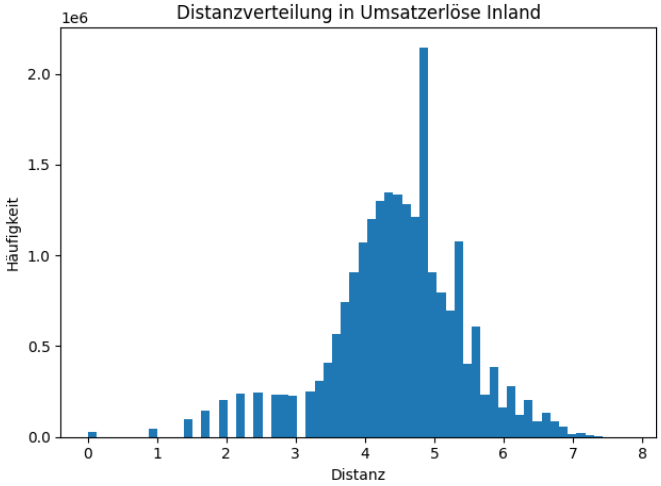

Abbildung 5.3: Distanzverteilung in Umsatzerlöse Inland

6 Erprobung der Verfahren auf dem erweitertem Datensatz

In diesem Kapitel werden alle Verfahren auf Basis der erweiterten Datenvorverarbeitung aus Kapitel 5 erprobt. Die erreichten Ergebnisse werden im Abschnitt 6.6 zusammengefasst.

6.1 Erprobung von DBSCAN

Wahl der Parameter und Anpassungen

Aufgrund der neuen Datenvorverarbeitung haben sich die Distanzen zwischen den Punkten geändert. Die neuen Werte für ϵ werden nach demselben Schema wie in Abschnitt 4.2 berechnet. Für das Konto *Umsatzerlöse EU* ergeben sich die Werte 2,3, 2,5, 2,7 und 2,9. Die Werte für das Konto *Verbrauch Verpackungen* sind 1,6, 1,9 und 2,1 und für das Konto *Umsatzerlöse Inland* 2,1, 2,3, 2,5 und 2,9. Die gewählten Parameter für *minPTS* blieben unverändert bei 4 und 10.

Erreichte Leistungen

Die erreichten Ergebnisse von DBSCAN auf Basis der neuen Datenvorverarbeitung sind in Tabelle 6.1 zu finden. Die Accuracy hat sich bei allen Parameterkombinationen verbessert. Es ist zu erkennen, dass ein größerer *minPTS*-Wert in besseren Ergebnissen resultiert. Trotz der neuen Datenvorverarbeitung werden neun Ausreißer selten erkannt, hierbei handelt es sich um dieselben Ausreißer, die auch schon in Abschnitt 4.2 nicht erkannt wurden.

ϵ	2,3		2,5		2,7		2,9	
minPTS	4	10	4	10	4	10	4	10
True-Negatives	609	601	618	613	622	619	624	623
True-Positives	17	18	13	17	13	17	9	17
False-Negatives	9	8	13	9	13	9	17	9
False-Positives	16	24	7	12	3	6	1	2
Accuracy	0,96	0,95	0,97	0,97	0,98	0,98	0,97	0,98
Recall	0,65	0,69	0,50	0,65	0,50	0,65	0,35	0,65
Precision	0,52	0,43	0,65	0,59	0,81	0,74	0,90	0,89
F-Score	0,58	0,53	0,57	0,62	0,62	0,69	0,50	0,76

Tabelle 6.1: Ergebnisse von DBSCAN auf dem Konto Umsatzerlöse EU

Auf dem Konto *Verbrauch Verpackungen* (Tabelle 6.2) wurden bessere Ergebnisse erzielt. Bei einem *minPTS*-Wert von 10 wurden mehr Ausreißer erkannt. Die doppelte Belegposition wurde nur bei einem ϵ-Wert von 1,6 und einem *minPTS*-Wert von 10 gefunden, jedoch wurden hier viele Belegpositionen fehlerhaft als Ausreißer markiert.

ϵ	1,6		1,9		2,1	
minPTS	4	10	4	10	4	10
True-Negatives	720	691	743	721	749	744
True-Positives	20	27	19	26	15	26
False-Negatives	7	0	8	1	12	1
False-Positives	31	60	8	30	2	7
Accuracy	0,95	0,92	0,98	0,96	0,98	0,99
Recall	0,74	1,00	0,70	0,96	0,56	0,96
Precision	0,39	0,31	0,70	0,46	0,88	0,79
F-Score	0,51	0,47	0,70	0,63	0,68	0,87

Tabelle 6.2: Ergebnisse von DBSCAN auf dem Konto Verbrauch Verpackungen

Bei dem Konto *Umsatzerlöse Inland* weisen alle Parameterkombination eine Accuracy zwischen 95 und 96 Prozent auf. Bei einem zu großen ϵ-Wert werden kaum Ausreißer erkannt, da diese alle Teil eines Clusters geworden sind. Auch hier verbessert sich der Recall bei einem *minPTS*-Wert von 10.

ϵ	2,1		2,3		2,5		2,9	
minPTS	4	10	4	10	4	10	4	10
True-Negatives	6231	6085	6272	6176	6295	6218	6319	6302
True-Positives	73	144	45	98	30	80	15	46
False-Negatives	168	97	196	143	211	161	226	195
False-Positives	116	262	75	171	52	129	28	45
Accuracy	0,96	0,95	0,96	0,95	0,96	0,96	0,96	0,96
Recall	0,30	0,60	0,19	0,41	0,12	0,33	0,06	0,19
Precision	0,39	0,35	0,38	0,36	0,37	0,38	0,35	0,51
F-Score	0,34	0,45	0,25	0,38	0,19	0,36	0,11	0,28

Tabelle 6.3: Ergebnisse von DBSCAN auf dem Konto Umsatzerlöse Inland

6.2 Erprobung von OPTICS

Wahl der Parameter und Anpassungen

Für die neue Erprobung von OPTICS wurden die gewählten Parameter für *maxDistance* verändert. Für die Auswahl geeigneter Distanzen wurde dieselbe Herangehensweise wie in Abschnitt 4.3 gewählt. Hieraus ergeben sich für das Konto *Umsatzerlöse EU* die Werte 2,9, 3,1, 3,2 und 3,4, für das Konto *Verbrauch Verpackungen* 1,5, 1,9, 2,1 und 2,3 und für das Konto *Umsatzerlöse Inland* 3,2, 3,4, 3,5 und 3,7. Der Parameter ϵ wurde bei ∞ belassen, da eine Reduzierung Einfluss auf das Laufzeitverhalten hat. Der *minPTS*-Wert bleibt analog zur vorherigen Erprobung bei 10 auf den Konten *Umsatzerlöse EU* und *Verbrauch Verpackungen* und 40 auf dem Konto *Umsatzerlöse Inland*.

Erreichte Leistungen

Zwischen den einzelnen Werten für *maxDistance* auf dem Konto *Umsatzerlöse EU* gibt es nur geringe Unterschiede in der erreichten Leistung. Im Gegensatz zur vorherigen Erprobung hat sich die Precision auf 0,94 verbessert. Im selben Zug hat sich der Recall jedoch leicht verschlechtert.

maxDistance	2,9	3,1	3,2	3,4
True-Negatives	622	624	624	624
True-Positives	17	17	15	15
False-Negatives	9	9	11	11
False-Positives	3	1	1	1
Accuracy	0,98	0,98	0,98	0,98
Recall	0,65	0,65	0,58	0,58
Precision	0,85	0,94	0,94	0,94
F-Score	0,74	0,77	0,71	0,71

Tabelle 6.4: Ergebnisse von OPTICS auf dem Konto Umsatzerlöse EU

Auf dem Konto *Verbrauch Verpackungen* zeigt OPTICS eine leicht bessere Leistung als bei der ersten Erprobung. Der F-Score steigt auf 0,85 bei einer *maxDistance* von 2,1. Die eine doppelte Buchung wird erst bei einer *maxDistance* von 1,5 erkannt, jedoch muss hier eine niedrige Precision in Kauf genommen werden.

maxDistance	1,5	1,9	2,1	2,3
True-Negatives	690	720	743	748
True-Positives	27	26	26	18
False-Negatives	0	1	1	9
False-Positives	61	31	8	3
Accuracy	0,92	0,96	0,99	0,98
Recall	1,00	0,96	0,96	0,67
Precision	0,31	0,46	0,76	0,86
F-Score	0,47	0,62	0,85	0,75

Tabelle 6.5: Ergebnisse von OPTICS auf dem Konto Verbrauch Verpackungen

Aus der Tabelle 6.6 geht hervor, dass sich Recall-Werte auf dem Konto *Umsatzerlöse Inland* im Vergleich zur ersten Erprobung verschlechtert haben. Der F-Score sinkt dabei nur leicht, da sich die Precision verbessert hat.

maxDistance	3,2	3,4	3,5	3,7
True-Negatives	6217	6250	6315	6325
True-Positives	93	84	76	46
False-Negatives	148	157	165	195
False-Positives	130	97	32	22
Accuracy	0,96	0,96	0,97	0,97
Recall	0,39	0,35	0,32	0,19
Precision	0,42	0,46	0,70	0,68
F-Score	0,40	0,40	0,44	0,30

Tabelle 6.6: Ergebnisse von OPTICS auf dem Konto Umsatzerlöse Inland

6.3 Erprobung von LOF

Wahl der Parameter und Anpassungen

Da sich nach der neuen Datenvorverarbeitung nur die Distanzen zwischen den Datenpunkten verändert hat, werden hier dieselben Parameter für *minPTS* und *contamination* verwendet, da diese unabhängig von der Distanz sind.

Erreichte Leistungen

Die Ergebnisse für LOF auf dem Konto *Umsatzerlöse EU* sind der Tabelle 6.7 zu entnehmen. Die erreichte Leistung von LOF unterscheiden sich nicht nennenswert von den Leistungen aus dem Abschnitt 4.4. Es ist jedoch eine leichte Verbesserung der Precision zu erkennen.

minPTS	10	25	50			
contamination	0,04	0,04	0,04	0,06	0,08	0,10
True-Negatives	612	616	617	604	594	582
True-Positives	5	17	17	18	18	19
False-Negatives	21	9	9	8	8	7
False-Positives	13	9	8	21	31	43
Accuracy	0,95	0,97	0,97	0,96	0,94	0,92
Recall	0,19	0,65	0,65	0,69	0,69	0,73
Precision	0,28	0,65	0,68	0,46	0,37	0,31
F-Score	0,23	0,65	0,67	0,55	0,48	0,43

Tabelle 6.7: Ergebnisse von LOF auf dem Konto Umsatzerlöse EU

Durch die neue Datenvorverarbeitung hat sich die Leistung von LOF leicht verbessert. Die doppelte Belegposition wurde schon bei einer *contamination* von 0,06 erkannt. Da bei diesem Wert bereits alle Ausreißer erkannt wurden, verschlechtert sich durch eine höhere *contamination* nur die Precision. Die detaillierten Ergebnisse sind in Tabelle 6.8 dargestellt.

minPTS	10	25	50			
contamination	0,04	0,04	0,04	0,06	0,08	0,10
True-Negatives	727	743	746	735	720	703
True-Positives	0	20	26	27	27	27
False-Negatives	27	7	1	0	0	0
False-Positives	24	8	5	16	31	48
Accuracy	0,93	0,98	0,99	0,98	0,96	0,94
Recall	0,00	0,74	0,96	1,00	1,00	1,00
Precision	0,00	0,71	0,84	0,63	0,47	0,36
F-Score	0,00	0,73	0,90	0,77	0,64	0,53

Tabelle 6.8: Ergebnisse von LOF auf dem Konto Verbrauch Verpackungen

Auf dem Konto *Umsatzerlöse Inland* haben sich die Ergebnisse von LOF in allen Punkten, unabhängig der Parameter *minPTS* und *contamination*, verschlechtert.

minPTS	40	500			
contamination	0,04	0,04	0,06	0,08	0,10
True-Negatives	6105	6158	6070	5965	5846
True-Positives	6	75	121	150	162
False-Negatives	235	166	120	91	79
False-Positives	242	189	277	382	501
Accuracy	0,93	0,95	0,94	0,93	0,91
Recall	0,02	0,31	0,50	0,62	0,67
Precision	0,02	0,28	0,30	0,28	0,24
F-Score	0,02	0,30	0,38	0,39	0,36

Tabelle 6.9: Ergebnisse von LOF auf dem Konto Umsatzerlöse Inland

6.4 Erprobung von ROCK

Wahl der Parameter und Anpassungen

Um eine Vergleichbarkeit der Ergebnisse aus dem Abschnitt 4.5 zu den neuen Ergebnissen zu wahren, wurden die Parameter für θ, *sizethreshold* und k nicht verändert. Für ϵ wurden die Werte auf Basis des Mittelwerts und der Standardabweichung neu berechnet. Es ergeben sich für das Konto *Umsatzerlöse EU* die Werte 1,7, 2,2, 2,6 und 3,0, für das Konto *Verbrauch Verpackungen* die Werte 1,5, 1,9, 2,1 und 2,3 und für das Konto *Umsatzerlöse Inland* 3,2, 3,4, 3,5 und 3,7.

Erreichte Leistungen

Auf dem Konto *Umsatzerlöse EU* verbessert sich der Recall bei einem ϵ-Wert von 1,7 auf 1,00. Der Recall bleibt bei allen anderen Kombinationen identisch. Des Weiteren ist zu erkennen, dass die Precision sich nur bei einem ϵ-Wert von 2,6 verbessert hat. Bei allen anderen ϵ-Werten verschlechterte sich die Precision. Der beste F-Score sank von 0,65 auf 0,63.

ϵ	1,7		2,2		2,6		3,0	
θ	0,25	0,5	0,25	0,5	0,25	0,5	0,25	0,5
True-Negatives	437	437	571	571	614	614	602	603
True-Positives	26	26	19	19	17	17	18	18
False-Negatives	0	0	7	7	9	9	8	8
False-Positives	188	188	54	54	11	11	23	22
Accuracy	0,71	0,71	0,91	0,91	0,97	0,97	0,95	0,95
Recall	1,00	1,00	0,73	0,73	0,65	0,65	0,69	0,69
Precision	0,12	0,12	0,26	0,26	0,61	0,61	0,44	0,45
F-Score	0,22	0,22	0,38	0,38	0,63	0,63	0,54	0,55

Tabelle 6.10: Ergebnisse von ROCK auf dem Konto Umsatzerlöse EU

Bei der Betrachtung der Ergebnisse von ROCK auf dem Konto *Verbrauch Verpackungen* aus Tabelle 6.11 fällt auf, dass die doppelte Belegposition öfter erkannt wird, jedoch kommt es häufiger zu einer großen Anzahl von False-Positives. Der beste F-Score bleibt bei 0,90.

ϵ	1,5		1,9		2,1		2,3	
θ	0,25	0,5	0,25	0,5	0,25	0,5	0,25	0,5
True-Negatives	708	708	743	683	746	729	713	726
True-Positives	27	27	26	27	26	27	27	27
False-Negatives	0	0	1	0	1	0	0	0
False-Positives	43	43	8	68	5	22	38	25
Accuracy	0,94	0,94	0,99	0,91	0,99	0,97	0,95	0,97
Recall	1,00	1,00	0,96	1,00	0,96	1,00	1,00	1,00
Precision	0,39	0,39	0,76	0,28	0,84	0,55	0,42	0,52
F-Score	0,56	0,56	0,85	0,44	0,90	0,71	0,59	0,68

Tabelle 6.11: Ergebnisse von ROCK auf dem Konto Verbrauch Verpackungen

Bei dem Vergleich der Ergebnisse aus Abschnitt 4.5 mit den Ergebnissen aus der Tabelle 6.12 ist zu erkennen, dass sich der F-Score deutlich verschlechtert hat. Sowohl der Recall als auch die Precision sind bei allen Parameterkombinationen gesunken. Der beste Recall liegt nun bei 0,41 statt 0,68. Die beste Precision sank von 0,71 auf 0,37.

ϵ	3,2		3,4		3,5		3,7	
θ	0,25	0,5	0,25	0,5	0,25	0,5	0,25	0,5
True-Negatives	6233	6221	6266	6206	6144	6205	6149	6182
True-Positives	67	69	121	77	100	83	99	98
False-Negatives	174	172	175	164	141	158	142	143
False-Positives	114	126	66	141	203	142	198	165
Accuracy	0,96	0,95	0,96	0,95	0,95	0,95	0,95	0,95
Recall	0,28	0,29	0,27	0,32	0,41	0,34	0,41	0,41
Precision	0,37	0,35	0,35	0,35	0,33	0,37	0,33	0,37
F-Score	0,32	0,32	0,31	0,34	0,37	0,36	0,37	0,39

Tabelle 6.12: Ergebnisse von ROCK auf dem Konto Umsatzerlöse Inland

6.5 Erprobung von Autoencodern

Wahl der Parameter und Anpassungen

Da sich die Anzahl der Input-Dimensionen nur leicht reduziert hat, war eine Anpassung des Modells nicht notwendig. Auch die Werte für die *contamination* wurden beibehalten, um die erreichten Ergebnisse besser vergleichen zu können.

Erreichte Leistungen

Durch die neue Datenvorverarbeitung werden alle Ausreißer auf dem Konto *Umsatzerlöse EU* schon bei einer *contamination* von 0,08 erkannt. Durch eine weitere Steigerung der *contamination* wird die Precision gesenkt. Bei einer *contamination* von 0,04 ist sowohl die Precision als auch der F-Score besser.

contamination	0,04	0,06	0,08	0,10
True-Negatives	618	606	599	586
True-Positives	19	20	26	26
False-Negatives	7	6	0	0
False-Positives	7	19	26	39
Accuracy	0,98	0,96	0,96	0,94
Recall	0,73	0,77	1,00	1,00
Precision	0,73	0,51	0,50	0,40
F-Score	0,73	0,62	0,67	0,57

Tabelle 6.13: Ergebnisse von Autoencoder auf dem Konto Umsatzerlöse EU

Auf dem Konto *Verbrauch Verpackungen* wurden ab einer *contamination* von 0,08 alle Ausreißer erkannt. Bei einer *contamination* von 0,04 und 0,06 wurde die doppelte Belegposition nicht erkannt.

contamination	0,04	0,06	0,08	0,10
True-Negatives	745	730	715	700
True-Positives	26	26	27	27
False-Negatives	1	1	0	0
False-Positives	6	21	36	51
Accuracy	0,99	0,97	0,95	0,93
Recall	0,96	0,96	1,00	1,00
Precision	0,81	0,55	0,43	0,35
F-Score	0,88	0,70	0,60	0,51

Tabelle 6.14: Ergebnisse von Autoencoder auf dem Konto Verbrauch Verpackungen

Die Tabelle 6.15 enthält die Ergebnisse des Autoencoders auf dem Konto *Umsatzerlöse Inland*. Bei einer *contamination* zwischen 0,04 und 0,06 wurden weniger Ausreißer mit der neuen Datenvorverarbeitung erkannt. Bei einer *contamination* von 0,10 stieg der Recall von 0,61 auf 0,71. Der beste F-Score sank von 0,41 auf 0,38.

contamination	0,04	0,06	0,08	0,10
True-Negatives	6174	6070	5953	5852
True-Positives	93	122	138	170
False-Negatives	148	119	103	71
False-Positives	173	177	394	119
Accuracy	0,95	0,94	0,92	0,91
Recall	0,39	0,51	0,57	0,71
Precision	0,35	0,31	0,26	0,26
F-Score	0,37	0,38	0,36	0,38

Tabelle 6.15: Ergebnisse von Autoencoder auf dem Konto Umsatzerlöse Inland

6.6 Bewertung der Ergebnisse

Durch eine erneute Erprobung der Verfahren auf Basis der verbesserten Datenvorverarbeitung konnten die erreichten Ergebnisse auf den Konten *Umsatzerlöse EU* und *Verbrauch Verpackungen* verbessert werden. ROCK bildet eine Ausnahme, da sich trotz der neuen Vorverarbeitung die Ergebnisse auf allen Konten leicht verschlechtert haben. Entgegen der Erwartung führte die verbesserte Datenvorverarbeitung auf dem Konto *Umsatzerlöse Inland* zu teils deutlichen Verschlechterungen bei allen Verfahren. Es ist zu vermuten, dass durch die neue Datenvorverarbeitung Attribute hinzugekommen sind, die nicht für die Erkennung der auffälligen Belegpositionen hilfreich waren. Dadurch haben sich die Distanzen zwischen unauffälligen Belegpositionen vergrößert und die Clusterbildung hat sich verändert. Diese neue Clusterbildung führt zu einer Verschlechterung der Metriken im Allgemeinen.

Der beste F-Score von 0,77 auf dem Konto *Umsatzerlöse EU* wurde von OPTICS erreicht. LOF erreichte auf dem Konto *Verbrauch Verpackungen* den besten F-Score von 0,90. Auf dem Konto *Umsatzerlöse Inland* wurde der beste F-Score von 0,45 DBSCAN erreicht. Die maximalen F-Scores der Verfahren liegen jedoch wenige Prozentpunkte auseinander. Es ist hervorzuheben, dass der Autoencoder auf den Konten *Umsatzerlöse EU* und *Verbrauch Verpackungen* alle Ausreißer erfolgreich identifizieren konnte und dabei eine geringe False-Positives-Rate aufwies.

7 Fazit

In diesem Kapitel werden die Ergebnisse dieser Arbeit zusammengefasst. Im Abschnitt 7.2 werden Möglichkeiten beschrieben, auf welche Weise die Analyse des Datensatzes verändert werden kann, um die Ergebnisse weiter zu verbessern.

7.1 Zusammenfassung

Ziel dieser Arbeit war es zu prüfen, ob bekannte Verfahren aus den Bereichen Clustering und maschinelles Lernen sinnvoll in der Wirtschaftsprüfung eingesetzt werden können. Der vorliegende Datensatz wurde in seiner Struktur beschrieben und analysiert. Auf Basis dieser Analysen wurden geeignete Attribute für die folgende Datenvorverarbeitung ausgewählt. Ziel der Datenvorverarbeitung war es die Ausprägungen der Attribute, die in Form von Zeichenketten vorlagen, in numerische Werte umzuwandeln, ohne die Bedeutung der Ausprägungen zu verfälschen. In der ersten Erprobungsphase wurden die fünf Verfahren DBSCAN, OPTICS, LOF, ROCK und Autoencoder eingesetzt, um die auffälligen Buchungspositionen zu identifizieren. Hierzu wurde eine Erwartung zu jedem Verfahren definiert und anschließend die Wahl der Parameter argumentiert. Die Erprobung hat ergeben, dass die Verfahren auf jedem Konto ähnliche Ergebnisse erzielt haben. Auf dem Konto *Verbrauch Verpackungen* wurden bei allen Verfahren die besten und auf dem Konto *Umsatzerlöse Inland* die schlechtesten Ergebnisse erreicht.

Mithilfe einer zweiten Datenvorverarbeitung wurde ein Versuch unternommen die Ergebnisse zu verbessern. Hierzu wurden weitere Informationen aus dem Datensatz extrahiert und die Reihenfolge ordinaler Attribute wurde kodiert. Eine erneute Erprobung der Verfahren hat ergeben, dass die Qualität der Ausreißer-Erkennung auf den Konten *Umsatzerlöse EU* und *Verbrauch Verpackungen* zugenommen hat. Die Ergebnisse auf dem Konto *Umsatzerlöse Inland* haben sich, entgegen der Erwartung, leicht verschlechtert. Während der beiden Erprobungen fielen regelmäßig wiederkehrende Belegpositionen auf,

die selten erfolgreich erkannt wurden. Diese scheinen sich nur geringfügig von anderen Belegpositionen zu unterscheiden.

Es hat sich zudem gezeigt, dass die sorgfältige Auswahl der Attribute eine große Auswirkung auf die Qualität der Ergebnisse besitzt. In einer früheren Arbeit [ST20], die denselben Datensatz analysiert hat, wurden auf dem Konto *Umsatzerlöse Inland* mit einer anderen Auswahl an Attributen bessere Ergebnisse erzielt. Die Ergebnisse auf den Konten *Umsatzerlöse EU* und *Verbrauch Verpackungen* konnten reproduziert und verbessert werden.

7.2 Mögliche zukünftige Verbesserungen und Ausblick

Da die Qualität der Datenvorverarbeitung einen großen Einfluss auf die Qualität der Ausreißer-Erkennung hat, ist es sinnvoll mehr Expertenwissen in der Datenvorverarbeitung einzusetzen. Hierzu ist es sinnvoll die Auswahl der Attribute nur auf solche zu beschränken, die nach Einschätzung eines Experten, relevant für die Erkennung auffälliger Buchungspositionen sind [Byr15]. Ebenso ist es für distanzbasierte Verfahren vorteilhaft, jedem Attribut eine individuelle Gewichtung in Abhängigkeit ihrer Bedeutung zuzuweisen. Auf diese Weise könnte verhindert werden, dass weniger bedeutende Attribute die Datenpunkte fälschlicherweise auseinanderziehen.

Bei der bisherigen Datenvorverarbeitung wird nur geringfügig auf die Belegpositionen anderer Konten eingegangen. Für die Bewertung einer Position können die Bewertungen der übrigen Positionen eingebunden werden. Wenn andere Belegpositionen auf dem Beleg auffällig sind, so ist es wahrscheinlich, dass andere Transaktionen auf demselben Beleg auch auffällig sind.

ROCK verwendet für die Berechnung der Ähnlichkeit zweier Datenpunkte die L2-Norm (Euklidische Norm). Ein alternativer Ansatz ist die Verwendung einer nicht metrischen Ähnlichkeitsfunktion [GRS99]. Ein Experte könnte so präziser bestimmen nach welchen Kriterien Datenpunkte ähnlicher sind. Das Ergebnis wären aussagekräftigere Cluster und eine bessere Erkennung von Ausreißern.

Bei bisherigen Studien wurden, bis auf wenige Ausnahmen, nur eine Instanz des Verfahrens verwendet, um Datenpunkte als Ausreißer zu klassifizieren [NG21]. Alternativ wäre es möglich mehrere Instanzen eines Verfahrens zu verwenden und für die Klassifizierung

eines Datenpunktes die Ergebnisse der einzelnen Verfahren zu kombinieren. Dieser Ansatz wurde schon erfolgreich mit Autoencodern [CSAT17] und mit DBSCAN und LOF [KGMV14] verfolgt. Zusätzlich ist es möglich Erweiterungen bisheriger Verfahren zu erproben. An dieser Stelle sind HDBSCAN [MHA17], OPTICS-OF [BKNS99] und CBLOF [HXD03] zu nennen.

Literaturverzeichnis

[ABpKS99] Mihael Ankerst, Markus M. Breunig, Hans peter Kriegel, and Jörg Sander. Optics: Ordering points to identify the clustering structure. pages 49–60. ACM Press, 1999.

[AKV17] Deniz Appelbaum, Alexander Kogan, and Miklos Vasarhelyi. Big data and analytics in the modern audit engagement: Research needs. *AUDITING: A Journal of Practice and Theory*, 36, 02 2017.

[BAAG+18] Paul Byrnes, Abdullah Al-Awadhi, Benita Gullkvist, Helen Brown-Liburd, Ryan Teeter, J. Warren, and Miklos Vasarhelyi. *Evolution of Auditing: From the Traditional Approach to the Future Audit: Theory and Application*, pages 285–297. Emerald Group PUB, 03 2018.

[BKNS99] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Optics-of: Identifying local outliers. In Jan M. Żytkow and Jan Rauch, editors, *Principles of Data Mining and Knowledge Discovery*, pages 262–270, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[BKNS00] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: Identifying density-based local outliers. *SIGMOD Rec.*, 29(2):93–104, May 2000.

[Byr15] P. E. Byrnes. Developing automated applications for clustering and outlier detection. 2015.

[CSAT17] Jinghui Chen, Saket Sathe, Charu Aggarwal, and Deepak Turaga. *Outlier Detection with Autoencoder Ensembles*, pages 90–98. 2017.

[EKSX96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press, 1996.

[Fle18] Holger Fleischer. *Handelsgesetzbuch*. Beck-Texte im dtv. dtv, München, Sonderausgabe, 62., überabeitete Auflage, Stand: 7. Dezember 2017 edition, 2018.

[GRS99] S. Guha, R. Rastogi, and K. Shim. Rock: a robust clustering algorithm for categorical attributes. In *Proceedings 15th International Conference on Data Engineering (Cat. No.99CB36337)*, pages 512–521, 1999.

[Haw80] D. M. Hawkins. *Identification of Outliers*. Springer Netherlands, Dordrecht, 1980.

[HHWB02] Simon Hawkins, Hongxing He, Graham Williams, and Rohan Baxter. Outlier detection using replicator neural networks. In Yahiko Kambayashi, Werner Winiwarter, and Masatoshi Arikawa, editors, *Data Warehousing and Knowledge Discovery*, pages 170–180, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[HXD03] Zengyou He, Xiaofei Xu, and Shengchun Deng. Discovering cluster-based local outliers. *Pattern Recognition Letters*, 24(9):1641–1650, 2003.

[KGMV14] H.D. Kuna, R. García-Martinez, and F.R. Villatoro. Outlier detection in audit logs for application systems. *Information Systems*, 44:22–33, 2014.

[KS16] W. R. Knechel and S. E. Salterio. *Auditing: Assurance and risk*. Routledge, 2016.

[MHA17] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11), March 2017.

[NG21] Jakob Nonnenmacher and Jorge Gómez. Unsupervised anomaly detection for internal auditing: Literature review and research agenda. *The International Journal of Digital Accounting Research*, pages 1–22, 01 2021.

[NLHL19] Won No, Kyungha Lee, Feiqi Huang, and Qiao Li. Multidimensional audit data selection (mads): A framework for using data analytics in audit data selection process. *Accounting Horizons*, 33, 05 2019.

[Nov19] Andrei Novikov. Pyclustering: Data mining library. *Journal of Open Source Software*, 4(36):1230, apr 2019.

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[SAP20] SAP. Sap, 2020 annual report. https://www.sap.com/docs/download/investors/2020/sap-2020-annual-report-form-20f.pdf, December 2020.

[SMG19] N Suri, M. Murty, and Athithan Gopalasamy. *Outlier Detection: Techniques and Application A Data Mining Perspective*. Springer, January 2019.

[SSB+18] Marco Schreyer, Timur Sattarov, Damian Borth, Andreas Dengel, and Bernd Reimer. Detection of anomalies in large scale accounting data using deep autoencoder networks, 2018.

[ST20] Martin Schultz and Marina Tropmann-Frick. Autoencoder neural networks versus external auditors: Detecting unusual journal entries in financial statement audits. In *53rd Hawaii International Conference on System Sciences, HICSS 2020, Maui, Hawaii, USA, January 7-10, 2020*, pages 1–10. ScholarSpace, 2020.

[ZNL19] Yue Zhao, Zain Nasrullah, and Zheng Li. Pyod: A python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, 20(96):1–7, 2019.

A Anhang

A.1 Datensatzbeschreibung

Attributname	Attributbeschreibung
AccountingYear	Geschäftsjahr
AccountingPeriod	Buchungsperiode (1-16)
docno	eindeutige ID für einen Buchungsbeleg
DocPos	ID für die Zeile eines Buchungsbelegs
doctype	Belegart (z.B. Ausgangsrechnung, Zahlung, Warenbewegung etc.)
postingkey	Buchungsschlüssel
PostingDate	Buchungsdatum (Zuordnung zu einer Buchungsperiode)
DocDate	Datum auf dem zugrundeliegenden "physischen"Beleg
CreationDate	Datum der Erzeugung des Belegs
DatesEqual	Gibt an, ob alle drei Datumsangaben identisch sind
OneDateOutsideAccountYear	Gibt an, ob eine der Datumsangaben außerhalb des Geschäftsjahres liegt
postingCloseToFiscalYearEnd	Gibt an, ob das PostingDate nahe am Geschäftsjahresende ist
creationBeforPosting	Gibt an, ob CreationDate vor PostingDate liegt
CreationDateDayOfWeek	Wochentag der Erzeugung des Belegs
accounttype	Art des Kontos (Sach-, Debitoren-, Kreditoren-, Material- oder Anlagevermögen-Konto)
accountno	Kontonummer der Belegposition/ zeile
accountname	Kontoname der Belegposition/- zeile
Currency	Währung (im Datensatz immer anonymisiert auf CURR)
DebitCreditIndicator	Soll-/Habenkennzeichen (D = Debit = Soll, C = Credit = Haben)

AmountLocalCurrency	absoluter Betrag
taxcode	Steuercode (z.B. Vorsteuer, Umsatzsteuer)
RevenueIndicator	Indikator ob die Buchung umsatzwirksam ist
userid	ID des Erstellers des Belegs
UserGroup	Gruppe/ Art des Erstellers (user, system, service)
debit_accountno_list	Liste alle Kontonummern auf der Soll-Seite des Belegs
debit_accountname_list	Liste alle Kontennamen auf der Soll-Seite des Belegs
credit_accountno_list	Liste alle Kontonummern auf der Haben-Seite des Belegs
credit_accountname_list	Liste alle Kontennamen auf der Haben-Seite des Belegs
DocNotes	Gibt an, ob eine Belegtext vorhanden ist
DocNotes_Detailed	Belegtext (Anmerkungen des Erstellers zum Beleg)
DocPosNotes	Gibt an, ob eine Belegpositionstext vorhanden ist
DocPosNotes_Detailed	Belegpositionstext (Anmerkungen des Erstellers zur Belegposition)
CredDebNumberList	Liste aller Debitoren-/Kreditorennummern des Belegs
creditornamelist	Liste aller Kreditorennamen des Belegs
debitornameList	Liste aller Debitorennamen des Belegs
DebCredCountryList	Liste aller Debitoren-/Kreditorenländer des Belegs
tcode	Systemfunktion, mit der der Beleg ertellt wurde (SAP Kürzel, z.B. MIGO = Wareneingang buchen)
tcodename	Langname der Systemfunktion, mit der der Beleg ertellt wurde (SAP Kürzel, z.B. MIGO = Wareneingang buchen)
recurringDocNo	Dauerbuchungsnummer
reversalDocNo	Ist mit der Belegnummer der Stornobuchung gefüllt, sofern die Buchung storniert wurde
reversalDocNoTrueFalse	
DocOrigin	Belegherkunft (spezifisch SAP, aus welchem Modul kommt die Buchung)

Tabelle A.1: Erläuterung der Attribute in dem Datensatz

A.2 Distanzen nach der ersten Datenvorverarbeitung

Distanz	Vorkommen in Umsatzerlöse EU	Vorkommen in Verbrauch Verpackungen	Vorkommen in Umsatzerlöse Inland
0,0000	114	897	9.118
1,4142	2.049	14.849	77.065
2,0000	7.231	70.218	279.291
2,4495	16.424	121.079	823.380
2,8284	21.867	69.460	1.895.628
3,1623	27.956	5.221	4.533.069
3,4641	48.037	6.380	6.371.990
3,7417	50.550	6.168	4.294.374
4,0000	24.408	3.986	1.775.693
4,2426	5.727	872	896.113
4,4721	2.605	42	457.787
4,6904	1.235	237	332.412
4,8990	2.151	833	215.283
5,0990	1.183	621	69.099
5,2915	31	703	26.298
5,4772	7	674	4.599
5,6569	0	13	204

Tabelle A.2: Abstände zwischen den Datenpunkten nach der erste Datenvorverarbeitung

A.3 Distanzen nach der zweiten Datenvorverarbeitung

Distanz	Vorkommen in Umsatzerlöse EU	Vorkommen in Verbrauch Verpackungen	Vorkommen in Umsatzerlöse Inland
0,0000	91	888	27.317
1,0000	407	3.568	47.968
1,4142	1.680	8.391	97.367
1,7321	2.506	14.600	144.759
2,0000	3.780	21.437	201.547
2,2361	4.907	26.211	236.972
2,4495	5.582	28.704	247.573
2,6458	6.028	29.667	235.351
2,8284	6.433	27.843	233.786
3,0000	8.028	25.415	228.748
3,1623	9.203	21.907	252.563
3,3166	10.274	18.580	309.220
3,4641	11.433	15.233	410.529
3,6056	11.907	12.036	566.544
3,7417	12.746	9.383	740.552
3,8730	13.026	7.229	908.885
4,0000	12.581	5.913	1.071.901
4,1231	12.719	4.670	1.202.338
4,2426	11.936	3.760	1.297.829
4,3589	10.712	3.118	1.347.817
4,4721	9.671	2.508	1.334.783
4,5826	8.574	2.087	1.283.012
4,6904	7.065	1.663	1.212.797
4,7958	5.861	1.329	1.127.425
4,8990	5.041	982	1.019.342
5,0000	4.031	709	906.940
5,0990	3.237	550	793.755
5,1962	2.556	352	695.154
5,2915	2.052	291	586.903
5,3852	1.744	250	489.965
5,4772	1.240	245	404.579

5,5678	975	228	332.850
5,6569	698	244	273.420
5,7446	587	266	231.756
5,8310	506	303	201.278
5,9161	392	296	181.521
6,0000	323	281	160.108
6,0828	261	269	148.023
6,1644	218	205	131.814
6,2450	179	188	118.896
6,3246	146	120	107.345
6,4031	123	96	96.146
6,4807	54	85	84.967
6,5574	33	62	72.561
6,6332	16	34	59.892
6,7082	6	20	49.218
6,7823	2	20	39.009
6,8557	1	7	31.008
6,9282	1	4	23.241
7,0000	3	3	17.909
7,0711	0	1	12.268
7,1414	0	1	8.957
7,2111	0	1	5.881
7,2801	0	0	3.939
7,3485	0	0	2.379
7,4162	0	0	1.395
7,4833	0	0	771
7,5498	0	0	394
7,6158	0	0	169
7,6811	0	0	46
7,7460	0	0	18
7,8102	0	0	3

Tabelle A.3: Abstände zwischen den Datenpunkten nach der zweiten Datenvorverarbeitung

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