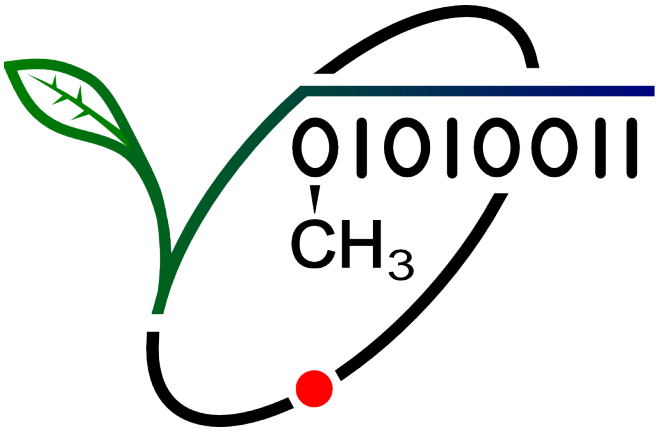
Spezialschulteil des Albert-Schweitzer-Gymnasiums Erfurt



Automatisierte Multiobjektclusterung mithilfe von Machine-Learning-Verfahren

Methoden der optimalen Clusterung verschiedener Datensätze mit Hilfe eines neuronalen Netzes

Gruppenmitglieder:	Hagen Jacob (18), Jan Edgar König (18), Robert Vetter (18)
Erarbeitungsort:	Spezialschulteil des Albert-Schweitzer-Gymnasiums in Erfurt
Projektbetreuer:	Herr Johannes Süpke
Fachgebiet:	Mathematik / Informatik
Wettbewerbssparte:	Jugend forscht
Bundesland:	Thüringen
Wettbewerbsjahr:	2024
Link zum Quellcode:	*Hier klicken*

Detaillierter Projektüberblick zum besseren Verständnis des Projektes

In einer zunehmend digitalisierten Welt spielt die Automatisierung von Prozessen eine immer wichtigere Rolle. Aktuell wird die Analyse von Informationen oft noch händisch von Datenanalysten durchgeführt, die zahlreiche Datenpunkte in Diagrammen korrekt ihren jeweiligen Gruppierungen zuordnen müssen. Diese Zuweisung lässt sich heutzutage in nahezu allen Lebensbereichen wiederfinden: Vom Marketing, wo Kundenverhalten analysiert wird, über die Finanzwelt, in welcher Datenanalysten riesige Mengen an Marktdaten verarbeiten, bis hin zur Medizin, wo die Erkennung von Krankheitsbildern bei Patienten durch eine solche Klassifikation unterstützt wird. Da diese manuelle Gruppierung jedoch sehr zeitintensiv sein kann, ist es wesentlich effizienter, Verfahren zu entwickeln und einzusetzen, die automatisiert eine sinnvolle Klassifizierung dieser Informationen durchführen.

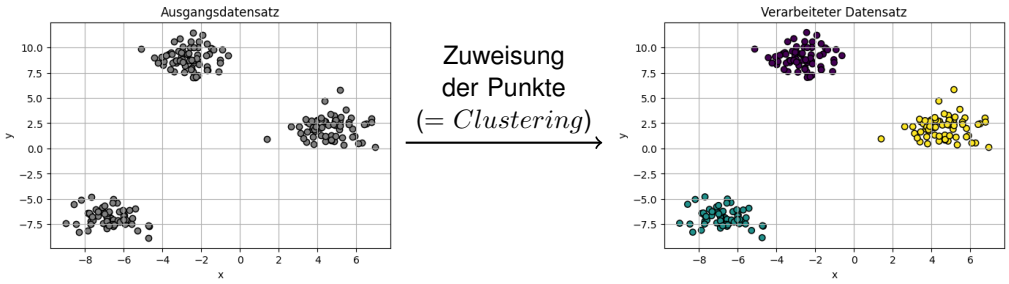


Abbildung 1: Vergleich von ungeclusterten und geclusterten Datenpunkten

Clustering-Algorithmen sind solche Verfahren, die es ermöglichen, Datenmengen mit unterschiedlichen Eigenschaften zu analysieren und in Gruppen einzuteilen. Dabei clustert jeder Algorithmus einen vorliegenden Datensatz unterschiedlich gut. Grundlegend haben wir im Zuge dieser Arbeit drei große Ziele erreicht, die sich hauptsächlich auf vollständig neue Ansätze begründen:

1. Erkennung und Extraktion von Ausreißern (also Messfehlern) in den Daten
2. Erkennung der Form und Kontur einer gegebenen Punktemenge
3. Bestimmung des am besten geeigneten Clustering-Algorithmus für einen gegebenen Datensatz

Somit haben wir sowohl für Schritt zwei, als auch für Schritt drei im folgenden Ablauf eine automatisierte Lösung entwickelt:

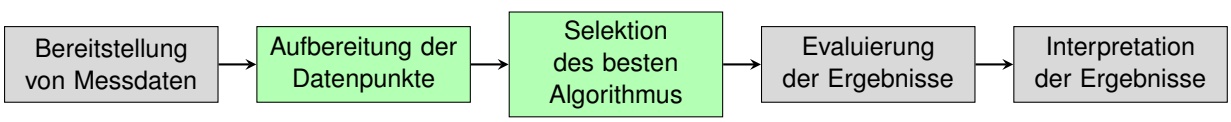


Abbildung 2: Prozessablauf bei der Clusteranalyse

Die Idee, eine Arbeit über das Themenfeld Clustering zu schreiben, haben wir unserem Außenbetreuer vom Fraunhofer-Institut IIS, Dr. Wolfgang Felber, zu verdanken.

Inhaltsverzeichnis

1 Fachliche Kurzfassung **2**

2 Motivation und Fragestellung **3**

3 Hintergrund und theoretische Grundlagen **4**

4 Vorgehensweise, Materialien und Methoden **6**

4.1 Schwerpunktsetzung der Arbeit 6

4.2 Detektion und Analyse von Anomalien in den Datensätzen 6

4.3 Bestimmung der Form des Datensatzes 8

4.4 Selektion weiterer Features zur exakten Charakterisierung des Datensatzes 10

4.5 Generation von Daten und Trainingsprozess des neuronalen Netzes 12

5 Ergebnisse **12**

6 Ergebnisdiskussion **14**

7 Fazit und Ausblick **16**

8 Literaturverzeichnis **17**

9 Abbildungsverzeichnis **18**

10 Unterstützungsleistungen **19**

1 Fachliche Kurzfassung

Nachdem wir uns mit den theoretischen Grundlagen auseinandergesetzt haben, widmeten wir uns der Detektion von Ausreißern. Da diese in der Regel Messfehler sind, verfälschen sie die tatsächlichen Daten und müssen daher extrahiert werden. Hierbei entwickelten wir ein Verfahren, das alle Ausreißer in einem Datensatz automatisiert über folgende zwei Methoden erkennt:

1. Den Clustering-Algorithmus $DBSCAN$
2. Die Ausreißererkennungsmethode $Local - Outlier - Factor$ (kurz LOF)

Dabei fiel uns auf, dass sich beide Verfahren gegenseitig sehr gut ergänzen, weswegen wir die Ergebnisse dieser kombinierten. Darüber hinaus stellt die Optimierung der Paramater beider Algorithmen nahezu eine eigene Arbeit für sich dar. Dennoch möchten wir sie in dieser Arbeit nur kurz anreißen, da wir auf 15 Seiten beschränkt sind. Diese Extraktion der Messfehler ist jedoch lediglich die Vorverarbeitung der Daten. Um den am besten geeigneten Clustering-Algorithmus zu bestimmen, entschieden wir uns für die Entwicklung eines neuronalen Netzes, welches charakteristische Eigenschaften der Datensätze ($= Features$) als Eingabe erhält. Somit kann klar vorhergesagt werden, welcher Algorithmus bei welchem Datensatz als geeignet erscheint. Hier trägt die Erkennung der Form der Punktemenge einen maßgeblichen Anteil zur Qualität der Vorhersage bei. In Zuge dessen entwickelten wir zwei Methoden, welche beide die Konturen eines Datensatzes ermitteln. Anschließend wird sich für die Bessere entschieden. Aus dieser lassen sich wiederum Features ableiten, die als Eingabe für unser neuronales Netz verwendet werden. Im Folgenden sind einige von Ihnen aufgelistet:

- Exzentrizität der Kontur
- Verhältnis der konvexen Hülle zur Konturfläche
- Kompaktheit der Datenpunkte
- Kurtosis und Schiefe der Datenpunkte

Anschließend generierten wir 5184 synthetische Datensätze und änderten zusätzlich die Punkteanordnung der generierten Datensätze leicht ab, um eine höhere Variabilität der Datensätze zu gewährleisten und somit unser neuronales Netz robuster zu machen. Dieses erkennt den korrekten Clustering-Algorithmus bei einem synthetischen Eingabedatensatz mit einer Wahrscheinlichkeit von $99,81\%$. Es arbeitet jedoch nicht nur bei synthetischen Datensätzen sehr genau, sondern auch bei realen Messdaten. Entsprechend kann das System auch noch mit mehr Features versehen werden bzw. mit anderen Trainingsdaten trainiert werden, wodurch auch die Anwendung im alltäglichen Gebrauch nicht weit entfernt scheint.

2 Motivation und Fragestellung

In einer zunehmend digitalisierten Welt, in der automatisierte Prozesse die Datenanalyse dominieren, liegt eine entscheidende Herausforderung in der manuellen Auswahl des geeigneten Clustering-Algorithmus. Trotz der erfolgreichen Automatisierung des eigentlichen Clustering-Prozesses, bleibt das Problem der Detektion von Ausreißern sowie die Entscheidung für das optimale Verfahren eine komplexe Aufgabe für Datenanalysten und Data Scientists.
Die Fragestellung unserer Forschung konzentriert sich auf die Möglichkeit der Entwicklung eines Systems, das eigenständig den bestgeeigneten Clustering-Algorithmus für einen gegebenen Datensatz auswählt. Können wir ein solches Instrument schaffen, das nicht nur die Genauigkeit der Datenklassifizierung verbessert und die Arbeitsbelastung von Analytikern drastisch reduziert, sondern auch eine Art Beweis liefern, dass sich eine solche Selektion automatisieren lässt? Falls sich dies als möglich herausstellen sollte, wäre des Weiteren zu klären, mit welcher Genauigkeit das Produkt arbeiten könnte. Die Schaffung eines automatisierten Entscheidungsprozesses für die Auswahl von Clustering-Verfahren eröffnet nicht nur effizientere Datenanalysen, sondern auch eine Vielzahl von Möglichkeiten:

- Durch die Automatisierung der Auswahl können Datenanalysten Zeit sparen und sich vermehrt auf die Interpretation der Analyseergebnisse konzentrieren, anstatt sich mit der manuellen Auswahl von Clustering-Methoden zu befassen.
- Ein intelligentes System, das den besten Algorithmus für bestimmte Datensätze identifiziert, kann die Genauigkeit der Klassifizierung verbessern und somit zur präziseren Analyseergebnissen führen.
- Die Möglichkeit, sich automatisch an unterschiedliche Datenszenarien anzupassen, macht das System flexibel und vielseitig einsetzbar, unabhängig von der Komplexität der vorliegenden Daten. Es ist beliebig erweiterbar und somit auch in der Lage, sich an eine Vielzahl von Datensätzen anzupassen.
- Automatisierte Auswahlprozesse könnten auch dazu beitragen, bislang unentdeckte Muster oder Trends in den Daten zu identifizieren, was zu neuen Erkenntnissen und innovativen Ansätzen führen kann.

Die Schaffung eines solchen automatisierten Entscheidungsprozess markiert nicht nur einen Fortschritt in der Datenanalyse, sondern öffnet auch Türen zu neuen Möglichkeiten und Erkenntnissen, die weit über die bisherigen manuellen Ansätze hinausgehen. Auf unser aktuelles Forschungsthema sind wir bei der Auseinandersetzung mit dem Problem des Clusterings im Zuge unserer Projektarbeit gekommen und haben direkt das Potenzial erkannt, dass ein solches System besitzen könnte.

3 Hintergrund und theoretische Grundlagen

Im Folgenden sollen die für diese Ausarbeitung wichtigsten Clusteringalgorithmen kurz vorgestellt werden. Hierbei sind vier Clustering-Algorithmen besonders relevant, da das bereits erwähnte neuronale Netz aus ihnen den besten auswählen soll.

K-Means ist ein sehr effizienter und einfacher Algorithmus, der auf der Zuordnung von Schwerpunkten für die jeweiligen Cluster basiert. Ein großer Nachteil von K-Means ist, dass damit nur Datensätze mit Clustern, die eine eher kreisförmige Kontur besitzen, sinnvoll analysiert werden können. Längliche oder ineinander verschlungene Cluster können nur schwerlich durch Schwerpunkte voneinander abgetrennt werden.[1]

DBSCAN ist ein dichtebasierter Algorithmus. Das bedeutet, dass größere Mengen an Datenpunkten, welche nah aneinander sind, jeweils zu Clustern zusammengefasst werden. Übrig gebliebene Datenpunkte können dann als Ausreißer identifiziert werden. Der Vorteil ist hierbei, dass die Form der Cluster nicht erfolgsentscheidend ist. Nachteilig ist jedoch die - im Vergleich zu K-Means - erhöhte Laufzeit.[2]

Gaussian-Mixture-Models (GMMs) sind stochastische Modelle, mit denen für jeden Punkt in einem Datensatz die Wahrscheinlichkeiten für die Zugehörigkeit zu jedem Cluster ermittelt werden. Der Vorteil ist hierbei, dass anders als bei den eben genannten Algorithmen auch sich überlappende Cluster gefunden werden können. Nachteile bestehen in einer relativ hohen Komplexität und darin, dass eine erhöhte Anzahl an Datenpunkten notwendig ist, um gute Ergebnisse zu erhalten. [3]

Beim **hierarchischen Clustering** werden zunächst alle Punkte als einzelne Cluster aufgefasst und dann nach und nach zu immer größeren Gruppen zusammengefügt. Als Grundlage dafür dient der Abstand der Gruppen zueinander. Die Vorteile sind hierbei, dass sich das Verfahren gut für hierarchische Organisationsstrukturen, wie z.B. in Unternehmen, eignet und auf einem simplen Konzept beruht. Die sehr hohe Laufzeit macht den Algorithmus jedoch nur für geringe Datenmengen praktikabel. [4]

Wie zu erkennen ist, haben die Algorithmen verschiedenste Vor- und Nachteile (siehe Abb. 3). Es gibt eine Vielzahl weiterer Methoden, jedoch wurden diese vier ausgewählt, da sie durch ihre komplementären Eigenschaften ein sehr breites Spektrum an Datenerscheinungen abdecken.

Ein weiterer wichtiger Aspekt des maschinellen Lernens ist die Verwendung von Evaluierungsmetriken zur qualitativen Bewertung des Clusterings. Ein Beispiel für eine solche Metrik ist der Silhouette-Score, welcher eine verhältnismäßige Aussage über die Abstände der Punkte innerhalb eines Cluster und zwischen den Clustern selbst trifft.[5]

---

[1] vgl. Raschka, Sebastian; Mirjalili, Vahid: *Python Machine Learning - Machine Learning and Deep Learning with Python, scikit-learn, and Tensorflow 2*, 3. Aufl., Birmingham (Vereinigtes Königreich), Packt, 2019

[2] vgl. Chauhan, Nagesh Singh: *DBSCAN Clustering Algorithm in Machine Learning*, `https://www.kdnuggets.com/2020/04/dbscan-clustering-algorithm-machine-learning.html` [Zugriff am 30.10.2023]

[3] vgl. Dadi, Harihara; Venkatesh, P.; Poornesh, P.; L., Narayana Rao; Kumar, N. : *Tracking Multiple Moving Objects Using Gaussian Mixture Model* [Zugriff am 30.10.2023]

[4] vgl. Sultana, Shaik Irfana: *How the Hierarchical Clustering Algorithm Works*, `https://dataaspirant.com/hierarchical-clustering-algorithm` [Zugriff am 30.10.2023]

[5] vgl. Gaido, Marco: *Distributed Silhouette Algorithm: Evaluating Clustering on Big Data*, `https://arxiv.org/pdf/2303.14102.pdf` [Zugriff am 30.12.2022]

In der folgenden Grafik ist die Performance verschiedener Algorithmen an entsprechenden synthetischen Datensätzen abgebildet:

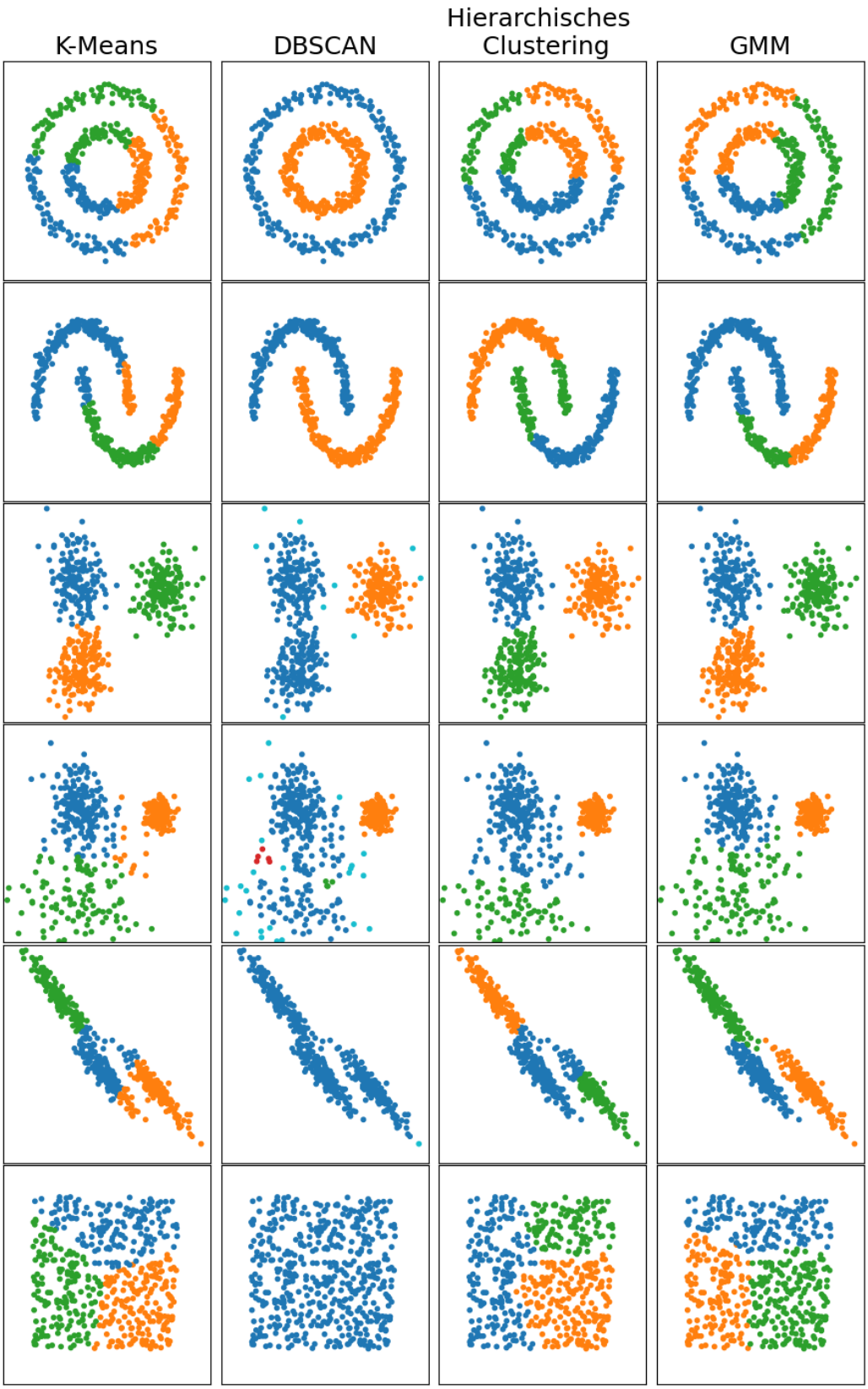


Abbildung 3: Vergleich verschiedener Algorithmen an verschiedenen Punkteverteilungen (Namen der Verteilungen von oben nach unten: $make\_circles$, $make\_moons$, $make\_blobs$,$make\_gaussian\_quantiles$, $make\_blobs$ (andere Parameter), $no\_structure$)

4 Vorgehensweise, Materialien und Methoden

4.1 Schwerpunktsetzung der Arbeit

Bei der Entwicklung eines neuronalen Netzes liegt der Schwerpunkt häufig auf der Extraktion charakteristischer Merkmale der Daten, wobei das Endresultat in der Regel lediglich im besten Fall ein System ist, welches eine Vorhersage über einen konkreten Sachverhalt treffen kann. Somit ist auch bei uns der Weg zu einem solchem System der Hauptbestandteil der Arbeit. Insbesondere Sachverhalte, die wir in unserer Arbeit behandelt haben, so jedoch nach unserem Kenntnisstand noch nie irgendwo anders schriftlich festgehalten wurden, erfordern ein tiefgreifendes Verständnis und im Zuge dessen auch einen ausführlichen Lösungsweg. Dieses hat zur Folge, dass das aktuelle Kapitel weitaus länger als die empfohlenen zwei Seiten ist und das Kapitel zu den Ergebnissen entsprechend kürzer ist. Des Weiteren möchten wir in der folgenden Arbeit unser System an folgendem Beispiel mit Praxisbezug erläutern. Dabei ist links ein Ausschnitt aus einer Verkehrsaufnahme eines Kraftfahrzeuges zu sehen, wobei im folgenden Bild Merkmale gelabelt wurden, welche wiederum als Datenpunkte in ein Koordinatensystem extrahiert wurden. Grundlegend beschäftigen wir uns nur mit der Punktemenge (rechte Abbildung) und nicht mit den Schritten, die zur Aufnahme von diesen Punkten passiert sind (linke und mittlere Abbildung).

Abbildung 4: Prozess der Datenaufnahme bis hin zur Darstellung im Koordinatensystem

Hierbei muss jedoch beachtet werden, dass diese Weiterverarbeitung von Bilderkennungsdaten am Beispiel des autonomen Fahrens nur ein mögliches Beispiel ist. Unser System funktioniert mit allen Arten von Punkteverteilungen, seien es synthethische Datensätze (siehe Abb. 3) oder reale Datensätze. Wir werden unser System jedoch zum besseren Verständnis im Folgenden immer an diesem Beispiel erklären.

4.2 Detektion und Analyse von Anomalien in den Datensätzen

Anomalien oder Ausreißer sind Datenpunkte, die sich deutlich von der Mehrheit der anderen Datenpunkte in einem Datensatz unterscheiden. Anomalien passen häufig nicht zu den Clustern und liegen daher weit entfernt von allen Clusterzentren. Die Detektion von Ausreißern ist zurzeit Gegenstand aktueller Forschung. Hier wurde von uns eine Lösung entwickelt, welche unseren Anforderungen genügt und in der Lage ist, einen Großteil der Ausreißer korrekt zu identifizieren. Hierbei konnte auf das Vorwissen von den verschiedenen Clustering-Algorithmen zurückgegriffen werden: Der DBSCAN-Algorithmus ist mit den richtigen Eingabeparametern in der Lage, solche Ausreißer zu identifizieren. Ihm werden als Hyperparameter ϵ und $minPts$ zugewiesen. Dabei hängt das Clustering-Ergebnis maßgeblich von diesen beiden Parametern ab:

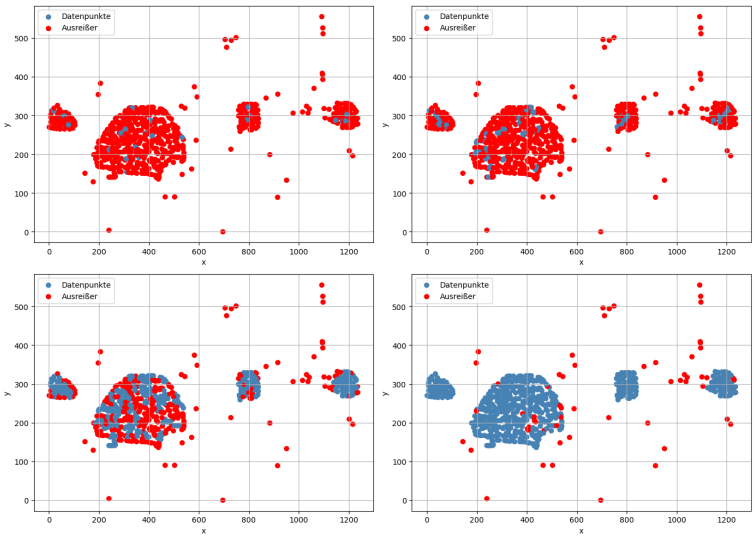


Abbildung 5: Vergleich der Leistung des DBSCAN-Algorithmus bei verschiedenen Parameterkombinationen

Die Methode zur Bestimmung von ϵ im DBSCAN-Algorithmus basiert auf der Berechnung der Distanzen zu den k nächsten Nachbarn für jeden Datenpunkt in einem Datensatz. Nach dem Sortieren dieser Distanzen wird der ϵ-Wert so gewählt, dass er dem Abstand zum weitesten der k nächsten Nachbarn für einen bestimmten Prozentsatz der Datenpunkte entspricht.[6] Somit ließ sich ein Parameter schon sehr genau bestimmen. Nun musste noch der korrekte Wert für $minPts$ gefunden werden. Zur Bewertung der verschiedenen Cluster, die durch das Variieren von $minPts$ entstanden, diente der Silhouette-Score als Evaluierungsmetrik. Zuletzt wurde sich für den Wert von $minPts$ entschieden, der den höchsten Silhouette-Score erzielt hat. Jedoch ist DBSCAN alleine noch nicht in der Lage, alle Ausreißer zu identifizieren. Aufgrund dieser Ungenauigkeit wurde noch eine weitere Ausreißererkennungsmethode inkludiert, nämlich die „Local Outlier Factor“ (kurz LOF). Die LOF identifiziert Anomalien durch den Vergleich der lokalen Dichte eines Datenpunktes mit den Dichten seiner Nachbarn, wobei ein hoher LOF-Wert auf einen Ausreißer hinweist.[7] Abschließend wurden alle gefundenen Anomalien beider Methoden zusammengefügt und das Ergebnis entsprechend ausgegeben:

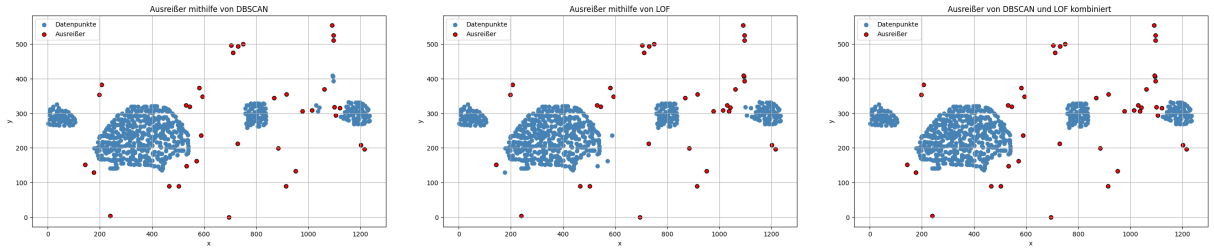


Abbildung 6: Ausreißer der einzelnen Methoden

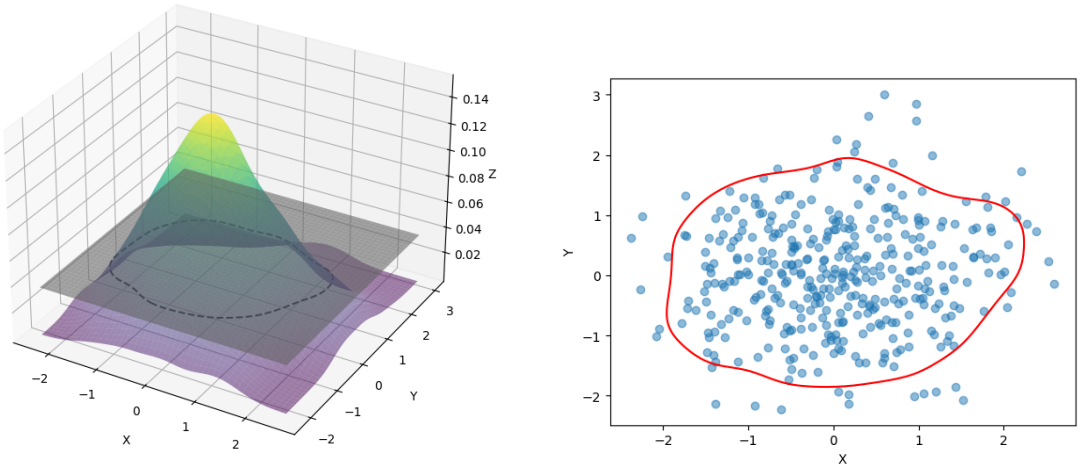
Somit haben wir die Aufbereitung der Datenpunkte (siehe Abb. 2) automatisiert gelöst.

[6] vgl. Sefidian, Amir Masoud: *How to determine epsilon and MinPts parameters of DBSCAN clustering*, `https://www.sefidian.com/2022/12/18/how-to-determine-epsilon-and-minpts-parameters-of-dbscan-clustering/` [Zugriff am 30.10.2023]

[7] vgl. Matzer, Michael: *Grundlagen Statistik & Algorithmen, Teil 7 - So deckt der Local Outlier Factor Anomalien auf*, `https://www.bigdata-insider.de/so-deckt-der-local-outlier-factor-anomalien-auf-a-803652/` [Zugriff am 30.10.2023]

4.3 Bestimmung der Form des Datensatzes

Die Bestimmung der Form des Datensatzes ist ein entscheidender Schritt bei der Selektion des am besten geeigneten Clustering-Algorithmus. Während beispielsweise K-Means sich für sphärische Clusterformen eignet, erzielt Gaussian-Mixture-Models ein besseres Ergebnis bei normalverteilten (also häufig elliptischen) Clusteranordnungen. Für die Erkennung der Form eines Datensatzes gibt es noch keinen fertigen Algorithmus, da dieses Problem aufgrund seiner Komplexität weiterhin Gegenstand aktueller Forschung ist. Dennoch wurde ein System geschaffen, welches basierend auf dem jeweiligen Eingabedatensatz approximierte Vorhersagen über die Form treffen kann. Um eine Aussage über diese tätigen zu können, mussten zunächst einmal Cluster angenähert werden. Hierbei wurde auf einen herkömmlichen Clustering-Prozess verzichtet, denn die Auswahl eines geeigneten Algorithmus wäre bereits das Ergebnis des neuronalen Netzes und damit nicht für die Feature-Bestimmung geeignet. Deshalb wurde entschieden, die Dichte als ausschlaggebendes Kriterium zu verwenden, wobei ein Gauß'scher Kerndichteschätzer verwendet wurde, um eine solche Dichtefunktion zu ermitteln.[8] Nach der Bestimmung dieser wurde für eine Höhe h, welche das Produkt der Durchschnittsdichte und einem Vorfaktor x ist, die Kontur bestimmt, die sich aus allen Punkte der Höhe h zusammensetzte. Bei mehreren Clustern bzw. Konturen werden darüber hinaus die Schnittpunkte über Dichtevergleiche einer passenden Kontur zugeordnet. Um das beste x zu wählen, wurden die Konturen erneut mittels des bereits vorgestellten Silhouette-Scores bewertet. Dabei wurden sehr rasch zufriedenstellende Resultate erzielt:



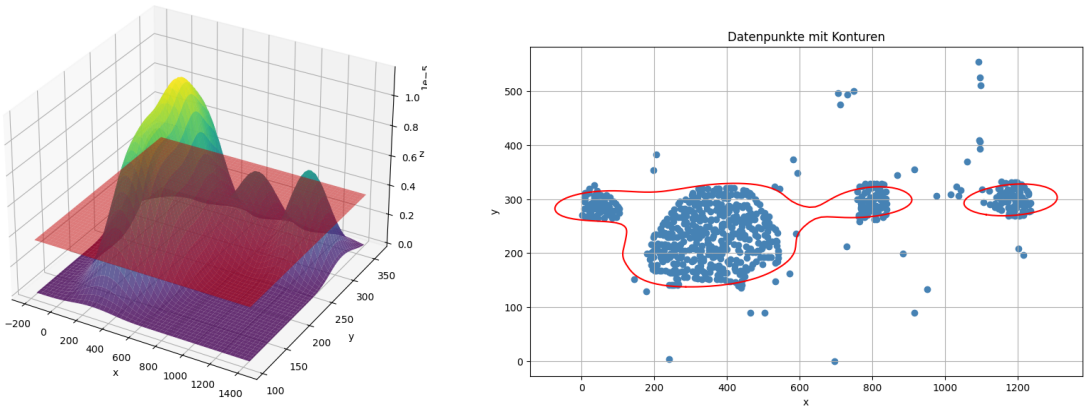
3D-Grafik der Dichteverteilung

2D-Grafik der resultierenden Kontur

Abbildung 7: Erkennung der Kontur nach Methode 1

Es fiel jedoch auf, dass in einigen Fällen die Konturen den Datensatz fehlerhaft widerspiegelten:

[8] vgl. Spodarev, Prof. Dr. Evgeny: *Dichteschätzer*, `https://www.uni-ulm.de/fileadmin/website_uni_ulm/mawi.inst.110/lehre/ss13/Stochastik_I/Skript_4.pdf` [Zugriff am 04.10.2023]



3D-Grafik der Dichteverteilung

2D-Grafik der resultierenden Kontur

Abbildung 8: Fehlerhafte Konturerkennung am Beispiel des realen Datensatzes

Deshalb wurde ein zweites Verfahren implementiert, um auch diese Fälle richtig abbilden zu können. Für dieses zweite Verfahren wurde zunächst ein zweidimensionales Histogramm erstellt[9], welches die Verteilung der gegebenen Datenpunkte verbildlicht:

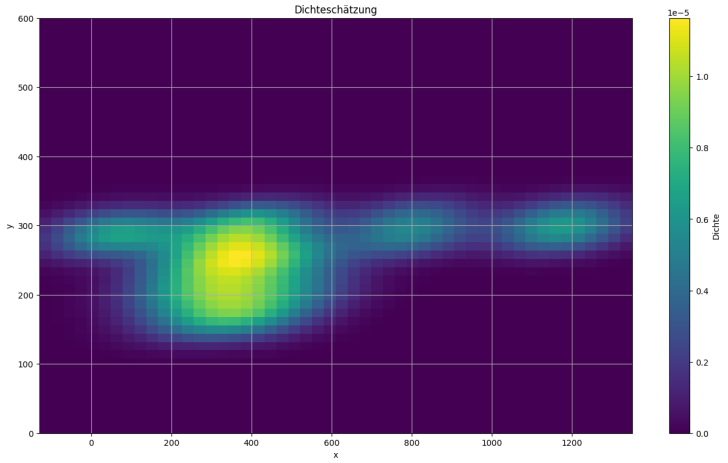


Abbildung 9: Histogramm zur Konturerkennung

Um nun Konturen zu detektieren, wurde der sogenannte Sobel-Operator verwendet. Dieser dient zur Konturerkennung, indem er hochfrequente Bereiche eines Gradienten als Kante erkennt.[10] Mittels eines Gauß-Filters wurden die entstandenen Umrisse vereinfacht, welche dann der Ausgangspunkt für die Konturen waren. Auch hier wurde sich erneut der Hyperparameteroptimierung bedient, indem der σ-Parameter für die Unschärfe durch den Silhouette-Score optimiert wurde:

[9] vgl. Saddique, Asif: *Introduction to ROOT*, `https://indico.cern.ch/event/555909/contributions/2265949/attachments/1325123/1988911/Root_Lecture2.pdf` [Zugriff am 04.10.2023]

[10] vgl. Fisher, Robert/Simon Perkins/Ashley Walker/Erik Wolfart: *Sobel Edge Detector*, `https://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm` [Zugriff am 04.10.2023]

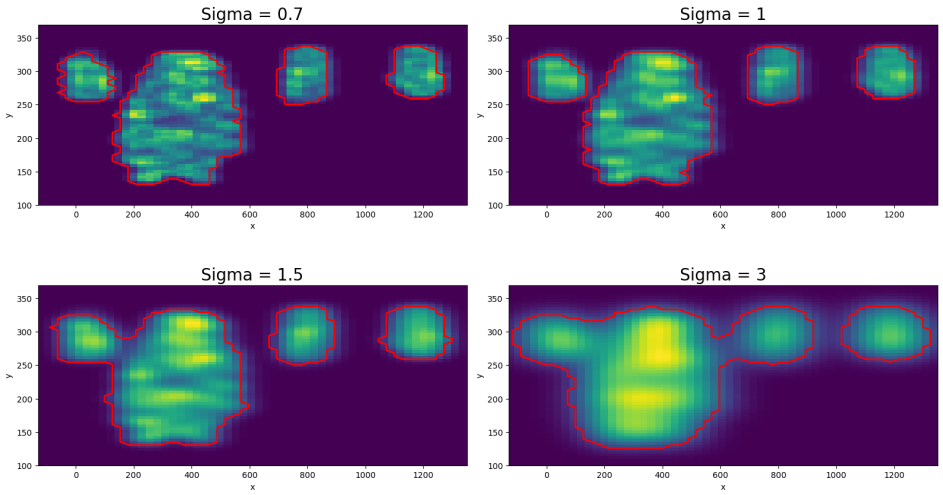


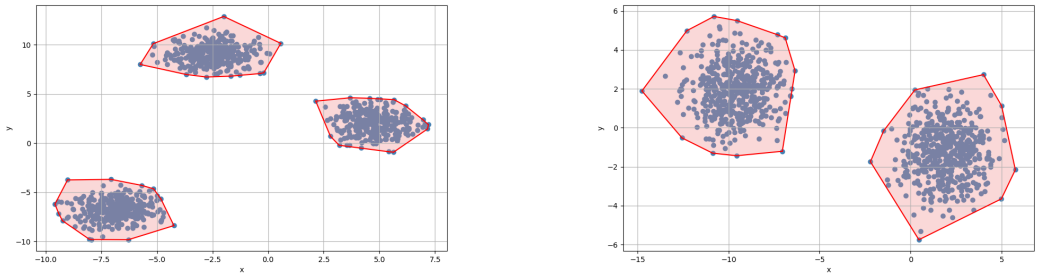
Abbildung 10: Vergleich der Auswirkung verschiedener σ-Werte auf die Konturerkennung

Es wurden zunächst die besten Konturen evaluiert, die aus beiden Verfahren hervorgingen, und anschließend mithilfe des Silhouette-Scores verglichen. Anhand dieser Ergebnisse wurde die passendere beider Konturen identifiziert, die dann verwendet wurde, um detaillierte Aussagen über die Form der vermeintlichen Cluster sowie die allgemeine Struktur des Datensatzes zu treffen. Die beste Kontur wäre somit entweder bei $\sigma = 0.7$ oder $\sigma = 1$.

4.4 Selektion weiterer Features zur exakten Charakterisierung des Datensatzes

Exzentrizität der Kontur

Die Kontur des Datensatzes als alleiniges Feature würde jedoch nicht ausreichen, um den Datensatz vollständig beschreiben zu können. Daher ist es notwendig, noch einige weitere Merkmale zu extrahieren, um eine genauere Aussage über die Beschaffenheit des Datensatzes treffen zu können. Hierbei kann die Exzentrizität der Konturpunkte einen ersten Eindruck über die Verteilung der Datenpunkte vermitteln. Ein Exzentrizitätswert nahe null deutet auf eine annähernd kreisförmige Kontur hin, während ein Wert nahe eins auf eine stark elliptische Form hinweist.



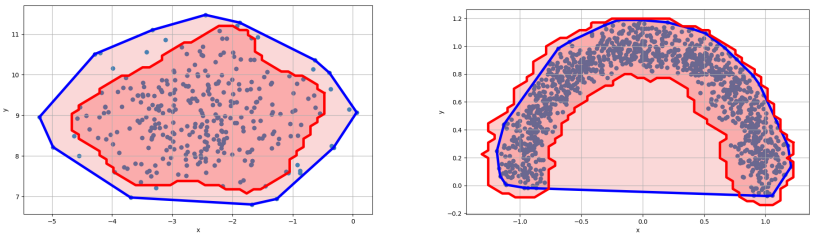
Kontur mit hoher Exzentrizität

Kontur mit niedriger Exzentrizität

Abbildung 11: Vergleich der Exzentrizitäten zweier Konturen

Verhältnis der konvexen Hülle zur Konturfläche

Ein weiteres maßgebliches Merkmal zur Beschreibung eines Datensatzes ist das Verhältnis der Fläche einer konvexen Hülle zur tatsächlichen Konturfläche. Ein niedriger Wert würde vermuten lassen, dass die Konturpunkte eng an der konvexen Hülle liegen, während ein hoher Wert auf das Vorhandensein von Einbuchtungen oder anderen Abweichungen von der konvexen Form verweist. Somit ermöglicht dieses Verhältnis einen Einblick in die allgemeine Geometrie des Datensatzes.



Niedriges Verhältnis beider Flächen Hohes Verhältnis beider Flächen

Abbildung 12: Vergleich der Verhältnisse der konvexen Hülle zur Konturfläche

Kurtosis und Schiefe der Datenpunkte

Die Kurtosis und die Schiefe sind statistische Größen, die zur Erfassung der Punkteverteilung des Datensatzes dienen.

1. Das statistische Maß der Kurtosis wird herangezogen, um zu bestimmen, in welchem Ausmaß die Verteilung der Datenpunkte spitzer oder flacher als eine Normalverteilung ausfällt. Sie ermöglicht es, Einblicke in die zentralen Bereiche und die Extrema der Datenverteilung zu erhalten, und erleichtert dadurch Rückschlüsse auf potentielle Ausreißer.

2. Die Schiefe gibt das Maß der Asymmetrie der Verteilung der Datenpunkte im Vergleich zur Normalverteilung an. Bei einer positiven Schiefe wird eine Verteilung beschrieben, bei welcher der rechte Teil der Kurve flacher verläuft als der linke Graphenabschnitt (rechtsschiefe Verteilung). Eine negative Schiefe hingegen weist auf eine linksschiefe Verteilung hin, bei der der linke Teil der Kurve flacher als der rechte Teil verläuft. [11]

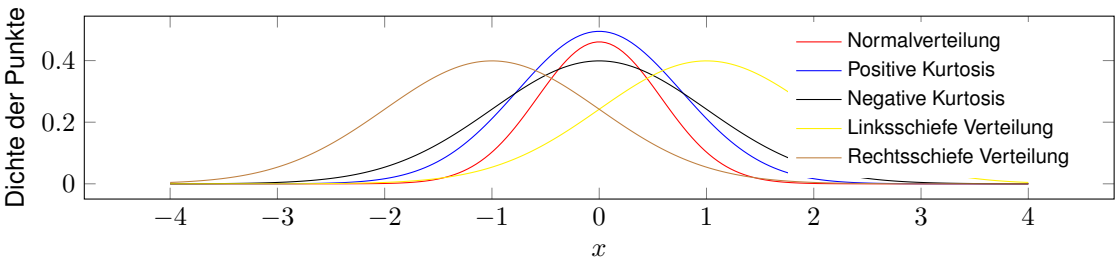


Abbildung 13: Verschiedene Punkteverteilungen im Vergleich zur Normalverteilung

[11] vgl. Gavali, Suvarna: *Skewness and Kurtosis: Quick Guide*, `https://www.analyticsvidhya.com/blog/2021/05/shape-of-data-skewness-and-kurtosis/` [letzer Zugriff: 30.10.2023]

4.5 Generation von Daten und Trainingsprozess des neuronalen Netzes

Die beschriebenen Features dienen als Grundlage für das neuronale Netz. Um dieses jedoch trainieren zu können, werden noch entsprechende Trainingsdaten benötigt. Dabei wurde eine Klasse $DataAugmentor$ erstellt, welche verwendet wird, um einzelne Datensätze leicht abzuändern. Dies ist besonders hilfreich, um die Vielfalt und Robustheit des neuronalen Netzes zu erhöhen. In dieser Klasse können zufällige Punkte entfernt und hinzugefügt, Datenpunkte skaliert und verschoben sowie die Daten um ihren Schwerpunkt gedreht werden. Nachdem alle generierten Datensätzen in einer Liste abgespeichert wurden, werden für jeden Datensatz die Features berechnet und zusammen mit dem am besten geeigneten Clustering-Algorithmus für einen spezifischen Datensatz in einer Tabelle abgespeichert. Dabei steht in der ersten Spalte der jeweils beste Clustering-Algorithmus (den wir jeweils manuell festlegten), wobei in den übrigen Spalten die extrahierten Features aufzufinden sind:

	0	1	2	3	4	5	6	7	8	9
K-Means	0.548814	0.715189	0.602763	0.544883	0.423655	0.645894	0.437587	0.891773	0.963663	0.383442
K-Means	0.791725	0.528895	0.568045	0.925597	0.071036	0.087129	0.020218	0.832620	0.778157	0.870012
K-Means	0.978618	0.799159	0.461479	0.780529	0.118274	0.639921	0.143353	0.944669	0.521848	0.414662
DBSCAN	0.264556	0.774234	0.456150	0.568434	0.018790	0.617635	0.612096	0.616934	0.943748	0.681820
DBSCAN	0.359508	0.437032	0.697631	0.060225	0.666767	0.670638	0.210383	0.128926	0.315428	0.363711
DBSCAN	0.570197	0.438602	0.988374	0.102045	0.208877	0.161310	0.653108	0.253292	0.466311	0.244426
GMM	0.158970	0.110375	0.656330	0.138183	0.196582	0.368725	0.820993	0.097101	0.837945	0.096098
GMM	0.976459	0.468651	0.976761	0.604846	0.739264	0.039188	0.282807	0.120197	0.296140	0.118728
GMM	0.317983	0.414263	0.064147	0.692472	0.566601	0.265389	0.523248	0.093941	0.575946	0.929296

Abbildung 14: Ausschnitt der in der Excel-Tabelle gespeicherten Features

5 Ergebnisse

Im Zuge unserer Arbeit haben wir es geschafft, ein System zu entwickeln, welches basierend auf dem Eingangsdatensatz den besten Clustering-Algorithmus vorhersagt. Für die Netzarchitektur wird ein sequentielles Modell verwendet, welches aus zwei Hidden-Layers mit 48 und 32 Neuronen besteht. Diese optimalen Werte wurden durch das systematische Probieren verschiedener Parameterkombinationen ermittelt.

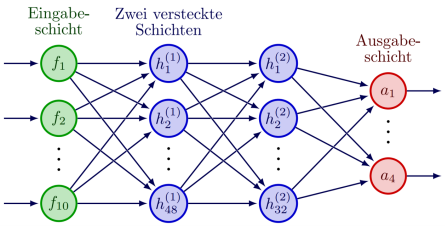


Abbildung 15: Aufbau eines neuronalen Netzes

Nach der Definition wird das Netzmodell trainiert und schließlich anhand der synthetischen Testdaten evaluiert, wobei es mit einer Wahrscheinlichkeit von $99,81\%$ den besten Algorithmus vorhersagt. Darüber hinaus testeten wir unser System auch an einigen realen Datensätzen. Hier konnte ungefähr für 70% aller Datensätze der korrekte Algorithmus bestimmt werden.

Dieser Weg zu einem neuronalen Netz, welches mit einer so hohen Prozentzahl auf synthetische Datensätze funktioniert, war jedoch schon sehr steinig und geprägt von Hindernissen und Schwierigkeiten, auf welche wir im Folgenden eingehen möchten.

Ausreißererkennung

Das Ziel der Ausreißererkennung war es, Anomalien (häufig Messfehler) aus dem Datensatz zu entfernen, um das Ergebnis den neuronalen Netzes nicht zu verfälschen. Hierbei nutzten wir die Fähigkeit des DBSCAN-Algorithmus, Ausreißer in Daten zu erkennen und kombinierten diese mit der schon bestehenden Methode $Local - Outlier - Factor$. Eine Herausforderung stellte die Ermittlung der optimalen Parameterwahl des DBSCAN-Algorithmus dar, welches wir durch systematisches Probieren mit einer anschließenden Bewertung durch den Silhouette-Score lösten.

Extraktion der Features

Hier war die erste Herausforderung, die Form des gegebenen Datensatzes beschreiben zu können. Dazu entwickelten wir zwei Methoden:

- Bei der ersten Methode wird mithilfe eines 3D-Dichteplots eine Ebene durch diesen gelegt, wobei die Schnittpunkte der Ebene mit dem 3D-Plot die Konturen darstellen. Diese horizontale Fläche wird sukzessive auf verschiedene Höhen gelegt. Die daraus resultierende Kontur wird jeweils mithilfe des Silhouette-Scores bewertet und sich zuletzt für die beste Kontur entschieden. Dennoch funktionierte diese Methode nicht für alle Datensätze, da der 3D-Dichteplot in manchen Fällen nicht wie gewünscht den Datensatz repräsentierte oder die Ebene nicht auf der korrekten Höhe angesetzt wurde. Er ermöglichte uns jedoch, einen ersten Einblick in die Formerkennung zu erhalten und mögliche Schwierigkeiten direkt zu erkennen.
- Um ein funktionierendes System zur Formerkennung zu erhalten, entwickelten wir eine weitere Methode. Diese nutzt einen Gauß-Filter, um das Bild des Datensatzes unscharf zu machen. Der Grad der Unschärfe wird durch den σ-Parameter variiert. Nun werden mithilfe des Sobel-Operators Konturen für spezifische σ-Werte erkannt und diese durch den Silhouette-Score bewertet. Abschließend werden die Konturen nach Methode 1 und jene nach Methode 2 wiederum verglichen und sich für die bessere der beiden entschieden. Nach einigen Tests haben wir herausgefunden, dass sich beide Techniken zur Formerkennung ausgezeichnet gegenseitig ergänzen.

Darüber hinaus implementierten wir neben den in Kapitel 4.4 schon vorgestellten Features insgesamt zehn Weitere.

Generierung der Daten und Trainingsprozess des neuronalen Netzes

Zuletzt erstellten wir 5184 verschiedene synthetische Datensätze (dazu gehören u.a. folgende Datensätze aus dem Modul `sklearn`: $make\_blobs$, $make\_moons$, $make\_gaussian\_quantiles$, $make\_circles$, siehe Abb. 3), veränderten sie leicht mithilfe der Klasse $DataAugmentor$, berechneten die Features und speicherten sie in einer Tabelle. Diese dienten als Trainingsdaten für das neuronale Netz.

6 Ergebnisdiskussion

Unsere Forschung begann mit der Bestimmung von potenziell relevanten Features, die charakteristische Merkmale eines Datensatzes abbilden können. Eine wichtige Erkenntnis war dabei, dass die Features nur zielführend sind, wenn sie eine Aussage über die Form eines Datensatzes treffen. Wir schafften es, zwei eigene Methoden zur Formbestimmung zu entwickeln, welche zuverlässige Ergebnisse liefern, jedoch selbst keine Clustering Prozesse darstellen. Diese Überlegungen legten zusammen mit unserer gut funktionierenden Ausreißererkennung den Grundstein für den Erfolg unseres Vorhabens.

Anhand der insgesamt zehn ausgewählten Features konnte das neuronale Netz mit mehr als 5000 synthetisch generierten Datensätzen trainiert werden. Beim Training sahen wir uns mit zwei wesentlichen Herausforderungen, dem Over- bzw. Underfitting, konfrontiert. Dabei stellt das Overfitting eine Entwicklung des Netzes dar, welche sich zu sehr an den Trainingsdaten orientiert, und somit Vorhersage für unbekannte Datensätze nur wenig erfolgreich sind, wobei dafür oft zu einseitige und zu wenig generalisierte Trainingsdaten verantwortlich sind. Im Gegensatz dazu ist das Netz beim Underfitting nicht in der Lage, die Beziehung zwischen den Features und den Ergebnissen ausreichend darzustellen, wobei dafür hauptsächlich zu einfache und generalisierte Trainingsdaten verantwortlich sind. Beim Training unseres Netzen griffen wir auf verschiedene Möglichkeiten zur Augmentierung der Trainingsdaten zurück und konnten so eine Vielzahl an Trainingsdaten erzeugen, welche es nicht nur möglich machten, die komplexen Zusammenhänge der einzelnen Features zu modellieren, sondern auch durch die verschiedensten Erscheinungen Robustheit in das Netz zu bringen, und somit generelle Prognosen ermöglichten, ohne sich zu sehr an den synthetischen Trainingsdaten zu orientieren. Das finale Netz konnte eine zufriedenstellend hohe Genauigkeit aufweisen.

Damit konnten wir beweisen, dass es möglich ist, für einen gegebenen Datensatz den bestgeeigneten Clustering-Algorithmus zuverlässig automatisiert auszuwählen.

Zudem hat die Anwendung unseres Verfahrens auf reale Datensätze gezeigt, dass es in der Lage ist, auch in komplexen Szenarien, in Anbetracht der lediglich zehn verwendeten Features, hohe Genauigkeitsraten zu erreichen. Für bessere Ergebnisse wäre es notwendig, das System aktiv auf reale Daten zu trainieren und entsprechend auch die Features anzupassen. Dennoch ist seine Praxistauglichkeit bereits erkennbar.

Eine der bemerkenswertesten Eigenschaften unseres Systems zur Auswahl von Clustering-Algorithmen ist seine Anpassungsfähigkeit. In einer sich ständig verändernden Datenlandschaft ist Flexibilität von entscheidender Bedeutung. Unser System kann sich an neue Datentypen, Strukturen und Anforderungen anpassen, da es auf einer datengetriebenen Grundlage arbeitet. Dies bedeutet, dass es nicht nur für aktuelle Datensätze effektiv ist, sondern auch für zukünftige Datenherausforderungen gewappnet ist.

Ein weiterer wichtiger Aspekt unseres Projekts ist die Transparenz und Nachvollziehbarkeit. Data Scientists und Analysten können oft mit komplexen und undurchsichtigen Modellen konfrontiert werden. Unser System basiert auf einem neuronalen Netz, das auf klaren und verständlichen Daten trainiert wurde. Dies ermöglicht es den Anwendern, die Entscheidungsfindung des Systems nachzuvollziehen und zu verstehen, warum ein bestimmter Clustering-Algorithmus ausgewählt wurde. Diese Transparenz kann das Vertrauen in das System stärken und die Akzeptanz in der Branche fördern.

Unsere Forschung und Entwicklung eines automatisierten Systems zur Auswahl von Clustering-Algorithmen hat nicht nur zur Schaffung eines effizienten Werkzeugs geführt, sondern auch neue Perspektiven für die Datenanalyse eröffnet. Die Automatisierung dieses entscheidenden Schritts kann nicht nur die Arbeit von Data Scientists erleichtern, sondern auch zu einer erheblichen Verbesserung der Qualität der Datenanalysen führen.

Unsere Arbeit könnte daher als Grundlage für zukünftige Entwicklungen dienen, indem weitere Algorithmen und spezifische Features integriert werden, um die Genauigkeit, die Anwendungsbereiche und die Effizienz weiter zu verbessern. In einer sich schnell entwickelnden digitalen Welt können solche Fortschritte einen erheblichen Beitrag zur Lösung komplexerer Probleme leisten und Innovationen in verschiedensten Bereichen fördern.

7 Fazit und Ausblick

Das Hauptziel dieser Projektarbeit war die Entwicklung eines Programms, das automatisch den passenden Algorithmus für das Clustering eines beliebigen Datensatzes auswählt. Unsere Lösung für dieses Problem funktioniert zuverlässig und stellt daher einen Machbarkeitsbeweis dar. Zudem öffnen sich gleichzeitig eine große Bandbreite an Erweiterungsmöglichkeiten.

Bisher wurde das Verfahren hauptsächlich auf künstliche und nur einige wenige reale Datensätze angewendet. Um die Funktionalität und Brauchbarkeit für die Realität zu zeigen, könnte das neuronale Netz gezielt mit realitätsbezogenen Datensätzen verschiedenster Art trainiert werden. Im Zuge dessen könnte auch die Auswahl der Features erweitert werden.

Ein wesentliches Erweiterungsfeld bietet zudem die Integration weiterer Clustering-Algorithmen. Derzeit ist das neuronale Netz in der Lage, aus vier verschiedenen Algorithmen den optimalen auszuwählen, was eine breite Abdeckung vieler Datensätze ermöglicht. Dennoch existieren spezialisierte Verfahren, die für bestimmte Datentypen effizienter sind, weshalb eine Erweiterung die Rechenprozesse erheblich beschleunigen könnte.

Zum Abschluss lässt sich sagen, dass auf eine sehr erfolgreiche Arbeitsphase zurückgeblickt werden kann. Um das Ziel zu erreichen, mussten einige Hürden überwunden werden. Dennoch hat die Herausforderung immer viel Freude bereitet und dabei geholfen, Vieles zu lernen.

8 Literaturverzeichnis

Gedruckte Literatur

Raschka, Sebastian; Mirjalili, Vahid: *Python Machine Learning - Machine Learning and Deep Learning with Python, scikit-learn, and Tensorflow 2*, 3. Aufl., Birmingham (Vereinigtes Königreich), Packt, 2019

Internetliteratur

Chauhan, Nagesh Singh: *DBSCAN Clustering Algorithm in Machine Learning*, https://www.kdnuggets.com/2020/04/dbscan-clustering-algorithm-machine-learning.html [Zugriff am 30.10.2023]

Dadi, Harihara; Venkatesh, P.; Poornesh, P.; L., Narayana Rao; Kumar, N.: *Tracking Multiple Moving Objects Using Gaussian Mixture Model*, https://www.researchgate.net/publication/305709395_Tracking_Multiple_Moving_Objects_Using_Gaussian_Mixture_Model[Zugriff am 30.10.2023]

DataNovia (Hrsg.): *Cluster Validation Statistics: Must Know Methods*, https://www.datanovia.com/en/lessons/cluster-validation-statistics-must-know-methods [Zugriff am 30.10.2023]

Fisher, Robert; Simon Perkins; Ashley Walker; Erik Wolfart: *Sobel Edge Detector*, https://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm [Zugriff am 04.10.2023]

Gaido, Marco: *Distributed Silhouette Algorithm: Evaluating Clustering on Big Data*, https://arxiv.org/pdf/2303.14102.pdf [Zugriff am 30.12.2022]

Gavali, Suvarna: *Skewness and Kurtosis: Quick Guide*, https://www.analyticsvidhya.com/blog/2021/05/shape-of-data-skewness-and-kurtosis/ [Zugriff am: 30.10.2023]

Matzer, Michael: *Grundlagen Statistik & Algorithmen, Teil 7 - So deckt der Local Outlier Factor Anomalien auf*, https://www.bigdata-insider.de/so-deckt-der-local-outlier-factor-anomalien-auf-a-803652/ [Zugriff am 30.10.2023]

National Institute of Standards and Technology (Hrsg.): *Measures of Skewness and Kurtosis*, https://www.itl.nist.gov/div898/handbook/eda/section3/eda35b.htm [Zugriff am 06.10.2023]

Roux, Maurice: *A Comparative Study of Divisive and Agglomerative Hierarchical Clustering Algorithms*, https://hal.science/hal-02085844/document [Zugriff am 30.10.2023]

Saddique, Asif: *Introduction to ROOT*, https://indico.cern.ch/event/555909/contributions/2265949/attachments/1325123/1988911/Root_Lecture2.pdf [Zugriff am 04.10.2023]

Sefidian, Amir Masoud: *How to determine epsilon and MinPts parameters of DBSCAN clustering*, https://www.sefidian.com/2022/12/18/how-to-determine-epsilon-and-minpts-parameters-of-dbscan-clustering/ [Zugriff am 30.10.2023]

Spodarev, Prof. Dr. Evgeny: *Dichteschätzer*, https://www.uni-ulm.de/fileadmin/website_uni_ulm/mawi.inst.110/lehre/ss13/Stochastik_I/Skript_4.pdf [Zugriff am 04.10.2023]

Weisstein, Eric W.: *Skewness*, https://mathworld.wolfram.com/Skewness.html [Zugriff am 06.10.2023]

9 Abbildungsverzeichnis

Alle Abbildungen wurden von den Autoren der Arbeit selbst mithilfe eigener Programme erstellt. Keine von den verwendeten Grafiken wurde aus dem Internet oder sonstiger Literatur entnommen.

10 Unterstützungsleistungen

- Dr. Wolfgang Felber, Head of Department for Satellite Based Positioning, Fraunhofer Institut IIS, Nürnberg, hat uns bei der Erstellung unserer Arbeit unterstützt. Er hat uns durch eine grundlegende Themenidee (Themengebiet: „automatisiertes Clustering“) mit Herrn Christopher Sobel bekannt gemacht, woraus unsere finale Zielsetzung hervorging. Weiterhin konnten wir in regelmäßigen Abständen unseren Arbeitsfortschritt vorstellen, erstes Feedback erhalten, sowie den Aufbau unserer schriftlichen Arbeit und anstehende Präsentationen (Themenverteidigung, Zwischenstandsverteidigung) vorstellen und Tipps und Verbesserungsvorschläge erhalten.
- Christopher Sobel, Machine Learning & Validation, Fraunhofer Institut IIS, Nürnberg, hat uns bei der Erstellung unserer Arbeit unterstützt. Er hat uns durch einen Workshop am Fraunhofer IIS einen Einstieg in das automatisierte Clustering gegeben, aus welchem unsere Zielstellung eines Systems zur automatisierten Auswahl geeigneter Clustering-Algorithmen hervorging. Weiterhin konnten wir Herrn Sobel unseren Arbeitsfortschritt in regelmäßigen Abständen vorstellen und erstes Feedback erhalten.
- Udo Weitz, ehemaliger Schulleiter des Spezialschulteils des Albert-Schweitzer-Gymnasiums in Erfurt, vermittelte uns den Kontakt zu Dr. Wolfgang Felber vom IIS. Dadurch kamen wir erstmals mit dem Thema des Clusterings in Berührung.