Kapitel 4

Anomaliedetektion

Anomaliedetektion beschreibt die Aufgabe, Trends, Muster und Punkte in einem Datensatz zu finden, die nicht dem Normalzustand entsprechen [8]. Anders gesagt lautet das Ziel: die Punkte finden, die sich von den anderen Punkten im Datensatz stark unterscheiden [27, Kap. 10]. Diese andersartigen Datenpunkte oder -sequenzen werden in der Regel als Anomalie, Ausreißer oder Ausnahmen bezeichnet, wobei Anomalie der geläufigste Begriff ist. Anomaliedetektion findet große Verwendung in verschiedenen Anwendungsbereichen, wie z. B. in der Netzwerktechnik zur Erkennung von potenziellen Angriffen durch Eindringlinge in ein Netzwerk anhand von ungewöhnlichem Traffic [3]. Auch in der Medizin können nach einem Elektrokardiogramm (EKG) durch Anomaliedetektion Herzrhythmusstörungen erkannt werden [10], genau wie eine Bank ein Interesse an Anomalien im Kreditkartenverhalten ihrer Kunden hat, um Betrugsfälle zu erkennen [16, 7].

Die simpelste Herangehensweise zur Erkennung von Anomalien ist die, dass zuerst definiert wird, welche Punkte im Datensatz normalem Verhalten entsprechen und alle davon abweichenden Punkte als Anomalie zu kennzeichnen. Doch so einfach die Herangehensweise wirkt, so anfällig ist sie auch für Fehler. Dabei heben sich einige Herausforderungen hervor.

Zum Einen die Frage, wo genau die Grenze zwischen normalem und anomalem Verhalten liegen soll. Eine Region zu definieren, die jeden möglichen normalen Punkt beinhaltet und jedmöglichen anomalen Punkt ausschließt, ist nicht trivial und oft nicht präzise durchführbar. So ist es durchaus möglich, dass in manchen Fällen anomale Punkte als normal bezeichnet werden, und normale Punkte als anomal, je nachdem, wo die Grenze liegt.

Es stellt sich ebenfalls die Frage, ob eine Anomalie einer binären Natur unterliegt: Entweder es handelt sich um eine Anomalie oder einen Normalzustand. Doch die Wahrheit liegt oft in der Mitte. Weicht ein Punkt oder eine Sequenz bereits nur leicht vom Normal ab, so kann es bereits erste Hinweise auf mögliches zukünftiges anomales Verhalten in einer Zeitserie geben, bevor sich solche Datenpunkte als

Anomalie zeigen. Deshalb ist es hilfreich, charakterisieren zu können, wie weit der Punkt oder die Sequenz vom Normal abweicht. Diese Charakterisierung kann dabei als *Anomaly Score* bezeichnet werden und beispielsweise eine Dezimalzahl zwischen 0 und 1 sein.

Normalzustände sind in Zeitserien oft zeitvariant und daher schwer festzuhalten bei einer kontinuierlichen Datenaufzeichnung. Zudem sind Normalzustände und Abweichungen davon in unterschiedlichen Bereichen auch unterschiedlich signifikant. Während beim menschlichen Körper eine geringe Abweichung der Körpertemperatur bereits gravierend sein kann, ist die gleiche relative Abweichung in einer anderen Domäne wie in einem Aktienkurs weniger drastisch und unterliegt dementsprechend auch einem Anpassungsbedarf, bevor es an die Erkennung möglicher Anomalien geht.

Daraus lässt sich direkt zum nächsten Problem übergehen. Die Unterscheidung zwischen globalen und lokalen Anomalien [6]. Hier ist der Kontext wichtig: Eine Person mit einer Körpergröße von mehr als 2 m ist in ihrer Nachbarschaft sicherlich eine Anomalie, während sie in einem Basketballteam kaum herausragt - im wahrsten Sinne des Wortes. Diese Art der Anomalie wird auch als kontextuelle Anomalie bezeichnet [30, S. 12].

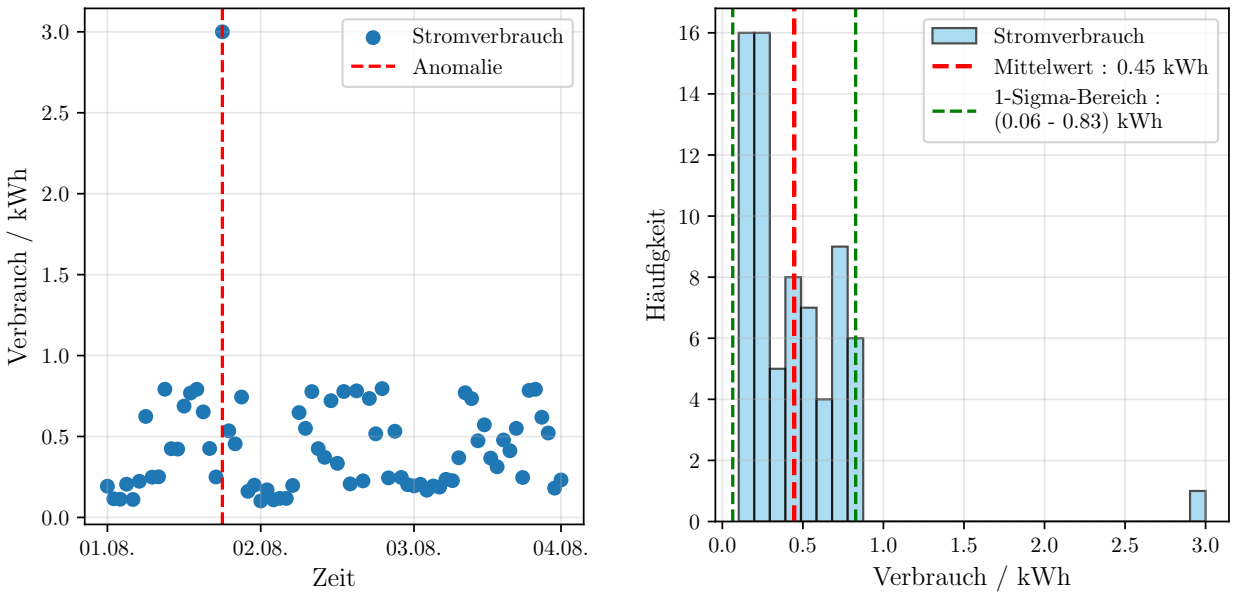
4.1 Anomaliearten

Doch bevor eine Auswahl an geeigneten Verfahren oder Algorithmen zur Anomaliedetektion getroffen wird, muss zuerst verstanden werden, welche verschiedenen Arten von Anomalien es gibt und wie sich diese voneinander unterscheiden. Auch wenn Studien zeigen, dass es durchaus Algorithmen gibt, die über mehrere verschiedene Kategorien gut abschneiden [30, S. 30 - 31] [24], so soll zunächst für jede Kategorie mindestens ein passender Kandidat gefunden werden. Diese werden dann in einem nächsten Schritt kreuzweise getestet, um auch solche Allrounder entdecken zu können. Dabei ist auch immer der Kontext der Anwendung wichtig. Wie eingangs erwähnt, sind für verschiedene Tätigkeitsfelder verschiedene Anforderungen an die Präzision oder Genauigkeit gestellt, weshalb immer die spezifischen Anforderung bedacht werden müssen, und nicht jeder Algorithmus gleich performant ist über mehrere Datensätze hinweg.

Für die Kategorien wird sich zunächst auf wenige, für diese Arbeit relevante, beschränkt: **Punktanomalien**, **Subsequenzanomalien** und **Korrelationsanomalien**, abgeleitet von Chandola et al. [8].

4.1.1 Punktanomalien

Ein einzelner Datenpunkt, der stark von den anderen Punkten im Datensatz abweicht, heißt Punktanomalie [8]. Genauer gesagt, wenn ein Datenpunkt weit außerhalb der Wahrscheinlichkeitsverteilung



(a) Stündliche Smart Meter Messdaten

(b) Histogramm des gemessenen Stromverbrauchs

Abbildung 4.1: Beispielszenario einer Punktanomalie: Stromverbrauch eines Haushaltes über den Zeitraum von drei Tagen. Anhand des Histogramms wird die Anomalie verdeutlicht.

des Datensatzes liegt, ist er anomal [27, Kap. 10]. Punktanomalien können recht leicht erkannt werden, da Punktanomalien stark vom Mittelwert und vom Median des Datensatzes abweichen. Wenn von Ausreißern gesprochen wird, sind damit typischerweise Punktanomalien gemeint.

Als Beispielszenario dient ein Smart Meter, das den stündlichen Stromverbrauch misst. In Abb. 4.1a ist der gemessene Stromverbrauch dargestellt mit einer klar erkennbaren Punktanomalie am 01.08. um 18 Uhr. Die Anomalie wird mit bloßem Auge deutlich und kann auch mit statistischen Größen nachgewiesen werden, wie in Abb. 4.1b anhand der Häufigkeitsverteilung und dem Mittelwert sowie dem Median zu sehen ist. Das Histogramm dient als gute Approximation für die Wahrscheinlichkeitsverteilung der Messwerte, und zeigt entsprechend die Eindeutigkeit des Ausreißers.

Um nun eine Aussage treffen zu können, ist es wichtig den Kontext der vorliegenden Daten zu kennen. Wenn Daten für ein weitaus größeres Zeitfenster vorliegen, z. B. für eine Woche oder einen Monat, könnte sich möglicherweise zeigen, dass der hohe Verbrauch öfter und regelmäßiger vorkommt als im gezeigten Zeitraum von drei Tagen. Ob eine globale oder lediglich eine lokale Anomalie vorliegt, wird mit einem größeren Datensatz besser erkennbar. Die Anomalie könnte beispielsweise auf das gelegentliche Betreiben einer Sauna im Haus zurückführbar sein, dann würde es sich lediglich um eine lokale Anomalie handeln und in einem größeren Zeitraum in bestimmten Abständen öfter vorkommen, und wäre somit keine globale Anomalie [27, Kap. 10].

Punktanomalien sind im Kontext dieser Arbeit tendenziell weniger relevant, sollen aber aufgrund

ihrer grundsätzlichen Bedeutung bzgl. Anomaliedetektion als einfachste Kategorie trotzdem beleuchtet werden, um entsprechende Algorithmen, die der Erkennung solcher Punktanomalien zuzuordnen sind, auch gegenüber anderen Anomalien zu testen.

4.1.2 Subsequenzanomalien

Eine Zeitserie wird gem. Gl. 3.1 bereits als eine Menge definiert. Demnach wird eine Subsequenz $S_{i,j} = \{ S_i, \ldots, S_j \} \subseteq S$ von der Zeitserie S umfasst, mit der Länge oder Mächtigkeit $| S_{i,j} | = j - i + 1$ und $| S_{i,j} | \geq 1$ [24] und stellt somit einen Ausschnitt der urpsrünglichen Zeitserie dar. Subsequenzanomalien sind Muster in Zeitreihen, die von anderen Mustern innerhalb der gleichen Zeitreihe abweichen [8][30, S. 12]. Im Gegensatz zu Punktanomalien beziehen sich Subsequenzanomalien auf mehrere konsekutive Datenpunkte, die ein ungewöhnliches Muster bilden. Eine anomale Subsequenz kann also bedeuten, dass die Datenpunkte innerhalb der Subsequenz Werte in einem normalen, zu erwartenden Bereich annehmen, aber der zu Grunde liegende Trend ungewöhnlich ist [8][5, S. 17]. Solche ungewöhnlichen oder einzigartigen Trends und Entwicklungen können auf zukünftig auftretende Probleme hindeuten, die sonst unentdeckt bleiben würden.

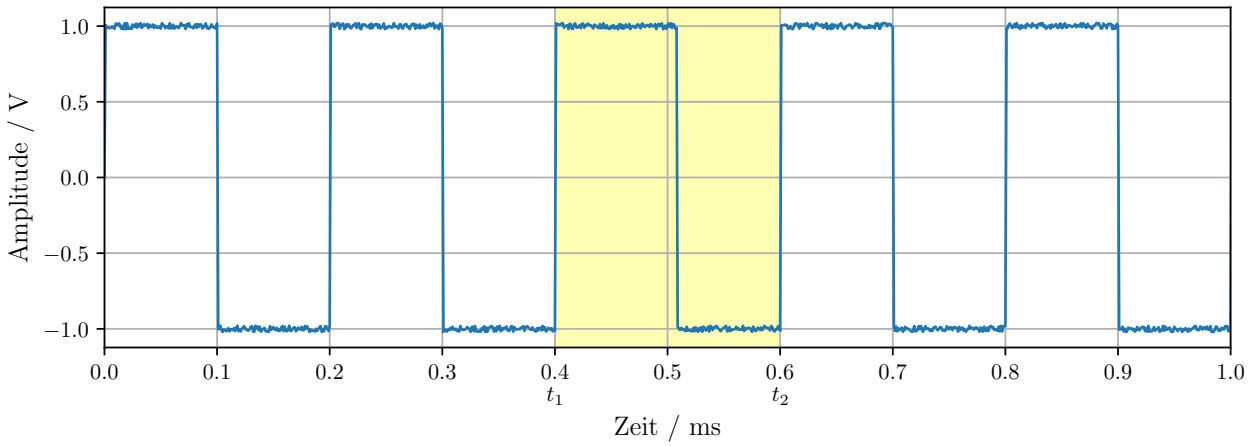


Abbildung 4.2: Einfaches Beispiel einer Subsequenzanomalie: Rechteckspannung, die zwischen -1 und +1 V oszilliert mit einer Frequenz von 5 kHz. Auffällig ist die Periode zwischen $t_1 = 0.4\,\text{ms}$ und $t_2 = 0.6\,\text{ms}$, bei der eine verspätete abfallende Flanke zu beobachten ist.

Das Beispiel in Abb. 4.2 zeigt eine sichtbare Subsequenzanomalie, die verspätete abfallende Flanke einer gemessenen Rechteckspannung. Das Muster zwischen t_1 und t_2 ist also merklich anders verglichen zu den restlichen 0,2 ms langen Perioden und daher eine Anomalie.

Bei der Analyse von EKG Daten spielen Subsequenzanomalien eine wichtige Rolle und können wertvolle Rückschlüsse auf die Herzgesundheit liefern [10]. Abb. 4.3 zeigt EKG-Daten eines Patienten mit

monomorpher ventrikulärer Tachykardie. Diese kann zu Kammerflimmern übergehen, welches unbehandelt sogar zu einem Herzstillstand führen kann [28][11, S. 131 ff.].

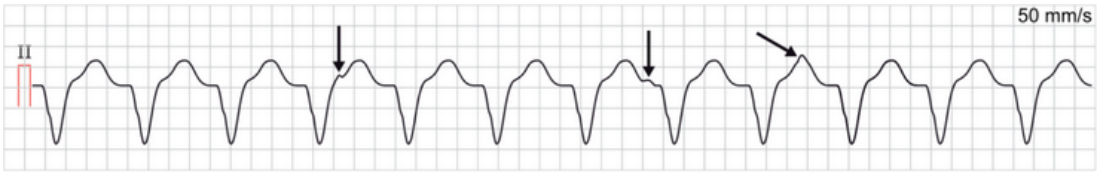


Abbildung 4.3: EKG Kanal mit Diagnose: Ventrikuläre Tachykardie [28]

Sichtbar sind die einzelnen Unregelmäßigkeiten im EKG Verlauf. Die Pfeile kennzeichnen die sog. P-Wellen, die Informationen darüber liefern, dass Vorhöfe und Herzkammern nicht synchron schlagen [28][11, S. 31 f.]. Durch die Irregularitäten lässt sich also erkennen, dass für den untersuchten Patienten eine Behandlung notwendig ist und betont die Wichtigkeit, diese Anomalien zu erkennen, um wesentlich Schlimmeres zu verhindern.

Darin liegt auch eine der Herausforderungen der Subsequenzanomaliedetektion: Ab wann ist ein Trend, der so noch nicht aufgetreten ist, Grund genug, um Maßnahmen zu ergreifen? Es bedarf also menschlicher Expertise zur Einordnung und Interpretation von Anomalien, eben wie bei EKG Daten.

4.1.3 Korrelationsanomalien

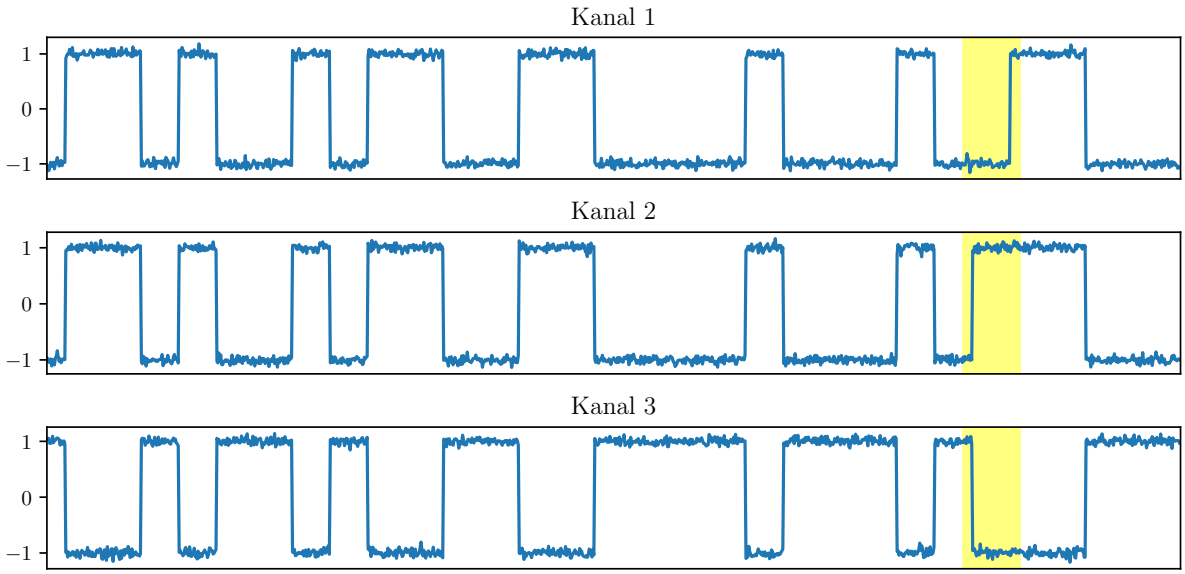


Abbildung 4.4: Korrelationsanomalie zwischen Kanal 1 und den Kanälen 2 und 3 im gelb markierten Bereich. Quelle: Datensatz *CoMuT* [20]

Während Punkt- und Subsequenzanomalien sowohl für univariate als auch multivariate Datensätze und Zeitserien auftreten können, sind Korrelationsanomalien nur möglich bei zwei oder mehr Dimensionen einer Zeitreihe und betrachten die Interaktionen zwischen verschiedenen Kanälen. Von einer Korrelationsanomalie spricht man bei Abweichungen dieser Beziehung zwischen zwei oder mehreren Kanälen [30, S.12-13] [31].

Im vorliegenden Beispiel in Abb. 4.4 ist ein Auszug aus dem Datensatz *CoMuT* - **Co**rrelated **Mu**ltivariate **T**ime Series [20] dargestellt. Die Zeitreihe besteht aus drei Kanälen, die zu zufälligen Zeitpunkten sprungartig ihren Wert zwischen -1 und 1 wechseln und jeweils leicht verrauscht sind. Kanal 1 und 2 sind stark korreliert, während Kanal 3 stark antikorreliert zu den beiden ersten Kanälen ist. Diese Korrelation wird im markierten Bereich verletzt, da Kanal 1 zeitlich versetzt zu den anderen beiden Kanälen springt – somit liegt eine Korrelationsanomalie vor.

4.2 Algorithmen zur Anomaliedetektion

In der Literatur gibt es eine breite Spanne an erprobten, möglichen Algorithmen zur Detektion der gesuchten Anomalietypen [24] [4] [18] [31]. Da der zeitliche Rahmen dieser Arbeit begrenzt ist und der vorrangige Fokus die Anwendung der Anomaliedetektionsthematik ist, müssen bei der Auswahl der Algorithmen einige Einschränkungen vorab festgelegt werden. Dazu gehört, dass ein Algorithmus bereits implementiert wurde und optimalerweise als Open Source Python Bibliothek o.ä. zur Verfügung steht oder die Idee durch bereits vorhandene Komponenten einfach selbst implementiert werden kann. Desweiteren soll es sich zu Beginn um Algorithmen handeln, die ohne vorheriges Training auskommen, also sog. Unsupervised Learning Algorithmen. Dazu eignen sich die folgenden Techniken bzw. Algorithmen am Besten.

4.2.1 Histogram-Based Outlier Score

Der erste Algorithmus zur Detektion von Punktanomalien ist der zur Ermittlung des **Histogram-Based Outlier Score** (HBOS), der sich sowohl für univariate als auch multivariate Zeitserien eignet. Dabei wird für jede Dimension eine Analyse der Häufigkeitsverteilung per Histogramm durchgeführt und anhand dessen der namensgebende Outlier Score berechnet. Die Häufigkeit aller Werte innerhalb eines Bins wird als Dichte aufsummiert, und die Dichte ergibt gem. Gl. 4.1 invers logarithmiert den Outlier Score. So erhalten Bins mit geringen Häufigkeiten, also die Bins mit selten auftretenden Werten - wahrscheinliche Anomalien - einen hohen Outlier Score und können so als Anomalie detektiert werden [13].

$$\mathrm{HBOS}\,(p) \;=\; \sum_{i=0}^{d} \log\left(\frac{1}{\mathrm{hist}_i(p)}\right) \tag{4.1}$$

Die Häufigkeitsanalyse über den kompletten Datensatz bzw. eine sehr große Datenmenge führt dazu, dass lokale Anomalien nicht erkannt werden, da diese Werte trotzdem innerhalb der erwarteten Verteilung liegen und im globalen Kontext nicht herausstehen, wie Abb. 4.5 zeigt. Einfache Abhilfe liefert die in Abb. 4.6 dargestellte Erweiterung um ein gleitendes Fenster variabler Größe. Der HBOS wird dann innerhalb eines jeden Fensters ermittelt, wodurch der zeitliche Kontext mit einbezogen wird und auch lokale Anomalien erkannt werden. Multivariate Systeme können ebenfalls einfach analysiert werden, indem für jede Variable oder Dimension separate Histogramme erstellt werden.

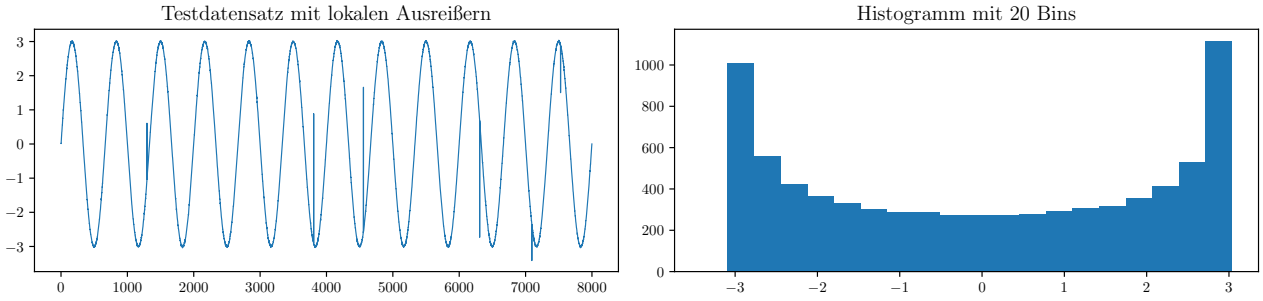


Abbildung 4.5: Problematik der Detektion lokaler Punktanomalien: Da lokale Anomalien innerhalb des erwarteten Wertebereichs und damit der erwarteten Verteilung liegen, werde diese vom Algorithmus nicht als Ausreißer erkannt.

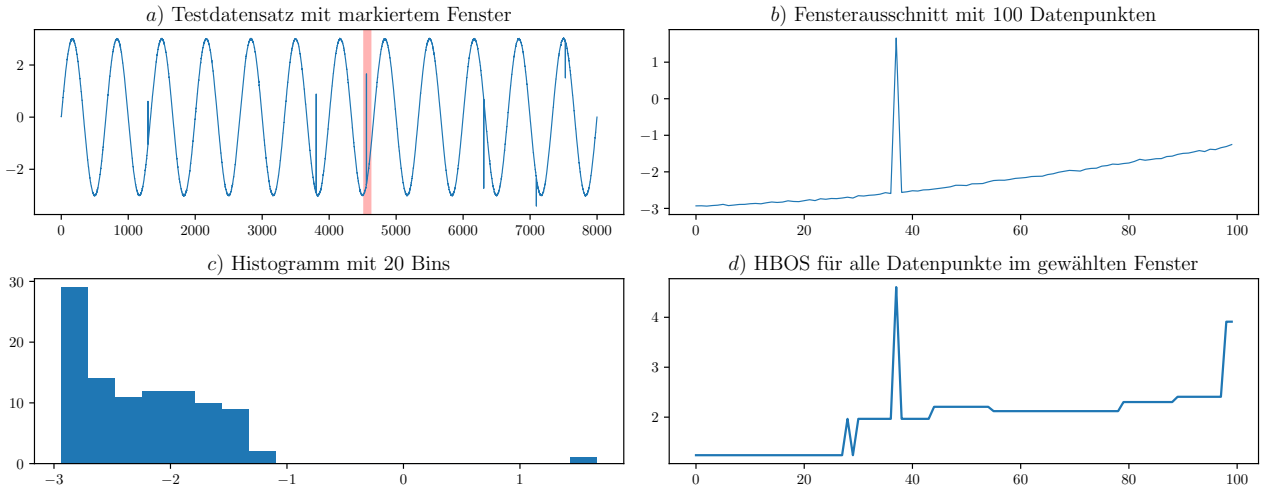


Abbildung 4.6: Lösung der Detektionsproblematik für lokale Anomalien durch ein gleitendes Fenster. Beispielhaft dargestellt an einer Fenstergröße von $n = 100$.

4.2.2 Sliding Window Z-Score

Durch die Berechnung eines gleitenden Mittelwerts sowie der entsprechenden gleitenden Standardabweichung kann der Z-Score eines Datenpunkts gem. Gl. 4.2 bestimmt werden. Der Z-Score wird in einer ähnlichen Form bereits im Jahre 1969 in *Grubb's Test* erstmals erwähnt [14]. Dabei bezeichnet x_i den zu untersuchenden Datenpunkt, μ_W den Mittelwert und σ_W die Standardabweichung des Fensters [8, S. 15:31].

$$Z_i = \frac{|\, x_i - \mu_W \,|}{\sigma_W} \tag{4.2}$$

Um für die Zeitserie den Zeitbezug beizubehalten, werden die statistischen Größen mit einem gleitenden Fenster berechnet, um auch lokale Ausreißer, deren absolute Werte innerhalb der globalen Verteilung liegen, detektieren zu können. Dabei ist die Fenstergröße zunächst fest. Entsprechend des Z-Scores deutet ein hoher Wert auf eine Anomalie hin, aufgrund der großen Abweichung zum gleitenden Mittelwert trotz Skalierung durch die Standardabweichung. Damit ist der Z-Score eine dimensionslose Größe, die direkt als Anomaly Score interpretiert werden kann.

4.2.3 GrammarViz 2.0

Der nächste Algorithmus namens **GrammarViz** wird zur Detektion von Subsequenzanomalien eingesetzt [26]. Die Funktionsweise basiert auf der Diskretisierung einer Zeitreihe in Symbole. Für Subsequenzen unterschiedlicher Größe wird dann versucht, diese mit Grammatikregeln zu beschreiben. Kann sich eine Subsequenz nicht entsprechend der am häufigsten auftretenden Grammatikregeln charakterisieren lassen, so handelt es sich wahrscheinlich um eine anomale Sequenz, beispielsweise durch ein ungewöhnliches Muster oder eine neue Trendentwicklung.

Da zum Erlernen von Grammatikregeln diskretisierte Daten benötigt werden, erfolgt die Diskretisierung in einem ersten Schritt mit dem Algorithmus SAX (***S**ymbolic **A**ggregation Appro**X**imation*), der aus Datensätzen bzw. -sequenzen äquivalente Symbole erzeugt [22]. Aufgrund der Natur von Subsequenzanomalien wird SAX per gleitendem Fenster angewandt. Innerhalb jedes Fensters wird die Subsequenz Z-normalisiert (vgl. Z-Scoring in Gl. 4.2) und anschließend in w gleichwahrscheinliche Segmente diskretisiert, mit w als Anzahl an Symbolen. Durch die Z-Normalisierung folgt die Sequenz einer Gauß-Verteilung und wird entspr. Tab. 4.1 sowie am Beispiel $w = 6$ in Abb. 4.7 unterteilt.

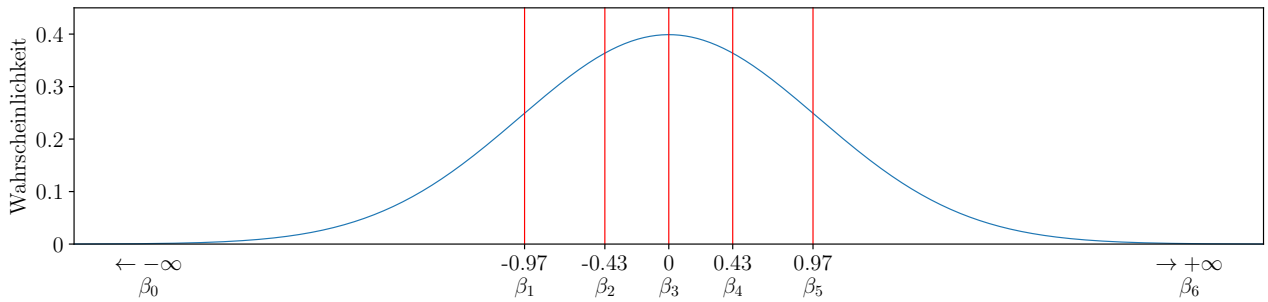


Abbildung 4.7: Normalverteilung mit $w = 6$ gleichwahrscheinlichen Segmenten gem. Tab. 4.1

β_i \ w	3	4	5	6	7	8	9	10
β_1	$-0,43$	$-0,67$	$-0,84$	$-0,97$	$-1,07$	$-1,15$	$-1,22$	$-1,28$
β_2	$0,43$	0	$-0,25$	$-0,43$	$-0,57$	$-0,67$	$-0,76$	$-0,84$
β_3		$0,67$	$0,25$	0	$-0,18$	$-0,32$	$-0,43$	$-0,52$
β_4			$0,84$	$0,43$	$0,18$	0	$-0,14$	$-0,25$
β_5				$0,97$	$0,57$	$0,32$	$0,14$	0
β_6					$1,07$	$0,67$	$0,43$	$0,25$
β_7						$1,15$	$0,76$	$0,52$
β_8							$1,22$	$0,84$
β_9								$1,28$

Tabelle 4.1: z-Werte einer Normalverteilung für Segmente mit gleichem Flächeninhalt ergo gleicher Wahrscheinlichkeit. β_i mit $1 \leq i \leq w-1$ sowie $\beta_0 = -\infty$ und $\beta_w = +\infty$ entsprechen den z-Koordinaten für die jeweilige Aufteilung der z-Achse. Quelle: [22]

Die diskretisierte Subsequenz produziert dann eine Menge an SAX Worten, die mit dem Grammatikinferenzalgorithmus Sequitur [21] weiterverarbeitet werden, um rekursiv kontextfreie Grammatikregeln aufzustellen. Mithilfe einer GUI kann ein Datensatz analysiert werden und mit der erzeugten **Rule Density Curve** sowie den einzeln aufgelisteten potenziell anomalen Sequenzen untersucht werden. Die Rule Density Curve gibt Aufschluss darüber, wieviele Grammatikregeln pro Datenpunkt greifen. Das heißt anschaulich: je mehr Grammatikregeln, desto normaler ist ein Datenpunkt bzw. eine Subsequenz [25].

4.2.4 Sliding Window Isolation Forest Density

Der Algorithmus namens **Sliding Window Isolation Forest Density** - kurz **SWIFD** - dient der Anomalieerkennung in Zeitreihen durch die Berechnung einer Anomaliedichtekarte mittels des Isolation Forest-Verfahrens [17]. Der schematische Ablauf ist dabei Abb. 4.8 zu entnehmen. Zunächst werden aus der Zeitreihe über gleitende Fenster statistische Merkmale extrahiert. Hierzu wird die

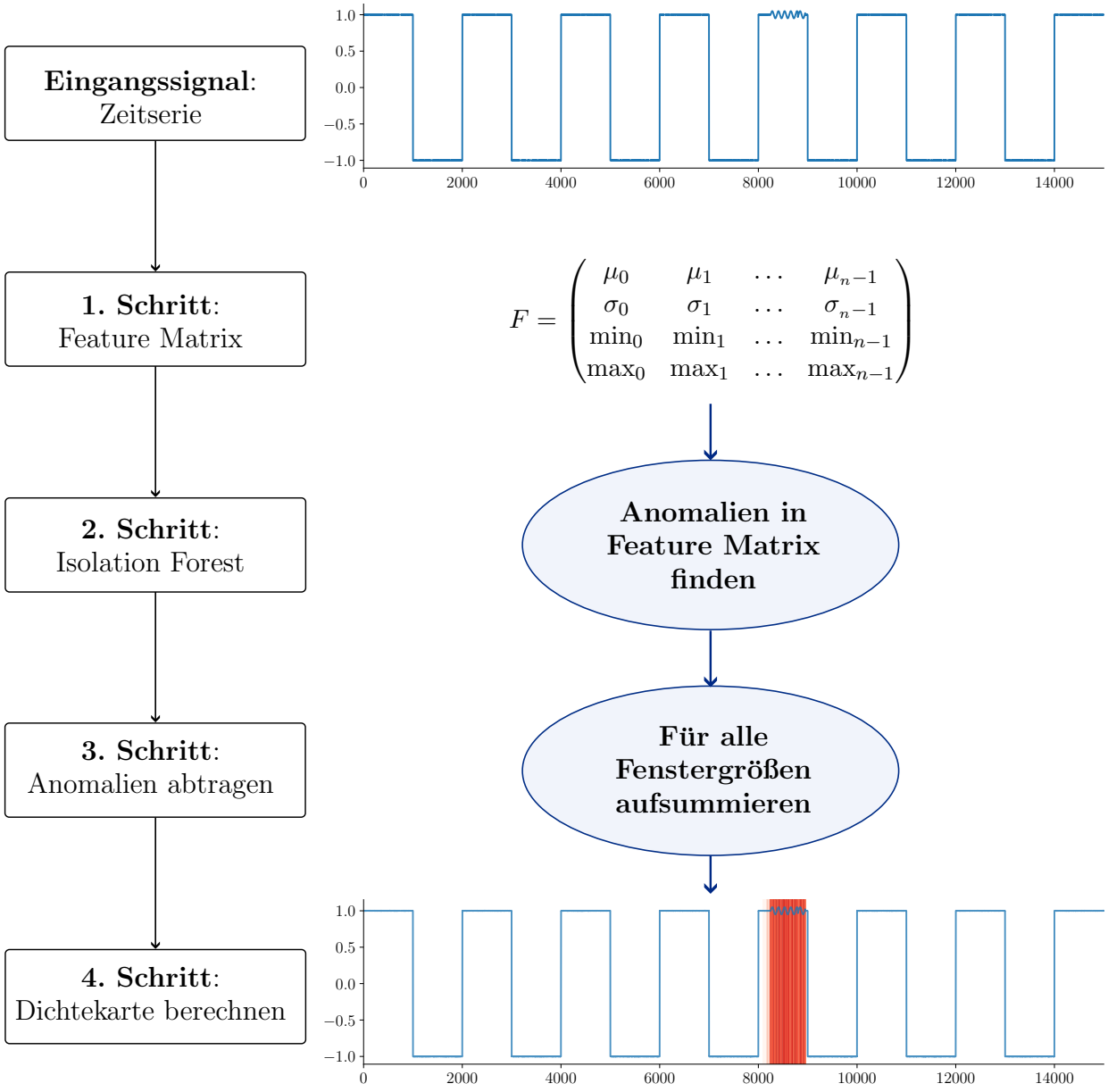


Abbildung 4.8: Schematischer Ablauf des SWIFD Algorithmus

Zeitreihe segmentiert, wobei jedes Fenster durch Mittelwert, Standardabweichung, Minimum und Maximum beschrieben wird. Die Fenstergröße wird aus einer vordefinierten Menge gewählt, wobei sich die Schrittweite in Abhängigkeit der Fenstergröße ergibt.

Die so gewonnenen Merkmalsvektoren werden anschließend mit dem Isolation Forest-Algorithmus verarbeitet, der auf der Grundidee von Liu et al. [17] basiert und um die gleitenden Fenster von Ding et al. [12] weiterentwickelt wurde. Dieser konstruiert Entscheidungsbäume, in denen isolierte Datenpunkte – also potenzielle Anomalien – mit kürzeren Pfaden identifiziert werden als reguläre Datenpunkte. Während herkömmliche Isolation-Forest-Ansätze binäre Entscheidungen über Anomalien treffen, integriert SWIFD eine Dichtebetrachtung. Anstatt einzelne Punkte als Anomalien zu markieren, wird für jedes Fenster ein Dichtewert berechnet, indem erkannte Anomalien aufsummiert werden.

Um lokale Schwankungen zu glätten und die visuelle Interpretierbarkeit zu verbessern, erfolgt abschließend eine Glättung der Anomaliedichte mithilfe eines Gaußschen Filters. Dieser Ansatz kombiniert somit die Effizienz von Isolation Forest mit einer kontinuierlichen Dichteschätzung, wodurch nicht nur isolierte Anomalien erkannt, sondern auch Bereiche mit einer erhöhten Anomaliewahrscheinlichkeit identifiziert werden können. Dies ist besonders nützlich für die Analyse von Zeitreihen mit strukturellen Veränderungen, bei denen einzelne Punktanomalien möglicherweise nicht ausreichen, um ein klares Bild der zugrunde liegenden Muster zu liefern.

4.2.5 Mahalanobis-Distanz mit SWIFD

Der Algorithmus **Mahalanobis-Distanz mit SWIFD** - kurz **MD-SWIFD** - wird zur Detektion von Korrelationsanomalien eingesetzt und kombiniert zwei wesentliche Konzepte: die Mahalanobis-Distanz und Isolation Forest, wie er in Abs. 4.2.4 bereits zur Subsequenzanomaliedetektion angewandt wird. Die Weiterentwicklung um die Mahalanobis-Distanz ermöglicht die Erkennung von Anomalien in der Korrelation in multivariaten Zeitserien [19].

Als eine Erweiterung der euklidischen Distanz wird sie in multivariaten Datensätzen verwendet, um die Entfernung eines Punktes von einem Mittelwert unter Berücksichtigung der Korrelationen zwischen den Dimensionen zu messen. Für einen Punkt x in einer multivariaten Zeitserie, den Mittelwert μ und die Kovarianzmatrix Σ, wird die Mahalanobis-Distanz D_M wie folgt berechnet:

$$D_M(x) = \sqrt{(x - \mu)^T \Sigma^{-1} (x - \mu)} \tag{4.3}$$

Die Mahalanobis-Distanz ist ein Maß dafür, wie stark ein Datenpunkt vom Mittelwert abweicht, wobei die Kovarianzmatrix die Streuung und die Korrelationen zwischen den Variablen der Zeitserie berücksichtigt. In Verbindung mit SWIFD wird der „zeitliche Verlauf“ der Mahalanobis-Distanz - also die Mahalanobis-Distanz, die zu jedem Zeitstempel der Zeitserie korrespondiert - als Analysesignal ver-

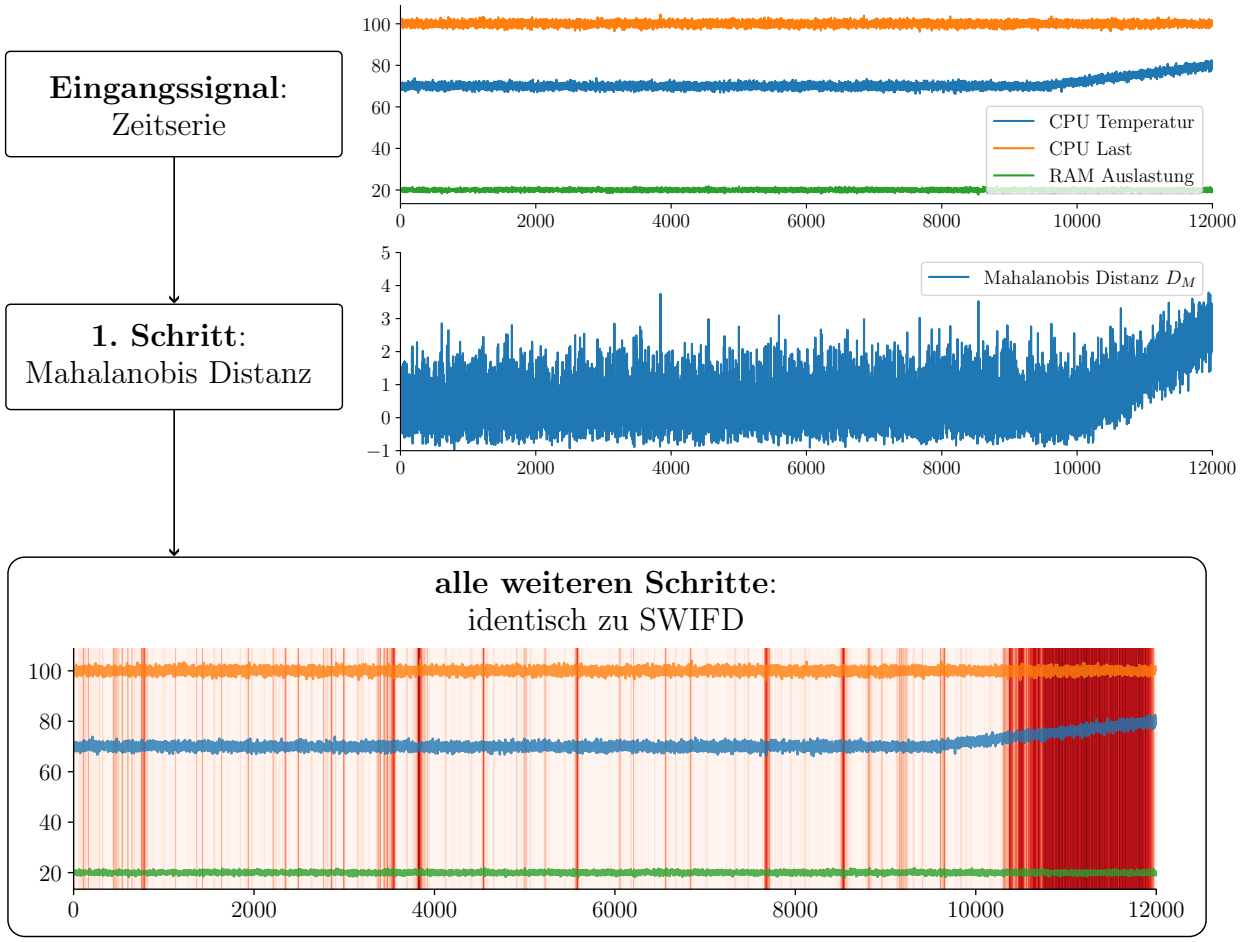


Abbildung 4.9: Schematischer Ablauf des MD-SWIFD Algorithmus als Erweiterung von SWIFD

wendet. In der Mahalanobis-Distanz stecken sämtliche Informationen über das Korrelationsverhalten der Zeitserie und so können am Verlauf der Distanz Anomalien detektiert werden.

MD-SWIFD macht sich also die Funktionalität von SWIFD zu Nutze und kann so in der weiterentwickelten Version Anomalien in der Korrelation mehrerer Variablen zuverlässig detektieren.

4.2.6 Elliptic Envelope Ansatz

Unter der Annahme, dass Daten einer multivariaten Normalverteilung folgen, wird der Elliptic Envelope verwendet, um Ausreißer in einem Datensatz zu identifizieren. Der Algorithmus geht davon aus, dass die Mehrheit der Datenpunkte innerhalb einer elliptischen Region um den Mittelpunkt der Verteilung liegt. Ziel des Verfahrens ist es, eine elliptische Grenze zu bestimmen, die die zentrale Datenverteilung beschreibt. Punkte, die außerhalb dieser Grenze liegen, gelten als Ausreißer.

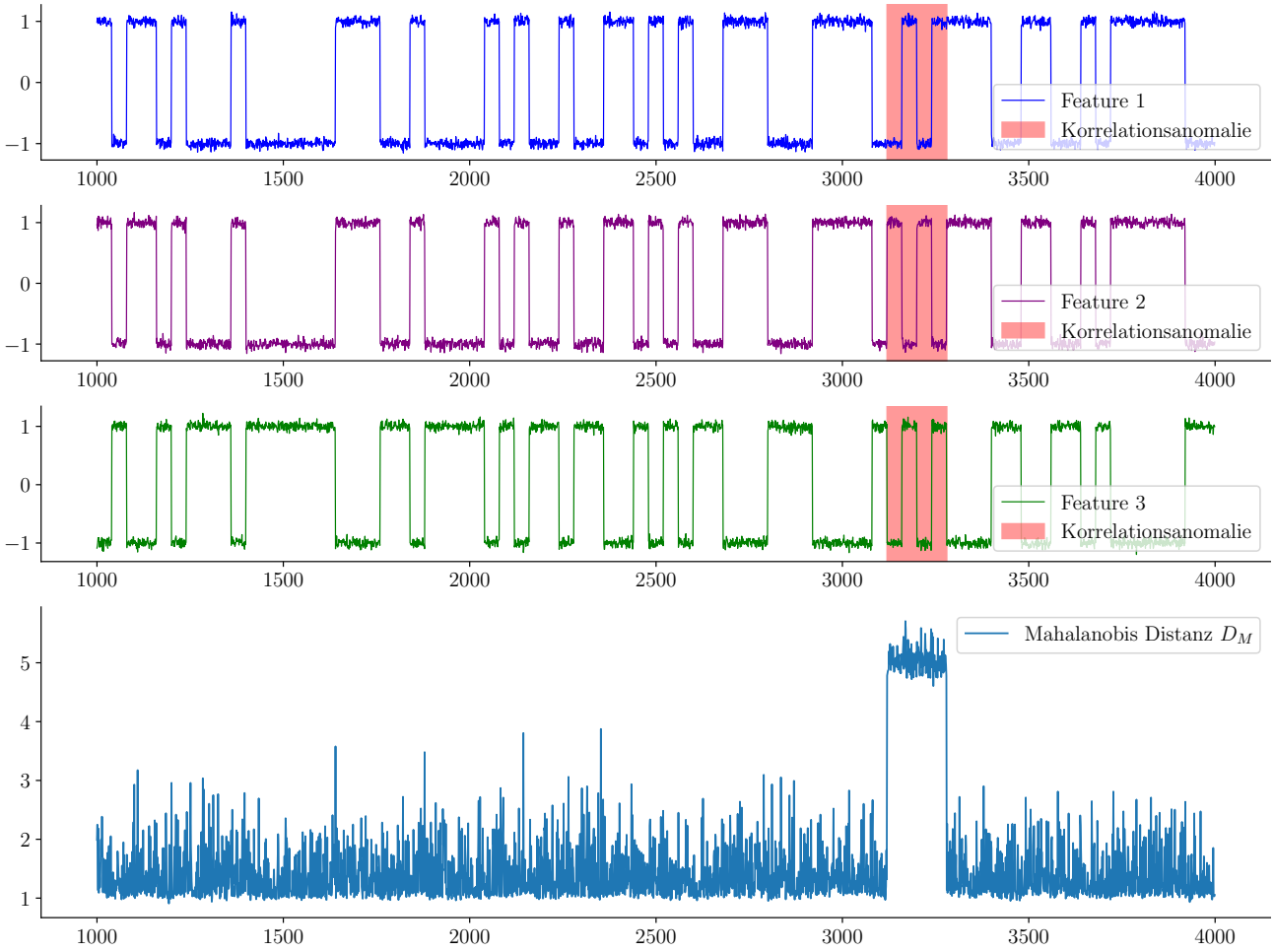


Abbildung 4.10: Ausschnitt aus dreidimensionaler Zeitreihe mit entsprechender Mahalanobis Distanz D_M. Anhand von D_M können Korrelationsanomalien durch Setzen eines Schwellwerts erkannt werden, wenn die Distanz oberhalb dieser Schwelle liegt.

Die Funktionsweise des Elliptic Envelope basiert auf der Schätzung der Kovarianzmatrix der Daten und der Bestimmung einer Ellipse, die diese am besten beschreibt. Mithilfe des Maximum-Likelihood-Verfahrens wird die multivariate Normalverteilung ermittelt, die die größten Anteile der Daten umfasst. Die Hauptachsen dieser Ellipse werden durch die Eigenwerte und Eigenvektoren der Kovarianzmatrix bestimmt. Um Ausreißer zu erkennen, wird eine Mahalanobis-Distanz berechnet, und Datenpunkte, die eine zu hohe Distanz aufweisen, werden als Anomalien identifiziert [2].

Der Elliptic Envelope wird häufig in Bereichen eingesetzt, in denen die Annahme einer Normalverteilung zutrifft, wie beispielsweise in der Analyse von verdächtigen Netzwerkaktivitäten [2]. Besonders nützlich ist der Algorithmus bei hochdimensionalen Daten, in denen einfache univariate Verfahren an ihre Grenzen stoßen. Da der Elliptic Envelope Korrelationen zwischen den Variablen berücksichtigt, kann er auch komplexere Datenstrukturen als univariate Methoden erfassen.

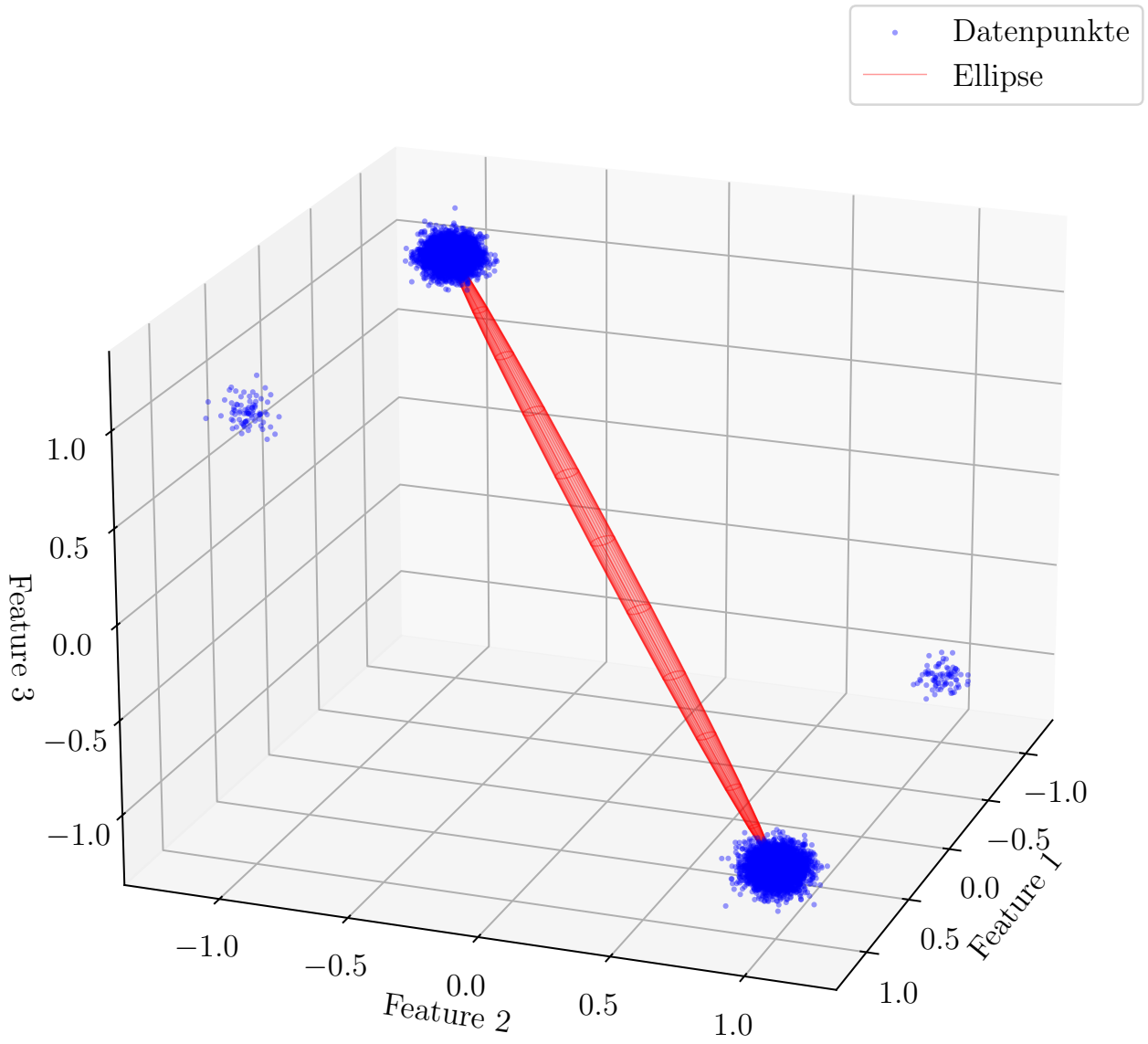


Abbildung 4.11: Ellipsoid auf Basis der Kovarianz der drei Kanäle

4.3 Semi-Supervised Learning Algorithmen

Eine weitere Möglichkeit der Anomaliedetektion ist die mit Algorithmen der Klasse **Semi-Supervised Learning**. Algorithmen dieser Klasse benötigen Trainingsdaten, die dem Normal entsprechen und sind daher zu Beginn der Implementierung zunächst unpraktischer als die der Klasse Unsupervised Learning. Der Grund dafür ist, dass erst eine erhebliche Menge an Training durchgeführt werden

muss, optimalerweise mit Daten, die den gesamten normalen Betriebsbereich abdecken können [9, S. 2-4] [24]. Dementsprechend werden Datenpunkte- oder sequenzen, die nicht dem Normal angehören oder als solches identifiziert werden können, als Anomalie gekennzeichnet.

Das Sammeln von großen Datensätzen ist kein Problem. Das SSP X1 System nimmt beispielsweise im Sekundentakt Daten auf und so entstehen schnell große Datensätze bzw. Zeitserien. Die Problematik liegt vielmehr darin, dass die Daten mit einem Label versehen werden müssen, zumindest implizit. Es muss sichergestellt sein, dass die gesamte vorliegende Zeitserie einem normalen Betriebszustand entspricht [9, S. 10 ff].

4.3.1 LSTM-Autoencoder

Einer in seiner Funktionalität interessanter Algorithmus ist der LSTM-Autoencoder (LSTM-AE). LSTM steht für Long-Short Term Memory, ist eine Erweiterung der Recurrent Neural Networks und erlaubt es, auf ein Langzeitgedächtnis zurückzugreifen und kommt gänzlich ohne Parametrisierung aus [15]. Ein Autoencoder ist ein Algorithmus, der auf Basis der Rekonstruktion von Sequenzen Anomalien detektiert und gehört demnach zur Kategorie der Subsequenzanomaliedetektion (vgl. Tab. 4.2).

Die Einordnung in die Klasse der Rekonstruktionsalgorithmen erfolgt nach Schmidl et al. [24] und beschreibt Rekonstruktionsalgorithmen als solche, die Subsequenzen in eine niederdimensionalere Domäne enkodieren, von wo sie wieder dekodiert bzw. rekonstruiert werden. Ein zu analysierender Datensatz wird also in Subsequenzen unterteilt und diese werden enkodiert. Mithilfe der Trainingsdaten werden die enkodierten Sequenzen rekonstruiert, wobei Abweichungen des Originals zur rekonstruierten Sequenz als Anomalie gelten dürfen, da sie mit den Trainingsdaten nicht übereinstimmen.

Die Kombination der beiden Prinzipien zum LSTM-AE erlaubt die Enkodierung der Subsequenzen unter Beibehalt langzeitiger Abhängigkeiten und Korrelationen und erweist sich so als sehr geeignet für große multivariate Zeitserien mit unterschiedlichen Abhängigkeiten und Korrelationen, wie sie in den Systemen der SSP X1 vorkommen [29].

Der Algorithmus wird im nächsten Kapitel denselben Tests unterzogen wie seine Unsupervised Learning Pendants. Schlussendlich ist auch eine Implementierung von LSTM-AE zur Anomaliedetektion im Kontext der SSP X1 und verwandter Systeme nicht auszuschließen.

4.4 Übersicht über alle ausgewählten Algorithmen

Abschließend folgt eine kurze Auflistung aller genannten Algorithmen mit stichwortartiger Kategorisierung nach Detektionsklasse, -prinzip sowie die Quelle der Implementierung. Unter dem Stichwort

Eigene Implementierung verbergen sich auch Komponenten, die aus Python Bibliotheken angewandt wurde, wie der Algorithmus zu **Isolation Forest** [23] oder die Implementierung des **LSTM** [1], die zur gewollten Funktionalität zusammengeführt und erweitert wurden. Im nächsten Kapitel erfolgt die Erprobung, Gegenüberstellung und Evaluierung sämtlicher Algorithmen derselben Detektionsklasse.

Algorithmus	**Detektionsklasse**	**Detektionsprinzip**	**Quelle**
HBOS	Punktanomalien	Histogramm	PyOD (Open Source) [32]
Sliding Window Z-Score	Punktanomalien	Z-Score	Eigene Implementierung
GrammarViz 2.0	Subsequenzanomalien	Grammatik	Open Source [25]
SWIFD	Subsequenzanomalien	Isolation Tree	Eigene Implementierung
MD-SWIFD	Korrelationsanomalien	Mahalanobis Distanz und Isolation Tree	Eigene Implementierung
Elliptic Envelope	Korrelationsanomalien	Kovarianzmatrix und Mahalanobis Distanz	sklearn (Open Source) [23]
LSTM-AE	Subsequenzanomalien	Rekonstruktion und Neuronale Netze	Eigene Implementierung

Tabelle 4.2: Übersicht über die verwendeten Algorithmen nach Kategorisierung in Detektionsklasse, Detektionsprinzip und Ursprung

Literaturverzeichnis

[1] Jason Ansel et al. “PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation”. In: *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*. ASPLOS ’24. La Jolla, CA, USA: Association for Computing Machinery, 2024, pp. 929–947. ISBN: 9798400703850. DOI: 10.1145/3620665.3640366. URL: https://doi.org/10.1145/3620665.3640366.

[2] Mohammad Ashrafuzzaman et al. “Elliptic Envelope Based Detection of Stealthy False Data Injection Attacks in Smart Grid Control Systems”. In: *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, Dec. 2020, pp. 1131–1137. DOI: 10.1109/ssci47803.2020.9308523.

[3] Jarosław Bernacki and Grzegorz Kołaczek. “Anomaly Detection in Network Traffic Using Selected Methods of Time Series Analysis”. In: *International Journal of Computer Network and Information Security* 7.9 (Aug. 2015), pp. 10–18. ISSN: 2074-9104. DOI: 10.5815/ijcnis.2015.09.02.

[4] Ane Blázquez-García et al. *A review on outlier/anomaly detection in time series data*. 2020. DOI: 10.48550/ARXIV.2002.04236.

[5] Paul Boniol and Themis Palpanas. “Detection of anomalies and identification of their precursors in large data series collections”. PhD thesis. 2021. URL: http://www.theses.fr/2021UNIP5206/document.

[6] Markus M. Breunig et al. “LOF: Identifying Density-Based Local Outliers”. In: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. SIGMOD/PODS00. ACM, May 2000, pp. 93–104. DOI: 10.1145/342009.335388.

[7] V. Ceronmani Sharmila et al. “Credit Card Fraud Detection Using Anomaly Techniques”. In: *2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)*. IEEE, Apr. 2019, pp. 1–6. DOI: 10.1109/iciict1.2019.8741421.

[8] Varun Chandola, Arindam Banerjee, and Vipin Kumar. “Anomaly detection: A survey”. In: *ACM Computing Surveys* 41.3 (July 2009), pp. 1–58. ISSN: 1557-7341. DOI: 10.1145/1541880.1541882.

[9] Olivier Chapelle, Alexander Zien, and Bernhard Schölkopf. *Semi-Supervised Learning*. Adaptive computation and machine learning series. Cambridge, Massachusetts: MIT Press, 2010. 1508 pp. ISBN: 9780262255899.

[10] Mooi Choo Chuah and Fen Fu. "ECG Anomaly Detection via Time Series Analysis". In: *Frontiers of High Performance Computing and Networking ISPA 2007 Workshops*. Springer Berlin Heidelberg, 2007, pp. 123–135. ISBN: 9783540747673. DOI: 10.1007/978-3-540-74767-3_14.

[11] Alan Davies and Alwyn Scott. *Starting to Read ECGs: A Comprehensive Guide to Theory and Practice*. Springer London, 2015. ISBN: 9781447149651. DOI: 10.1007/978-1-4471-4965-1.

[12] Zhiguo Ding and Minrui Fei. "An Anomaly Detection Approach Based on Isolation Forest Algorithm for Streaming Data using Sliding Window". In: *IFAC Proceedings Volumes* 46.20 (2013), pp. 12–17. ISSN: 1474-6670. DOI: 10.3182/20130902-3-cn-3020.00044.

[13] Markus Goldstein and Andreas Dengel. *Histogram-based Outlier Score (HBOS): A fast Unsupervised Anomaly Detection Algorithm*. Sept. 2012.

[14] Frank E. Grubbs. "Procedures for Detecting Outlying Observations in Samples". In: *Technometrics* 11.1 (Feb. 1969), pp. 1–21. ISSN: 1537-2723. DOI: 10.1080/00401706.1969.10490657.

[15] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 1530-888X. DOI: 10.1162/neco.1997.9.8.1735.

[16] Shanshan Jiang et al. "Credit Card Fraud Detection Based on Unsupervised Attentional Anomaly Detection Network". In: *Systems* 11.6 (June 2023), p. 305. ISSN: 2079-8954. DOI: 10.3390/systems11060305.

[17] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. "Isolation-Based Anomaly Detection". In: *ACM Trans. Knowl. Discov. Data* 6.1 (Mar. 2012). ISSN: 1556-4681. DOI: 10.1145/2133360.2133363. URL: https://doi.org/10.1145/2133360.2133363.

[18] Deepak T. Mane et al. "Detection of Anomaly using Machine Learning: A Comprehensive Survey". In: *International Journal of Emerging Technology and Advanced Engineering* 12.11 (Nov. 2022), pp. 134–152. ISSN: 2250-2459. DOI: 10.46338/ijetae1122_15.

[19] G. J. McLachlan. "Mahalanobis distance". In: *Resonance* 4.6 (June 1999), pp. 20–26. ISSN: 0973-712X. DOI: 10.1007/bf02834632.

[20] Felix Naumann. *CoMuT - Correlated Multivariate Time Series*. 2024. URL: https://hpi.de/naumann/s/comut.

[21] C. G. Nevill-Manning and I. H. Witten. "Identifying Hierarchical Structure in Sequences: A linear-time algorithm". In: (1997). DOI: 10.48550/ARXIV.CS/9709102.

[22] P. Patel et al. "Mining motifs in massive time series databases". In: *2002 IEEE International Conference on Data Mining, 2002. Proceedings.* ICDM-02. IEEE Comput. Soc, pp. 370–377. DOI: 10.1109/icdm.2002.1183925.

[23] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[24] Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. “Anomaly detection in time series: a comprehensive evaluation”. In: *Proceedings of the VLDB Endowment* 15.9 (May 2022), pp. 1779–1797. ISSN: 2150-8097. DOI: 10.14778/3538598.3538602.

[25] Pavel Senin et al. “GrammarViz 2.0: a tool for grammar-based pattern discovery in time series”. In: *Machine Learning and Knowledge Discovery in Databases*. Springer, 2014, pp. 468–472.

[26] Pavel Senin et al. “Time series anomaly discovery with grammar-based compression”. In: *International Conference on Extending Database Technology*. 2015. URL: https://api.semanticscholar.org/CorpusID:9124282.

[27] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to data mining*. New internat. edition. Always learning. Harlow: Pearson, 2014. 732 pp. ISBN: 9781292026152.

[28] *Ventrikuläre Tachykardie (VT)*. Aufgerufen: 08.01.2025. URL: https://ekgecho.de/thema/ventrikulaere-tachykardie-vt-ekg-kriterien-ursachen-klassifikation-behandlung-management/.

[29] Yuanyuan Wei et al. *LSTM-Autoencoder based Anomaly Detection for Indoor Air Quality Time Series Data*. 2022. DOI: 10.48550/ARXIV.2204.06701.

[30] Phillip Wenig. “Finding, Clustering, and Classifying Anomalies on Large and Multivariate Time Series”. en. PhD thesis. 2024. DOI: 10.25932/PUBLISHUP-66043.

[31] Phillip Wenig, Sebastian Schmidl, and Thorsten Papenbrock. “Anomaly Detectors for Multivariate Time Series: The Proof of the Pudding is in the Eating”. In: *2024 IEEE 40th International Conference on Data Engineering Workshops (ICDEW)*. IEEE, May 2024, pp. 96–101. DOI: 10.1109/icdew61823.2024.00018.

[32] Yue Zhao, Zain Nasrullah, and Zheng Li. “PyOD: A Python Toolbox for Scalable Outlier Detection”. In: *Journal of Machine Learning Research* 20.96 (2019), pp. 1–7. URL: http://jmlr.org/papers/v20/19-011.html.