

Bachelorarbeit

**Detektion von Zeitreihenanomalien in der
Niederspannung**

Joël Haubold

Juni 2020

Gutachter:

Prof. Dr. Rudolph

Dr.-Ing. Sebastian Ruthe

Technische Universität Dortmund
Fakultät für Informatik
Lehrstuhl für Computational Intelligence (LS-11)
<https://ls11-www.cs.tu-dortmund.de/>

Inhaltsverzeichnis

1 Einleitung	1
1.1 Motivation und Hintergrund	1
1.1.1 Anomaliererkennung auf Zeitreihen	1
1.1.2 Analyse des Niederspannungsnetzes	2
1.2 Aufbau der Arbeit	2
2 Grundlagen	3
2.1 Notationen	3
2.2 Anomalien	3
2.2.1 Anomalytypen	3
2.2.2 Komplikationen	4
2.3 Anomalieerkennung durch maschinelles Lernen	6
2.3.1 Überwachtes und unüberwachtes Lernen	6
2.3.2 Input und Output von Anomalieerkennungsverfahren	6
2.3.3 Robustheit	7
2.3.4 Streaming Data	7
2.3.5 Kriterien zur Performancebeurteilung	8
2.3.6 F-Measure	8
2.4 Arten von Anomalieerkennungsverfahren	8
3 Robust Random Cut Forest	9
3.1 RRCF Theory	9
3.1.1 RRCF Aufbau	10
3.1.2 Distanzbeibehaltung bei der RRCT Konstruktion	11
3.1.3 RRCF Instandhaltung	12
3.2 Anomalieerkennung über RRCF	14
3.2.1 Modellkomplexität eines RRCT	14
3.2.2 Verschiebung der Modellkomplexität durch einen Punkt \mathbf{x}	16
3.2.3 Codisp	18
4 Support Vector Machine	23

5 Tests auf Niederspannungsdaten	25
5.1 Aufmachung der Testdaten	25
5.1.1 Eignung der Daten für überwachte und unüberwachte Anomalieerkennung	27
5.1.2 Benötigte Eigenschaften eines Anomalieerkennungsverfahrens	27
5.2 Testen des RRCF Verfahrens	28
5.2.1 Implementierung der Tests	28
6 Fazit	31
A Weitere Informationen	33
Abbildungsverzeichnis	35
Tabellenverzeichnis	37
Algorithmenverzeichnis	39
Literaturverzeichnis	42
Erklärung	42

Kapitel 1

Einleitung

1.1 Motivation und Hintergrund

1.1.1 Anomaliererkennung auf Zeitreihen

Anomalieerkennung auf Zeitreihen ist ein weitreichendes Forschungsgebiet, sowohl an der großen Zahl möglicher Vorgehensweise gemessen, als auch an der Vielfalt der Anwendungsbereiche. [8] Einige Beispiele für den Nutzen der Erkennung von Anomalien darstellen sind:

- Finanzmärkte: Abrupte Einbrüche im Finanzmarkt müssen möglichst Frühzeitig erkannt werden um sich ausbreitenden Schaden zu verhindern oder einzudämmen.
- Benutzerhandlungen: Zeichnen sich Auffälligkeiten im Verhalten eines Benutzers ab so kann dies auf Situationen mit Handlungsbedarf hindeuten. So kann zum Beispiel etwaigen ungewollten Eingriffen in ein Computersystem entgegengewirkt werden.
- Biologische Daten: Zwar nicht direkt Zeitabhängig so können bestimmte biologische Forschungsprozesse, wie das platzieren einzelner Aminosäuren, analog zu temporalen Daten mit Methoden zur Zeitreihen anomalieerkennung unterstützt werden.
- Sensordaten: Viele physikalischen Anwendungen wird deren Verlauf anhand umfassender Sensordaten überwacht. Die hohe Quantität an Daten die kontinuierlich erfasst werden, macht es unmöglich diese alle per Hand auszuwerten und so kann automatisierte Anomalieerkennung dazu genutzt werden Ereignisse und Zusammenhänge in diesen Daten zu entdecken die ansonsten unbemerkt geblieben wären.

Diese Arbeit beschäftigt sich spezifisch mit dem folgend erläuterten Sensordatensatz, auf dem sie zwei Unterschiedliche Methoden miteinander vergleicht.

1.1.2 Analyse des Niederspannungsnetzes

Das deutsche Verteilnetz wurde ursprünglich mit dem Ziel gebaut, den in Großkraftwerken produzierten Strom und über das Transportnetz in die einzelnen Regionen Deutschlands transportiert wird, regional an die Endkunden (sowohl Industrie- und Gewerbekunden als auch Haushalte) zu verteilen. Das Verteilnetz ist dabei baumartig strukturiert und besteht aus der Hochspannungsebene die den Übergabepunkt des Transportnetz enthält und sich hin zur Mittelspannungsebene, Niederspannungsebene und schließlich den Endkunden verzweigt.

Mit zunehmender Integration von Erneuerbaren Energien wie Wind- und PV-Anlagen in die Mittel- und Niederspannungsebene steigt auch die Dynamik in den unteren Spannungsebenen. Lastflüsse die vorher stets von oben (Hochspannung) nach unten (Mittel-, Niederspannung) gerichtet waren, kehren sich in Teilen um und können zu einer lokal höheren Auslastung des Netzes führen. Hinzu kommen neue Verbraucher wie z.B. Elektrofahrzeuge die insbesondere in den frühen Abendstunden und über die Nacht verteilt das Netz stärker belasten.

Um diese Effekte erkennen und analysieren zu können, müssen die Niederspannungsebene zunächst messtechnisch erfasst werden. Die Firma PPC hat ein Messgerät entwickelt, welches sich in Ortsnetzstationen (Übergabepunkt von Mittel- zu Niederspannung) einbauen lässt und dort eine dreiphasige Spannungsmessung durchführen kann. Zusätzlich verfügt das Messgerät über eine Kommunikationsanbindung mit der sich die Daten abrufen und an einem zentralen Punkt aggregieren und auswerten lassen. Eine Teilmenge dieser Daten sind nun Bestand dieser Arbeit.

Datensatz

Maybe here???

1.2 Aufbau der Arbeit

In dieser Arbeit werden zuerst in Kapitel 2 die Grundsätze von Anomalieerkennung und mögliche Komplikationen die sie mit sich bring erläutert. In Kapitel 3 werden die zwei in der Arbeit eingesetzten Verfahren Robust Random Cut Forest", und Öne Dimensional Support Vector Machine erläutert. In Kapitel 4 wird auf die im Rahmen dieser Arbeit angewendete Implementierung und deren Ergebnisse eingegangen, sowie wie diese Ergebnisse geneinander Abschneiden. In Kapitel 5 wird, auf Basis dieser Ergebnisse, ein Fazit gezogen.

Kapitel 2

Grundlagen

2.1 Notationen

Die in dieser Arbeit verwendeten Notationen lehnen sich an die in dem Papier [6] verwendeten an:

- \mathbb{E} ddd
- $\mathbb{P}r$ ddd
- \mathcal{T} ddd

2.2 Anomalien

In einem gegebenen Datensatz Z an Punkten, wird einer dieser Punkte $x \in Z$ als Outlier bezeichnet, falls er sich signifikant in einen oder mehreren seiner Merkmale von den Punkten $Z - \{x\}$ unterscheidet. Seien Y alle anomalen Punkte aus Z . Ein Modell welches Z darstellt ist entsprechend wesentlich komplexer als ein Modell welches $Z - Y$, also nur die nicht-normalen *Inliners* von Z darstellt. Wie stark sich x in seinen Merkmalen von anderen Punkten in Z unterscheiden muss, beziehungsweise wie stark x die Komplexität des Modells von Z erhöht, damit x als Anomalie gesehen wird ist hängt oft von der jeweiligen Zielsetzung ab.

2.2.1 Anomalytypen

Grundsätzlich lassen sich Anomalien darüber inwiefern sie sich von den Inlinern abheben in drei Klassen unterteilen: [2]

- *Punktanomalien*: Wenn ein Datenpunkt sich stark von den normalen Merkmalsausprägungen im Datenset unterscheidet. Beispielsweise wäre bei Beobachtung des

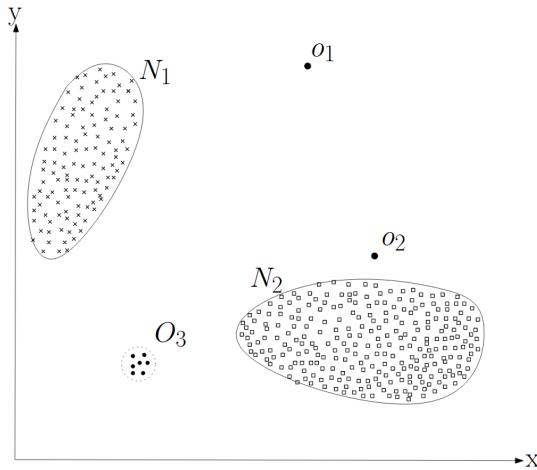


Abbildung 2.1: Ein Beispieldatensatz mit zwei Anomalien o_1 und o_2 , sowie eine Punktegruppe O_3 von 7 Anomalien. Die Gruppen N_1 und N_2 stellen die Inliner des Datensatzes da. Quelle: [4]

Kraftstoffverbrauchs pro Tag eines Autos ein Verbrauch von 50 Litern, mit einem normalen Verbrauch von 5 Litern pro Tag eine Punktanomalie

- *Kontextanomalien*: Wenn ein Datenpunkt in einem bestimmten Kontext in seinem Datensatz hervorsticht. Zum Beispiel können bei der Anomalieerkennung auf den Ausgaben einer Person, überdurchschnittlich hohe Ausgaben hohe Ausgaben an einem Feiertag normal sein, im Kontext eines Arbeitstages allerdings eine Anomalie darstellen.
- *Kollektivanomalien*: Wenn mehrere, über ein oder mehrere ihrer Merkmale zusammenhängende Datenpunkte, welche alleine keine Besonderheit darstellen würden, zusammen eine Anomalie darstellen. Beispielsweise sind bei einem Elektrokardiogramm (EKG) einzelne niedrige Werte Teil einer Inlinergruppe, eine Reihe lange zeitlich aufeinanderfolgender Werte allerdings ist eine Anomalie.

2.2.2 Komplikationen

Die Diversität von möglichen Datensätzen und deren Merkmalen macht es generell nicht möglich, ein allgemeines Vorgehen für die Erkennung von Anomalien zu bestimmen. Dazu kommen mögliche Eigenschaften die dies weiterhin erschweren, oder es bestimmten Vorgehen sogar unmöglich machen, Anomalie von Inliner zu unterscheiden. Ein Überblick über einige dieser ist hier aufgeführt:

Kontextabhängigkeit

Es ist zu beachten das bei zwei anomalen Punkten nicht die gleichen Grenzwerte für die einzelnen Merkwerte gelten müssen, es kommt vielmehr auf die Kombination der Merkmale

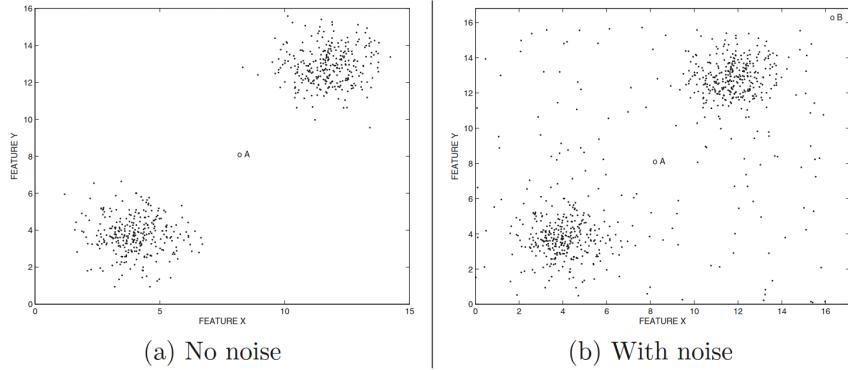


Abbildung 2.2: Der Einfluss von Rauschen auf einen Datensatz bestehend aus zwei Inlinergruppen und einem anomalen Punkt A . Quelle: [1]

an. Ein einfaches Beispiel ist ein über die Zeit stetig zunehmender Messwert. Ein Punkt dessen Wert zu Beginn aus der Zeitreihe nach oben ausreißt, ist wahrscheinlich anomal. Die Punkte die später durch den Trend der Zeitreihe diesen Wert überschreiten, sind deswegen aber nicht zwingend selber anomal, noch invalidieren sie den Status des Ausreißers als Anomalie. [10]

Duplikate

Erschwerend für die Anomalieerkennung kann es sein falls sich mehrere Anomalien eines Datensatzes ähneln, wie in Abbildung 2.1. Während sich die Punkte in O_3 eindeutig von den beiden Inliner-Punktegruppen N_1 und N_2 abgrenzen, so haben sie alleinstehend betrachtet dennoch untereinander eine starke Ähnlichkeit., ein Modell des dargestellten Datensatzes vereinfacht sich durch die einzelne Entfernung eines Punktes aus O_3 nicht. [6] Sollen die Punkte in O_3 von einem Anomalieerkennungsverfahren als Anomalie eingestuft werden, so muss entweder dem Verfahren mitgeteilt werden das Inliner Ähnlichkeiten zu den Punkten in N_1 und N_2 haben müssen, oder es muss so kalibriert werden, dass eine Ansammlung von 7 ähnlichen Punkten noch nicht als Inlinergruppe gesehen wird. Mehr dazu in Sektion 2.3.1

Rauschen

Je nach generierenden Prozess des Datensatzes kann es sein das in diesem neben der zu beobachtenden Größe, weitere Punkte aufgenommen werden, welche sich in ihren Merkmalen stark von den Inlinern unterscheiden, aber nicht von Relevanz für den Beobachter des Prozesses sind. [1] In den beiden Abbildungen 2.2 ist die Schwierigkeit die Rauschen bei der Anomalieerkennung mit sich bringt zu sehen. In Abbildung 2.2 (a) ist der Punkt A offensichtlich anomal. In 2.2 (b) könnte dieser allerdings Teil des Rauschens sein. Um den Punkt A als anomal markieren zu können, aber nicht den Rest des uninteressanten Rau-

schen, muss dem Anomalieerkennungsverfahren mitgeteilt werden das Punkte mit seinen Merkmalen als anomale gelten.

Mehrdimensionalität

Hat der zu untersuchende Datensatz eine hohe Dimensionalität in seinen Merkmalen, führt dies zu weiteren Problemen bei der Anomalieerkennung. Mit zunehmender Anzahl an Merkmalsdimensionen erhöhen sich die möglichen Kombinationen an Dimensionen auf denen nach anomalen Merkmalen gesucht werden kann exponentiell, womit der Aufwand der Anomalieerkennung ansteigen kann. Weiterhin führt diese Zunahme der möglichen Dimensionskombinationen auf denen gesucht werden kann, dass es immer wahrscheinlicher wird, für jeden Punkt mindestens eine solche Kombination zu finden, dass er auf dieser anomale ist. Umgekehrt wird es mit zunehmenden Dimensionen, auf denen man nach anomalen Ausprägungen suchen kann, schwieriger die relevanten Dimensionen zu finden. Es entsteht effektiv ein Rauschen, da die relevanten Dimensionen gegenüber den nicht relevanten untergehen. [5]

2.3 Anomalieerkennung durch maschinelles Lernen

Ein Anomalieerkennungsverfahren bietet generalisiert die Funktion auf einem Datensatz Anomalien zu erkennen. Dabei eignen sich nicht alle Verfahren für alle Datensätze, sei es weil sie für eine bestimmte Eigenschaft des Datensatzes nicht geeignet sind, oder umgekehrt weil sie zur Leistungsverbesserung bestimmte Eigenschaften im Datensatz voraussetzen.

2.3.1 Überwachtes und unüberwachtes Lernen

Generell lassen sich zur Anomalieerkennung angewandte maschinelle Lernverfahren in zwei Bereiche teilen, überwachtes und unüberwachtes Lernen:

Überwachtes Lernen

Überwachtes Lernen

Unüberwachtes Lernen

2.3.2 Input und Output von Anomalieerkennungsverfahren

Weiter Unterscheidungen lassen sich über Anomalieerkennungsverfahren darin machen, in welcher Form der Input auf Anomalien untersucht wird, und in Welcher Form das Anomalieerkennungsverfahren seine Ergebnisse ausgibt.

Arten von zu analysierenden Dateninstanzen

Auch darin in welcher Form die Anomalien erkannt werden sollen unterscheiden sich die möglichen Verfahren. Je nach Zielsetzung kann in einer Zeitreihe nach einzelnen oder Sequenzen von anomalen Datenpunkten gesucht, oder es können Zeitabschnitte nach Aufälligkeiten miteinander verglichen werden. Anders mag es auch von Nutzen seien, ganze Zeitreihen aus einer Gruppe von Zeitreihen als anomal zu bestimmen. [8]

Ergebnisse des Anomalieerkennungsverfahren

Das Ergebnis eines Anomalieerkennungsverfahrens, stellt die Beurteilung des Verfahrens gegenüber den eingegebenen Datensatz dar, ob die Eingabe oder die Elemente die diese ausmacht anomalous oder nicht sind, beziehungsweise um welche Art von Anomalie es sich handelt. Allgemein kann man zwischen zwei Ausgabearten der Ergebnisse unterscheiden: [2]

- *Bewertung*: Bei bewertenden Anomalieerkennungsverfahren wird jeder zu bewertenden Dateninstanz, ein Wert zugeordnet, dessen Größe darstellt wie sicher sich das Verfahren ist, ob die Instanz eine Anomalie ist. Entweder werden diese Werte dann einer genaueren Betrachtung unterzogen, oder es wird eine Grenze festgelegt, ab welchen Wert eine Dateninstanz als Anomalie interpretiert wird.
- *Kennzeichnung*: Bei einem kennzeichnenden Anomalieerkennungsverfahren bestimmt das Verfahren im Alleingang, ob eine Dateninstanz eine Anomalie ist oder nicht, beziehungsweise zu welcher Anomaliekategorie es gehört.

2.3.3 Robustheit

Die Robustheit eines Algorithmus beschreibt seine Stabilität gegenüber Anomalien im Trainingsdatensatz und gegenüber ungewollten Unterschieden zwischen dem Trainingsdatensatz und dem Testdatensatz. Weiterhin kann ein Anomalieerkennungsverfahren besonders Robust gegenüber einer Eigenschaft von Datensätzen, wie zum Beispiel Rauschen oder Mehrdimensionalität, sein, die sich allgemein negativ auf die Performance von auf ihrem Datensatz ausgeführten Algorithmen auswirkt.

2.3.4 Streaming Data

Space Time Anpassung des Modells, Live Ergebnisse

2.3.5 Kriterien zur Performancebeurteilung**2.3.6 F-Measure****2.4 Arten von Anomalieerkennungsverfahren**

Kapitel 3

Robust Random Cut Forest

In diesem Kapitel wird einer der beiden, auf den PPC Datensatz angewendeten Verfahren, der **Robust Random Cut Forest** (von hier an RRCF) in seinen Grundzügen beschrieben. Das Kapitel orientiert sich dabei an Artikel [6] und dem zugehörigen Supplement [7].

3.1 RRCF Theory

Der RRCF basiert auf, und ähnelt somit vielerlei dem in Kapitel 2 vorgestellten Isolation Forest. So versucht der RRCF ebenfalls Anomalien direkt vom Datensatz zu isolieren statt ein Profil einer normalen Klasse zu definieren. Auch basiert der RRCF ebenfalls auf dem Zufallsprinzip, und mittelt sein Ergebnis aus den einzelnen Ergebnissen der unabhängig konstruierten Bäume aus denen er besteht. Unterscheiden tut sich der RRCF allerdings in zweierlei Hinsicht:

1. Bei der Konstruktion der Bäume des RRCFs, werden die Dimensionen über die der zugrundeliegende Datensatz geteilt wird nicht uniform-zufällig, sondern nach der Größe der in ihnen vorhandenen Unterschieden der Punkte des Datensatzes gewichtet ausgewählt. So kann der Einfluss von unwichtigen Dimensionen (siehe Sektion 2.2.2) reduziert werden, und die Zugrundeliegenden Wahrscheinlichkeiten jedes Baumes über einen Datensatz bleiben konstant, unabhängig davon wie dieser Baum zustande kam.
2. Das Kriterium nach dem die Ausgabe des RRCFs berechnet wird bezieht sich nicht auf die Tiefe der Punkte, sondern auf den Effekt die eine beliebige diesen Punkt beinhaltende Gruppe von Punkten, auf die gesamte Modellkomplexität des Baumes hat. Diese Metrik ist allgemein robuster, insbesonders können Duplikate einer Anomalie nicht mehr ihre Erkennung als solche verhindern

In den folgenden Sektionen werden diese Unterschiede, sowie die dem RRCF zugrunde liegenden Theoreme dargestellt.

Tabelle 3.1: Ein Beispiel Datensatz über 3 Dimensionen mit numerischen Werten mit $S = \{x, y, z\}$ sowie die von Definition 3.1.1 in Schritt 1 berechnete Wahrscheinlichkeit $\frac{l_i}{\sum_j l_i}$ das S in Schritt 3 über die jeweilige Dimension partitioniert wird

Dimension	x	y	z	$\frac{l_i}{\sum_j l_i}$
1	5	10	6	$\frac{5}{35}$
2	2	8	12	$\frac{10}{35}$
3	25	5	5	$\frac{20}{35}$

3.1.1 RRCF Aufbau

Analog zu anderen Forest-Ansätzen aus dem Gebiet des maschinellen Lernens, besteht ein RRCF aus mehreren unabhängig voneinander konstruierten **Robust Random Cut Trees** (RRCT):

3.1.1 Definition (RRCT). Ein RRCT wird über ein Datensatz S mit j Dimensionen wie folgt generiert:

1. Wähle eine Dimension i aus den j Dimensionen. Dabei hat jede Dimension eine Wahrscheinlichkeit proportional zu $\frac{l_i}{\sum_j l_i}$, mit $l_i = \max_{x \in S} x_i - \min_{x \in S} x_i$ ausgewählt zu werden.
2. Wähle $X_i \sim \text{Uniform}[\min_{x \in S} x_i, \max_{x \in S} x_i]$
3. Teile S in $S_1 = \{x \mid x \in S, x_i \leq X_i\}$ und $S_2 = S \setminus S_1$ und fahre rekursiv auf S_1 und S_2 fort, solange $|S_1| > 1$ beziehungsweise $|S_2| > 1$.

In Schritt 1 wird die Dimension ausgewählt über die der Datensatz bei der Konstruktion des Baumes getrennt wird. Ein wichtiger Unterschied bei der Konstruktion eines RRCT zu der Konstruktion eines Baumes in einem Isolation Forest, wie in [9], ist dabei, dass die zur Trennung genutzte Dimension i nicht Uniform über alle Dimensionen j ausgewählt wird. Stattdessen werden die Dimensionen proportional dazu wie stark die Werte der einzelnen Punkte sich in den Dimensionen unterscheiden gewichtet bevor eine von ihnen gewählt wird.

In Schritt 2 wird darauf analog zum Isolation Forest Verfahren ein Trennwert X_i uniform aus der Wertespanne aller Punkte $x \in S$ der in Schritt 1 ausgewählten Dimension gewählt.

In Schritt 3 wird der Datensatz S dann über X_i partitioniert, sodass S_1 die Datenpunkte enthält die in Dimension i größer oder gleich groß wie X_i sind und S_2 die verbliebenen Datenpunkte, welche in i einen kleiner als X_i sind.

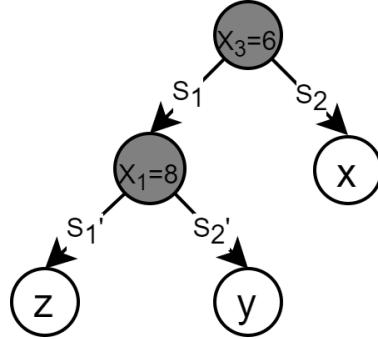


Abbildung 3.1: Ein möglicher, nach Definition 3.1.1 konstruierter RRCF über den in Tabelle 3.1 dargestellten Datensatz S . Die erste Partition erfolgte über die dritte Dimension mit einem nach Schritt 2 zufällig bestimmten Grenzwert von 6. Da S_1 darauf mehr als einen Punkt enthielt erfolgte eine weitere Partition über die erste Dimension und einen Grenzwert von 8

Beispielhaft würden in dem Datensatz von Tabelle 3.1 bei dem ersten Durchlauf der Baumkonstruktion die Dimensionen 1, 2 und 3 mit einer jeweiligen Wahrscheinlichkeit von $\frac{1}{7}$, $\frac{2}{7}$ und $\frac{4}{7}$, als die Dimension über die S partitioniert wird, ausgewählt werden. Je nach gewählter Dimension wird X_i darauf aus den Wertespannen $[5, 10]$, $[2, 12]$ beziehungsweise $[5, 25]$ uniform-zufällig gewählt. Ein möglicher RRCF, welcher sich aus dem in Tabelle 3.1 dargestellten Datensatz ergibt ist in Abbildung 3.1 dargestellt.

Jeder innere Knoten eines RRCTs $T = \mathcal{T}(S)$, über einen Datensatz S , entspricht demnach einer Partition, und enthält die entsprechende Dimension und den Grenzwert für diese Partition. Die Blätter des RRCTs entsprechen den einzelnen Punkten in S , welche über eine Reihe von Partitionen, entsprechend der Knoten entlang des Pfades von der Wurzel von T zu dem jeweiligen Blatt, von allen anderen Punkten in S isoliert wurden.

3.1.2 Distanzbeibehaltung bei der RRCT Konstruktion

Damit ein RRCF zur Anomalieerkennung eingesetzt werden kann, muss gezeigt werden, dass die RRCTs in der er die Punkte des zu untersuchenden Datensatzes auf eine Art speichert, die die Distanz zwischen den Punkten Beibehält. Ein Datenpunkt der sich im Datensatz anomalous abzeichneten muss, auch in einem aus diesem Datensatz gebauten RRCT als anomalous erkennbar sein. Dies ist gegeben durch folgendes Theorem:

3.1.2 Theorem (Distanzbeibehaltung). *Sei ein RRCT \mathcal{T} über einen Datensatz S mit d Dimensionen konstruiert. Sei das Gewicht eines Knotens von \mathcal{T} die Summe der Länge der Kanten der minimal begrenzenden Box der diesem Knoten untergeordneten Punkte $\sum_i l_i$, und sei die Baumdistanz zwischen zwei Knoten $u, v \in S$ das Gewicht des letzten*

gemeinsamen Vorfahrens von u und v . Dann ist die Baumdistanz von u und v mindestens $L_1(u, v)$ und in Erwartung maximal ein Vielfaches von $L_1(u, v)$ um den Faktor:

$$\mathcal{O}(d \log \frac{|S|}{L_1(u, v)}) \quad (3.1)$$

Beweis von Theorem 3.1.2

Sei für einen Datensatz S l_i erneut als die Wertespanne zwischen den niedrigsten und höchsten Wert von S in der Dimension i definiert. Sei $B(S)$ die *MinimalBoundingBox*(MBB) um alle Punkte in S . Sei dann $P(S) = \sum_i l_i$ die Summe der Seitenlängen von $B(S)$. Es ergibt sich:

3.1.3 Lemma. *Die Wahrscheinlichkeit das $u, v \in S$ durch eine Partition von S nach Definition 3.1.1 getrennt werden ist gegeben durch:*

$$\frac{1}{P(S)} \sum_i |u_i - v_i| \quad (3.2)$$

$P(S)$ entspricht der Summe der Länge aller Wertespanten l_i in denen in Schritt 1 und 2 von Theorem 3.1.1 ein Schnittpunkt gewählt wird. $\sum_i |u_i - v_i|$ entspricht der Summe der Wertespanten, auf denen die Wahl eines Schnittpunktes u und v trennen würde. Das Lemma folgt.

3.1.3 RRCT Instandhaltung

In diesem Abschnitt wird gezeigt das von einem RRCT $\mathcal{T}(S)$ effizient ein Punkt x gelöscht oder hinzugefügt werden kann, also die jeweiligen RRCTs $\mathcal{T}(S - \{x\})$ und $\mathcal{T}(S \cup \{x\})$ effizient erzeugt werden können.

Löschen einzelner Punkte

Soll ein Punkt u aus dem Baum \mathcal{T} gelöscht werden, so muss lediglich der Elternknoten k von u , welcher die Trennung mithilfe der u isoliert wurde darstellt, mit gelöscht werden, und der Elternknoten von k bekommt als neues Kind, dass nun verwaiste Kind von k . Siehe Bild ???

3.1.4 Theorem (Konsistenz der inneren Probabilität). *Sei ein RRCT \mathcal{T} welcher über einen Datensatz S konstruiert wurde. Wird ein Punkt $u \in S$ wie oben skizziert gelöscht, so hat der daraus resultierende Baum die gleiche Probabilität gegenüber über welche Dimensionen \mathcal{T} bei seiner Konstruktion partitioniert wird, wie ein RRCT der über $S - u$ konstruiert wurde. Parallel dazu hat ein RRCT der über $S \cup \{v\}$ mit $v \notin S$ konstruiert wird, die gleiche Probabilität wie der RRCT der aus dem hinzufügen von v zu \mathcal{T} resultiert*

Dieses natürliche Verhalten gegenüber dem hinzufügen und löschen von Punkten des RRCF Verfahrens, setzt es von vielen anderen Partitionierungsverfahren ab [6], insbesondere auch von anderen Baum konstruierenden Anomalieerkennungsverfahren Verfahren wie das Isolation Forest Verfahren, welche die über die zu partitionierende Dimension uniform-zufällig auswählen. Dies zeigt sich durch folgendes Beispiel:

Unterschiede beim Löschen eines Punktes

Beispiel mit Bild pro Fall 4+2 :)

Die so ermöglichten dynamischen Änderungen an den durch das RRCF Verfahren konstruierten Bäumen, ermöglicht unter anderem die effiziente Anomalieerkennung auf gestreamten Daten, da die neu eintreffenden Punkte in die bestehenden Bäume mit eingefügt werden können, anstatt das diese von Grund auf neu gebaut werden müssten.

3.1.5 Theorem (Die RRCT Konstruktion ist Stichproben unabhängig). *Sei S eine Stichprobe eines Datensatzes. Es kann ein RRCF über S gebildet werden, selbst wenn S dynamisch aktualisiert wird.*

Das Theorem folgt aus den bisher definierten. Theorem 3.1.2 sagt aus, dass der RRCT die in S gegebenen Abstände beibehält. Jedes auf S angewendete Stichprobenverfahren, welches die gewünschten Zusammenhänge beibehält, kann dementsprechend auch in einem RRCT abgebildet werden. Mit Theorem 3.1.4 ist der Prozess der RRCT Konstruktion unabhängig von den angewendeten Stichprobenverfahren. Soll beispielsweise eine Stichprobe von S der Größe $\rho|S|$, mit $\rho < 1$ uniform-zufällig erstellt werden, so müssen kann entweder ein RRCT über $\rho|S|$ uniform-zufällig ausgewählte Punkte von S konstruiert werden, oder es können $|S| - \rho|S|$ Punkte uniform-zufällig bestimmte Punkte aus einem bestehenden RRCT über S gelöscht werden. Beide Vorgehensweisen resultieren in den selben Probabilitäten, gegenüber der Struktur und den ausgewählten Dimensionen über die die Stichprobe partitioniert wurde, für den resultierenden Baum. Parallel dazu kann jedes weiter Stichprobenverfahren vor oder auch abhängig von der Größe des resultierenden Baumes nach der Konstruktion des RRCT angewandt werden. Es folgt:

3.1.6 Theorem. *Existiert ein Verfahren welches eine Stichprobe des Datensatzes S per Downsampling erstellt dann existiert für jede Downsampling Rate ein Algorithmus der einen RRCT über die Stichprobe erzeugt indem er Punkte aus dem RRCT über S löscht.*

Somit ist es möglich die Menge an Punkten mit der ein RRCF konstruiert wurde, nach seiner Konstruktion anzupassen. Aus Theorem 3.1.5 ergibt sich weiterhin:

3.1.7 Theorem. *Sei ein RRCT über einen Datensatz S konstruiert. Sei $u \notin S$. Da wir effizient den RRCT über $S \cup \{u\}$ konstruieren können indem wir u zu $\mathcal{T}(S)$ hinzufügen,*

können wir effizient den erwarteten Effekt von u auf die Platzierung der anderen Punkte in S bestimmen, sowie die erwartete Tiefe die u in $\mathcal{T}(S \cup \{u\})$ hat.

Diese Möglichkeit, kontrafaktische Fragen gegenüber dem Einfügen von u in $\mathcal{T}(S)$ effizient zu beantworten, eignet sich intuitiv der Anomalieerkennung. So kann entweder die erwartete Tiefe von u bestimmt werden, um über Theorem 3.1.2 den Grad der Normalität von u abzuschätzen, oder es kann der Unterschied den u zwischen $\mathcal{T}(S)$ und $\mathcal{T}(S \cup \{u\})$ erzeugt, bemessen werden. Eine konkrete Metrik dazu wird in der nächsten Sektion in Form des *Codisplacements(CoDisps)* vorgestellt.

3.2 Anomalieerkennung über RRCF

Um zu spezifizieren wie genau ein anomaler Punkt in einem *RRCF* erkannt wird, sei hier auf das Beispiel in Kapitel 2, der Menge bestehend aus schwarzen Kugeln und Würfeln, sowie einer grünen Kugel, zurückgegriffen. Hier lassen sich 2 Arten der Anomalieausprägung definieren:

1. Eine Anomalie ist einfach zu beschreiben, die grüne Kugel unterscheidet sich zwar nicht im Merkmal der Länge, aber im Merkmal der Farbe stark von den anderen Objekten der Menge. Ihre Unterscheidung von der Menge ist leicht abzugrenzen. Diese Kategorisierung ist die in Kapitel 2 verwendete.
2. Die Existenz einer Anomalie in einer Menge, macht es schwieriger diese Menge zu beschreiben. So müssen die Objekte der Menge nun nicht mehr nur nach Form, sondern auch nach Farbe differenziert werden. Der Fokus einer Beschreibung wird von einer Mehrzahl der Objekte zu einem einzigen verschoben.

Die beiden Anomalieausprägungen folgen auseinander. Das eine Anomalie über ihr hervorstechendes Merkmal einfach zu beschreiben ist, ist äquivalent dazu, dass die Beschreibung der Merkmale einer Menge einfacher wäre, würde diese Anomalie mit ihrem besonderen Merkmal beziehungsweise ihrem besonders ausgeprägtem Merkmal nicht existieren.

Der RRCF Algorithmus versucht die in Punkt 2 definierte, durch einen Punkt erzeugte Verschiebung (*Disp*) zu bestimmen. Dazu wird zuerst die Komplexität eines RRCTs definiert, um eine exakte Relation über den Effekt der im RRCT untergebrachten Punkte auf die Komplexität von diesem zu bestimmen.

3.2.1 Modellkomplexität eines RRCT

Sei jedem Zweig in einem RRCT ein Bit zugeordnet. Ein linker Zweig wird durch das Bit 0 und ein rechter Zweig durch das Bit 1 gekennzeichnet. Der Platz von jedem Punkt x in einem RRCT ist dann in diesem eindeutig durch die Folge an Bits entlang der Zweige

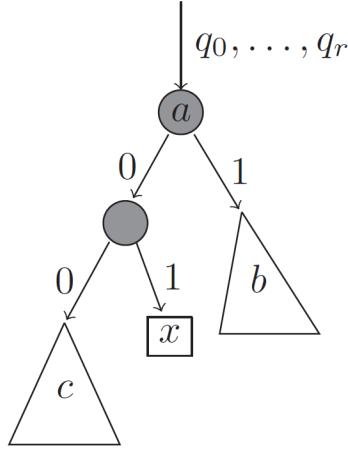


Abbildung 3.2: Ein Teilbaum T_1 über die Menge S_1 , eines RRCTs T , dessen Wurzel in T die Tiefe $r + 1$ hat. Der Knoten a stellt eine Partitionierung von S_1 in zwei Teilmengen da. q_0, \dots, q_r sind die Bits die die Position von a in T beschreiben. Quelle: [6]

von der Wurzel zu dem Punkt x , bestimmt. Siehe Abbildung 3.2, wo der Platz von x in T durch die Bitfolge $q_0, \dots, q_r, 0, 1$ definiert ist. Es bietet sich die folgende Definition 3.2.2 der Modellkomplexität eines RRCTs an:

3.2.1 Definition (Tiefe eines Punktes in x). Gegeben sei ein Satz an Punkten S und sei $T = \mathcal{T}(S)$ ein RRCT über S . Sei ein Punkt $x \in S$, mit der zugehörigen Bitfolge b . Dann sei:

$$f(x, S, T) = |b| \quad (3.3)$$

die Tiefe von x in T .

Die Tiefe eines Knotens eines Binärbaumes entspricht der Anzahl der Zweige zwischen ihm und der Wurzel. Da sich pro Zweig ein Bit in der zugeordneten Bitfolge eines Knotens eines RRCTs ergibt, folgt die Gleichung 3.3.

3.2.2 Definition (Modellkomplexität). Gegeben sei ein Satz an Punkten S und sei $T = \mathcal{T}(S)$ ein RRCT über S . Sei $f(x, S, T)$ mit $x \in S$ die Tiefe des Punktes x in T . Dann ist die Modellkomplexität von T :

$$|M(T)| = \sum_{x \in S} f(x, S, T) \quad (3.4)$$

Die definierte Modellkomplexität $|M(T)|$ entspricht somit der Summe der Länge der Bitfolgen aller Punkte in dem RRCT T . Anomalien in einem Datensatz sorgen somit für eine höhere Modellkomplexität, da diese nach 3.1.4, durch ihre Hervorstechenden Merkmale früh im RRCT Konstruktionsprozess isoliert werden, die restlichen Punkte also einen gebündelt einen weiteren Zweig herunter schickt.

3.2.2 Verschiebung der Modellkomplexität durch einen Punkt x

Parallel zu der Modellkomplexität $|M(T)|$ ist die Modellkomplexität des RRCTs $T' = \mathcal{T}(S - \{x\})$, also des RRCTs der aus der Entfernung des Punktes x aus dem RRCT T nach Theorem 3.1.4 gegeben durch:

$$|M(T')| = \sum_{x \in S - \{x\}} f(x, S - \{x\}, T) \quad (3.5)$$

Der Effekt den x auf die Modellkomplexität von T hat ist demnach:

$$|M(T)| - |M(T')| \quad (3.6)$$

Dabei ist zu beachten das der Term 3.6 nur für den Effekt gilt den x auf $|M(T)|$ hat, da nach Theorem 3.1.4 mit gegebenen T und x der durch das Entfernen von x aus T produzierte RRCT T' deterministisch bestimmt ist. Umgekehrt kann aber jeder einzelne T' aus beliebig vielen möglichen T und x hergeleitet werden, es handelt sich um eine viele-zu-einem Beziehung. Somit trifft der Term 3.6 keine Aussage über den Effekt den x in T' haben würde.

Ausgeweitet auf alle möglichen RRCTs $T = \mathcal{S}$ und allen möglichen $T = \mathcal{S} - \{\cdot\}$ ergibt sich für die erwartete Verschiebung der Modellkomplexität, die x im durchschnitt in allen T verursacht:

$$\begin{aligned} \mathbb{E}_T[|M(T)|] - \mathbb{E}_{T'}[|M(T')|] &= \sum_T \sum_{y \in S} \mathbb{P}_T[T] f(y, S, T) \\ &\quad - \sum_{T'} \sum_{y \in S - \{x\}} \mathbb{P}_{T'}[T'] f(y, S - \{x\}, T') \end{aligned} \quad (3.7)$$

$$\begin{aligned} &= \sum_T \sum_{y \in S - \{x\}} \mathbb{P}_T[T] f(y, S, T) \\ &\quad - \sum_{T'} \sum_{y \in S - \{x\}} \mathbb{P}_{T'}[T'] f(y, S - \{x\}, T') \\ &\quad + \sum_T \mathbb{P}_T[T] f(x, S, T) \end{aligned} \quad (3.8)$$

$$\begin{aligned} &= \sum_T \sum_{y \in S - \{x\}} \mathbb{P}_T[T] (f(y, S, T) - f(y, S - \{x\}, T')) \\ &\quad + \sum_T \mathbb{P}_T[T] f(x, S, T) \end{aligned} \quad (3.9)$$

Der Term 3.7 ergibt sich aus 3.2.2 und entspricht der durchschnittlichen Modellkomplexität aller über nach Definition 3.1.1 konstruierten RRCTs T und T' . In dem Term 3.8 ist die durchschnittliche Modellkomplexität des Punktes x getrennt von der des Rest

des Baumes dargestellt. Wie oben dargestellt ist nach 3.1.4 mit gegebenen T und x , das Resultat T' der Entfernung des Punktes x aus T deterministisch gegeben und es gilt somit:

$$\sum_{T'} \sum_{y \in S - \{x\}} \mathbb{P}r[T'] f(y, S - \{x\}, T') = \sum_T \sum_{y \in S - \{x\}} \mathbb{P}r[T] f(y, S - \{x\}, T') \quad (3.10)$$

Woraus der Term 3.9 folgt und sich folgende Definition gibt:

3.2.3 Definition (Verschiebung (*Displacement*) eines Punktes). Sei ein Satz an Punkten S und sei ein Punkt $x \in S$. Seien $T = \mathcal{T}(S)$ und $T' = \mathcal{T}(S - \{x\})$ RRCTs über S . Die bitweise Verschiebung die der Punkt x im RRCT T verursacht ist:

$$Disp(x, S) = \sum_T \sum_{y \in S - \{x\}} \mathbb{P}r[T] (f(y, S, T) - f(y, S - \{x\}, T')) \quad (3.11)$$

Zu bemerken gilt, dass die totale durch x durchschnittlich verursachte Vergrößerung der Modellkomplexität gegeben ist durch:

$$\mathbb{E}_T[|M(T)|] - \mathbb{E}_{T'}[|M(T')|] = Disp(x, S) + \sum_T \mathbb{P}r[T] f(x, S, T) \quad (3.12)$$

, also der Summe der Bits die zu der Bit-Repräsentation der Punkte $y \in S - \{x\}$ durch x hinzukommen, plus der Bits die x selbst darstellen. Der Fokus der Anomalieerkennung durch RRCTs liegt demnach auf der Erkennung eines Steigens der Komplexität des Datensatzes den ein Punkt des Datensatzes hervorruft, anstatt auf das Hervorstechen des Punktes an sich. Die Benutzung des Wortes Verschiebung, ergibt lässt sich über folgendes Lemma herleiten:

3.2.4 Lemma. *Die in durch einen Punkt $x \in S$ verursachte Verschiebung in einem RRCT $T = \mathcal{T}$ entspricht der Menge an Punkten, die Geschwister von x sind*

Beweis Lemma 3.2.4 Orientiert an Abbildung 3.2, ist die Bitrepräsentation jedes Punktes in c , also jedes Punktes welcher in dem Baum T ein Geschwister von x ist, gegeben durch:

$$q_0, \dots, q_r, 0, 0, \dots \quad (3.13)$$

Repräsentiert in Abbildung 3.3, welche den Teilbaum darstellt der sich aus dem Entfernen von x aus dem in 3.2 dargestellten RRCT ergibt, fällt durch das Entfernen nach 3.1.4, von x aus T ein Knoten auf dem Pfad der Wurzel von T zu den Punkten in dem Bereich c weg, womit sich für diese eine neue Bitrepräsentation gibt:

$$q_0, \dots, q_r, 0, \dots \quad (3.14)$$

Da der Pfad von der Wurzel von T , zu allen Knoten außerhalb des Bereiches c durch das Löschen von x unverändert bleibt, ergibt sich beziehend auf die Definition 3.2.1, für den Effekt von x auf die Länge der Bitrepräsentation jedes anderen Punktes in T :

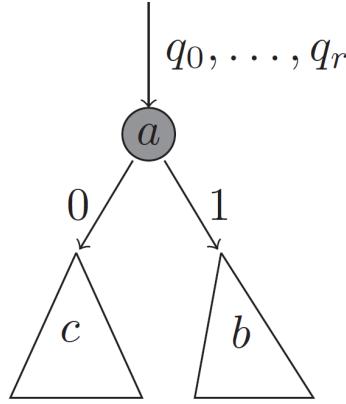


Abbildung 3.3: Ein Teilbaum T_2 über die Menge S_2 , eines RRCTs T , dessen Wurzel in T die Tiefe $r + 1$ hat. Der Knoten a stellt eine Partitionierung von S_2 in zwei Teilmengen da. q_0, \dots, q_r sind die Bits die die Position von a in T beschreiben. Quelle: [6]

$$f(y, S, T) - f(y, S - \{x\}, T') = \begin{cases} 1, & y \in c \\ 0, & \text{otherwise} \end{cases} \quad (3.15)$$

Es folgt für die Verschiebung von x in einem gegebenen Baum T :

$$Disp_T(x, S) = |c| \quad (3.16)$$

3.2.3 Codisp

Definition 3.2.3 bietet eine Möglichkeit der Anomaliedefinition. Diese ist allerdings stark anfällig gegenüber Duplikaten, wie in Sektion 2.2.2 definiert. Enthält die oben definierte Menge an Objekten 2 grüne Kugeln, so würde das Entfernen einer Kugel die Komplexität der Beschreibung der Menge nicht wesentlich vereinfachen. Ein genaueres Beispiel ergibt sich wie folgt:

Bei dem durch Abbildung 3.4 dargestellten Datensatz S , würde ein auf diesem konstruierten RRCT die Punkte o_1 und o_2 , basierend auf Theorem 3.1.2, aufgrund ihrer hohen Distanz L_1 zu allen anderen Punkten des Datensatzes wahrscheinlich schnell isolieren. $Disp(o_1, S)$ sowie $Disp(o_2, S)$ wäre, aufgrund ihrer somit folgenden hohen Anzahl an Punkten in Geschwisterknoten, ebenfalls hoch im Vergleich zu den Punkten in N_1 und N_2 . Die Punkte in O_3 würde in Erwartung, aufgrund ihrer hohen Distanz L_1 , ebenfalls schnell von allen Punkten nicht in O_3 getrennt werden. Aufgrund ihrer geringen Distanz L_1 untereinander würde die dafür verantwortliche Partitionierung, in Erwartung alle Punkte in O_3 , nach Schritt 3 der Definition 3.1.1 in eine Teilmenge partitionieren. In Erwartung ergibt sich ein RRCT wie in Abbildung 3.4. Da jedes Blatt, welches einen Punkt von O_3 enthält, eine geringe Anzahl an Blättern hat die von seinem Geschwisterknoten abstammen, ist

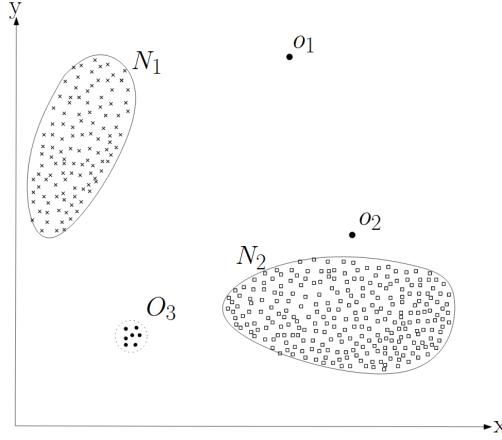


Abbildung 3.4: Ein Beispieldatensatz mit zwei Anomalien o_1 und o_2 , sowie eine Punktegruppe O_3 von 7 Anomalien. Die Gruppen N_1 und N_2 stellen die Inliner des Datensatzes da. Quelle: [4]

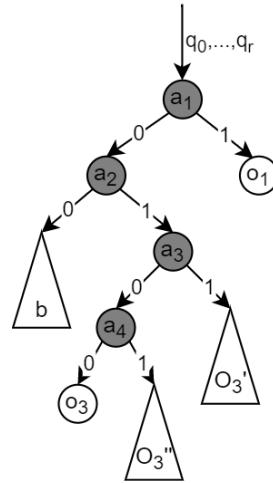


Abbildung 3.5: Ein Teilbaum, welcher

somit $Disp(o_3, S)$ für alle $o_3 \in O_3$ gering. Die Punkte O_3 können über Definition 3.2.3 nicht als Anomalie erkannt werden.

Robustheit gegenüber Duplikaten

Um über die Modellkomplexität einen anomalen Punkt $x \in S$ als solchen zu erkennen selbst wenn S Duplikate oder Beinah-Duplikate von x enthält, muss demnach das Vergleichsmodell betrachtet werden, bei dem ein Set an Punkten C , mit $x \in C$ entfernt wurden. Analog zu Term 3.9 ergibt sich für den erwarteten durchschnittlichen Unterschied in der Modellkomplexität aller RRCTs $T = \mathcal{T}(S)$ und $T'' = \mathcal{T}(S - C)$:

$$\mathbb{E}_T[|M(T)|] - \mathbb{E}_{T'}[|M(T'')|] = Disp(C, S) + \sum_T \sum_{y \in C} \mathbb{P}_r[T] f(y, S, T) \quad (3.17)$$

, wobei $Disp(C, S)$ der erwarteten Bit-Verschiebung, die die Punkte C im Durchschnitt über alle T verursachen entspricht:

$$Disp(C, S) = \sum_T \sum_{y \in S - C} \mathbb{P}_T[f(y, S, T) - f(y, S - C, T'')] \quad (3.18)$$

Die Bit-Verschiebung von x entspricht damit, basierend auf Term 3.18 und der Annahme, dass alle Punkte in C die gleiche Bit-Verschiebung zugeschrieben werden sollte, da es sich bei diesen in Erwartung um Duplikate oder Beinah-Duplikate von x handelt, $Disp(C, S)/|C|$. Dementsprechend wäre eine Methodik C zu wählen die Ermittlung des folgenden Maximums:

$$\max_{x \in C \subseteq S} Disp(C, S)/|C| \quad (3.19)$$

Dieser Methodik folgen allerdings zwei Probleme:

1. Die mögliche Anzahl an Sets von Punkten $x \in C \subseteq S$ wächst exponentiell zu S , weshalb Anomalieerkennung über Methodik 3.19 ineffizient wäre.
2. Wird S gestreamed, und der RRCT live über den Stream konstruiert, sind zum Zeitpunkt der Bewertung von x noch nicht alle Punkte von S , also nicht alle möglichen Punkte von C , sondern nur ein Sample $S' \subset S$ bekannt. Somit ist Methodik 3.19 nicht für Streaming-Daten geeignet.

Zur Lösung dieser Probleme, darf C für unterschiedliche Samples S' verschieden gewählt werden. Es ergibt sich die folgende Definition des *CollusiveDisplacements(Codisp)*, oder der Bit-Verschiebung mithilfe einer Gruppe von Punkten, eines Punktes:

3.2.5 Definition (CoDisp). Sei ein Datensatz S gegeben. Die erwartete durchschnittliche Bit-Verschiebung eines Punktes x in allen möglichen RRCTs $T = \mathcal{T}(S')$ über ein Sample $S' \subset S$ und die darüber gegebenen $T'' = \mathcal{T}(S - C)$, ist gegeben durch:

$$CoDisp(x, S, |S'|) = \mathbb{E}_{S' \subseteq S, T} \left[\max_{x \in C \subseteq S} \frac{1}{|C|} \sum_{y \in S - C} f(y, S', T) - f(y, S' - C, T'') \right] \quad (3.20)$$

Dabei kann T'' wieder aufgrund von Theorem 3.1.4 deterministisch von allen Kombinationen von T und C abgeleitet werden. Mit der nun durch *Codisp()* gegebenen, gegen Duplikate robusten Möglichkeit der Ermittlung des Effektes den ein Punkt innerhalb eines RRCTs auf die Modellkomplexität seines RRCTs hat, ergibt sich die zentrale Definition des RRCTs:

3.2.6 Definition. Die Ausreißer eines Datensatzes haben in einem über den Datensatz, oder über einem Sample über den Datensatz konstruierten RRCT in Erwartung einen hohen CoDisp()

Weiterhin gilt:

3.2.7 Lemma. Die $CoDisp(x, Z, |S|)$ kann effizient bestimmt werden

Beweis von Lemma 3.2.7 Analog zu dem Beweis von Lemma 3.2.4 ist der U

Kapitel 4

Support Vector Machine

Kapitel 5

Tests auf Niederspannungsdaten

Im Rahmen dieser Arbeit wurde die Performance von zwei Anomalieerkennungsverfahren auf dem ihr zugrundeliegendem PPC-Datensatz beurteilt. In diesem Kapitel wird nun zuerst auf die Eigenschaften des Datensatzes eingegangen, und darauf auf die Eignung der angewandten Anomalieerkennungsverfahren für diesen, sowie auf die Details ihrer jeweiligen Implementierung.

5.1 Aufmachung der Testdaten

Der PPC-Datensatz wurde im Jahr 2018 von in 17 unterschiedlichen Stellen des deutschen Niederspannungsnetz angebrachten Messstationen aufgezeichnet. Bemessen wurde dabei die absolute Spannung aller drei Stromphasen, im Regelfall in Abständen von bis zu 10 Sekunden, wobei jede Phase zeitgleich bemessen wurde. Je nach Station bilden die aufgenommenen Daten einen Zeitraum von mindestens 4 bis zu maximal 10 Monaten ab. Insgesamt gibt es x Punkte, welche sich in wie in Tabelle ?? dargestellt aufteilen.

Die absoluten Spannungswerte bewegen sich in einem Korridor von x bis y. Saisonbedingt steigt dieser bis zum Monat x, von wo an er wieder abnimmt. Tabelle ??

Die Punkte des PPC-Datensatzes wurden in der Nachbearbeitung zu bis zu 5 Anomalieklassen zugeordnet. Diese sind:

1. **Sprunganomalien:** Punkte direkt nach einer Trafostufung, also eine drei Punkte Kombination einer Messung deren jeweilige Spannungen in der jeweilig gleichen Phase entweder jeweils ungewöhnlich größer oder jeweils ungewöhnlich kleiner sind, als die 3 Spannungen der 3 Punkte Kombination der vorherigen Messung der Messstation.
2. **Zeitanomalien** Punkte direkt nach einer Messlücke, also Punkte deren Zeitpunkt weit länger als die üblichen 10 Sekunden hinter dem Zeitpunkt ihres Vorgängerpunkts liegt.

3. **Phasenanomalien:** Punkte welche sich von den jeweiligen Punkten der anderen beiden Phasen absetzen, also Punkte deren Spannung stark von der Spannung von mindestens einer Spannung der beiden zugleich aufgenommenen Punkte der beiden anderen Phasen unterscheidet.
4. **Saisonanomalien:** Punkte die mit dem saisonalen Trend der Zeitreihe brechen, also Punkte deren Spannungswerte sich stark von den Werten vorherigen Punkte unterscheiden, welche zu einer ähnlichen Uhrzeit und in der gleichen Tagesart gemessen wurden. Dabei wurde zwischen Werktagen und der Kombination aus Feier- und Wochenendtagen unterschieden.
5. **Stationsanomalien:** Punkte welche gegen den Trend des durchschnittlichen Verlaufs aller Zeitreihen verstößen, also Punkte deren Spannung sich stark von dem Durchschnitt der Spannung der Punkte aller anderen Stationen unterscheiden.

Die Anomalieklassen sind nicht exklusiv, jeder Punkt kann in mehreren Anomalieklassen sein. Dies trifft allerdings nur auf x Punkte von x anomalen Punkten zu

Während die Zeitreihen jeder Messtation ähnliche Verhaltensweisen aufweist, treten die vorhandenen Anomalien je nach Zeitreihe in jeweils unterschiedlicher Stärke und Frequenz auf. Die Daten sind jeweils punktweise gelabeled, es zeichnen sich allerdings punktübergreifende Muster für jede Anomalielasse ab:

- Aufgrund der Definition von Sprunganomalien, nach welcher jeder Punkt in einer Messung ein bestimmtes Verhalten gegenüber dem Punkt, der jeweils gleichen Phase der vorherigen Messung haben muss, damit diese Punkte als anomal gekennzeichnet sind, sind in einer Messung entweder alle Punkte eine Sprunganomalie oder keiner von ihnen. Analog dazu sind entweder alle Punkte einer Messung als Phasenanomalie gekennzeichnet oder keine.
- Ähnlich dazu treten Zeitanomalien fast ausschließlich in dreier Paaren von Punkten auf, welche eine Messung einer Station darstellen, da die ihnen zugrundeliegenden Zeitlücken, fast immer Messlücken einer Messstation entsprechen und so für jede Phase einer Station sich zeitliche Lücken bilden.
- Saison-, Phasen- und Stationsanomalien weisen ein stark geklusters Verhalten auf, wo der Großteil der ihnen zusammenhängenden Punkte direkt aufeinander folgen, wo der Spannungsverlauf einer Phase, wesentlich höher oder niedriger als erwartet ist.

5.1.1 Eignung der Daten für überwachte und unüberwachte Anomalieerkennung

Während die Verhaltensformen der Anomalieklassen sich überwachten Lernen anbieten können, wurde sich in dieser Arbeit dennoch für zwei unüberwachte Verfahren entschieden. Wie in Tabelle ?? zu sehen ist, sind die Anomalieklassen sehr gering vertreten, wodurch sich eine Knappheit an Daten mithilfe derer ein überwachtes Anomalieerkennungsverfahren die jeweiligen Anomalieklassen lernen könnte. Weiterhin entsprechen die Anomalien immer einer höheren oder niedrigeren Spannung als gewöhnlich, weshalb ein unüberwachtes Verfahren, diese über Bestimmung der jeweiligen Häufigkeit der eingegebenen Punkt als solche klassifizieren kann.

Daher wurden in dieser Arbeit zwei unüberwachte Verfahren zur Anomalieerkennung eingesetzt.

5.1.2 Benötigte Eigenschaften eines Anomalieerkennungsverfahrens

Es ergeben sich drei weitere Eigenschaften für die die gewählten Anomalieerkennungsverfahren geeignet sein müssen:

1. Aufgrund der oben beschriebenen Tendenz mancher Anomalien in Gruppen zu clustern muss ein auf dem Datensatz angewandtes Anomalieerkennungsverfahren robust gegenüber Duplikaten seien.
2. Da die Daten über die Messstationen live erfasst werden sollte das Anomalieerkennungsverfahren in der Lage sein, seinen Input als Stream zu empfangen. Weiterhin muss das Verfahren in der Lage sein sich der Änderung des durchschnittlichen Spannungswerts anzupassen, möglichst ohne das wegen dieser eigentliche Inliner als Anomalien klassifiziert werden.
3. Aufgrund der numerischen Klassifizierungskriterien der Anomalien, also dem Fehlen einer klaren Abgrenzung zwischen Inlinern und Anomalien, muss das Verfahren in der Lage sein die Grenze zwischen diesen zu approximieren. Im Falle eines unüberwachten Verfahrens ohne das es sich diesen anlernen kann.

Aufbau der Testsätze

Aufgrund des hohen Umfangs der PPC-Daten wurden die Tests auf einem Subset der RRCF Daten ausgeführt, diese waren jeweils definiert über:

- Die Station der zu testenden Zeitreihe
- Der Startzeitpunkt des zu testenden Zeitfensters
- Die zu testende Phase

- Die zu testende Anomaliekasse

Die Gesamtheit der Testsätze wurde representativ über die Daten ausgewählt. Dabei wurde ein Fokus auf Testsätze, welche anomale Abschnitte enthalten gelegt. Testsätze, welche nur aus Inlinern bestehen würden stellenweise hinzugefügt um die Klassifizierung von Inlinern als solche zu überprüfen.

5.2 Testen des RRCF Verfahrens

RRCF als unüberwachtes Anomalieerkennungsverfahren eignet sich zur Analyse des dieser Arbeit zugrunde legendem Datensatzes[3]:

- *Robust gegenüber Duplikaten*: Da der RRCF seine Anomalieeinschätzung in Form des CoDisps gibt, welches per Konzept Robust gegenüber Duplikaten ist, ist das Verfahren in der Lage auch mehrere sich nur schwach unterscheidende anomale Punkte in den Bäumen als solche zu klassifizieren.
- *Anwendbarkeit auf Streaming-Daten*: Neue Datenpunkte können in die konstruierten Bäume eingegliedert werden ohne das diese neu aufgebaut werden müssen.
- *Anpassung an Änderungen im Datensatz*: Da jeder RRCT eine endliche Anzahl an Punkten enthält, muss mit dem Einfügen von neuen Punkten in den RRCT, das Löschen von alten Punkten einhergehen. So kann das was der RRCT als Inliner klassifiziert würde, an das neuer normal angepasst werden.
- *Ausgabe in vorm einer Bewertung*: Für das RRCF Verfahren muss ein Grenzwert ermittelt werden, um den Codisp der jedem Punkt zugeordnet wird zu der binären Klassifizierung zwischen Inliner und Anomalie zu transformieren. So kann die tatsächliche Grenze der vorhandenen Anomalielabel ermittelt werden.

Ein weiterer Vorteil des RRCF Verfahrens, die effiziente Handhabung von hochdimensionalen Daten, wird hier nicht benutzt, öffnet aber weitere Alternativen zu der Handhabung des Datensatzes.

5.2.1 Implementierung der Tests

Über die Tests sollen die Parameter für den auf den PPC-Daten leistungsfähigsten RRCF gefunden werden. Zur Ermittlung der Leistungsfähigkeit wird dabei der MCC verwendet, um eine Balance zwischen den richtigen Klassifizierung von Inlinern und Anomalien zu finden, eine detailliertere Begründung dazu folgt in Sektion ??

Da in der Praxis der RRCF live auf den von den Messstationen aufgenommen Daten laufen soll, ist es Ziel der Tests diese Situation über Ausschnitte der Daten zu simulieren. Dazu wird wie folgt vorgegangen:

Ablauf der Testläufe

Jeder Testlauf testet, auf einem Testsatz die Leistungsfähigkeit einer Kombination der folgenden drei Parameter:

- **Baumgröße (ts):** Die Anzahl an Punkten die in jeden RRCT des RRCFs passen
- **Baumanzahl (nt):** Die Anzahl der RRCTs in dem konstruierten RRCF
- **Fenstergröße:** Die Größe der Fensterabschnitte, welche die Punkte aus denen die RRCTs gebaut werden ausmachen

Der Testlauf erfolgt in drei Schritten:

Schritt 1: Simulation der Praxis Es wird ein RRCF entsprechend der Parameter des Testlaufes erzeugt, um den in der Praxis bereits vorhanden, über die vorherig gestreamten Daten konstruierten RRCF zu simulieren. Dazu werden aus den letzten ts Punkten (wobei ein Punkt je nach Fenstergröße entweder ein alleinstehender Wert oder eine Reihe von Werten ist) vor dem von dem Testlauf definierten Startpunkt, nt Bäume konstruiert. Es genügt die einzelnen RRCTs nach Definition 3.1.1 zu konstruieren, anstatt die Punkte einzeln in die Bäume einzufügen, da es so nach Theorem 3.1.4 keinen Unterschied in den erwarteten Bäumen gibt.

Schritt 2: Streaming des Testsatzes Der Testsatz wird angefangen mit dem definierten Startpunkt durch jeden RRCT des RRCF gestreamed. Da die Größe jedes RRCTs nach Schritt 1 der definierten Baumgröße entspricht, muss mit jedem eingefügten Punkt ein Punkt aus dem RRCF entfernt werden. In den Testläufen wurde dabei immer der älteste Punkt, also der Punkt welcher am frühesten gemessen wurde gewählt. Mit dem Einfügen von jedem Punkt wird das CoDisp zu diesem vom RRCF berechnet und abgespeichert.

Schritt 3: Auswertung der Ergebnisse Basierend auf den generierten CoDisp Werten wird darauf der bestmögliche MCC über alle möglichen CoDisp-Grenzwerte, für ab wann ein Punkt als Anomalie klassifiziert wird ermittelt. Daraufhin wird für den ermittelten optimalen Grenzwert die Accuracy bestimmt, um eine Vergleichsmetrik mit Testabschnitten zu haben, ausschließlich Inliner enthalten.

Ergebnisse

Abweichungen, mcc vs accuracy, ergebnisse vs anomalie spezifische ergebnisse

Kapitel 6

Fazit

Anhang A

Weitere Informationen

Abbildungsverzeichnis

2.1	Ein Beispieldatensatz mit zwei Anomalien o_1 und o_2 , sowie eine Punktegruppe O_3 von 7 Anomalien. Die Gruppen N_1 und N_2 stellen die Inliner des Datensatzes da. Quelle: [4]	4
2.2	Der Einfluss von Rauschen auf einen Datensatz bestehend aus zwei Inlinergruppen und einem anomalen Punkt A . Quelle: [1]	5
3.1	Ein möglicher, nach Definition 3.1.1 konstruierter RRCF über den in Tabelle 3.1 dargestellten Datensatz S . Die erste Partition erfolgte über die dritte Dimension mit einem nach Schritt 2 zufällig bestimmten Grenzwert von 6. Da S_1 darauf mehr als einen Punkt enthielt erfolgte eine weitere Partition über die erste Dimension und einen Grenzwert von 8	11
3.2	Ein Teilbaum T_1 über die Menge S_1 , eines RRCTs T , dessen Wurzel in T die Tiefe $r + 1$ hat. Der Knoten a stellt eine Partitionierung von S_1 in zwei Teilmengen da. q_0, \dots, q_r sind die Bits die die Position von a in T beschreiben. Quelle: [6]	15
3.3	Ein Teilbaum T_2 über die Menge S_2 , eines RRCTs T , dessen Wurzel in T die Tiefe $r + 1$ hat. Der Knoten a stellt eine Partitionierung von S_2 in zwei Teilmengen da. q_0, \dots, q_r sind die Bits die die Position von a in T beschreiben. Quelle: [6]	18
3.4	Ein Beispieldatensatz mit zwei Anomalien o_1 und o_2 , sowie eine Punktegruppe O_3 von 7 Anomalien. Die Gruppen N_1 und N_2 stellen die Inliner des Datensatzes da. Quelle: [4]	19
3.5	Ein Teilbaum, welcher	19

Tabellenverzeichnis

3.1 Ein Beispiel Datensatz über 3 Dimensionen mit numerischen Werten mit $S = \{x, y, z\}$ sowie die von Definition 3.1.1 in Schritt 1 berechnete Wahrscheinlichkeit $\frac{l_i}{\sum_j l_i}$ das S in Schritt 3 über die jeweilige Dimension partitioniert wird	10
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----

Algorithmenverzeichnis

Literaturverzeichnis

- [1] AGGARWAL, CHARU C: *Outlier analysis*. In: *Data mining*. Springer, 2015.
- [2] AHMED, MOHIUDDIN, ABDUN NASER MAHMOOD und JIANKUN HU: *A survey of network anomaly detection techniques*. Journal of Network and Computer Applications, 60:19–31, 2016.
- [3] BARTOS, MATTHEW, ABHIRAM MULLAPUDI und SARA TROUTMAN: *rrcf: Implementation of the Robust Random Cut Forest algorithm for anomaly detection on streams*. Journal of Open Source Software, 4(35):1336, 2019.
- [4] CHANDOLA, VARUN, ARINDAM BANERJEE und VIPIN KUMAR: *Anomaly detection: A survey*. ACM computing surveys (CSUR), 41(3):1–58, 2009.
- [5] ERFANI, SARAH M, SUTHARSHAN RAJASEGARAR, SHANIKA KARUNASEKERA und CHRISTOPHER LECKIE: *High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning*. Pattern Recognition, 58:121–134, 2016.
- [6] GUHA, SUDIPTO, NINA MISHRA, GOURAV ROY und OKKE SCHRIJVERS: *Robust random cut forest based anomaly detection on streams*. In: *International conference on machine learning*, Seiten 2712–2721, 2016.
- [7] GUHA, SUDIPTO, NINA MISHRA, GOURAV ROY und OKKE SCHRIJVERS: *Supporting Information for: Robust random cut forest based anomaly detection on streams*. In: *International conference on machine learning*, Seiten 2712–2721, 2016.
- [8] GUPTA, MANISH, JING GAO, CHARU C AGGARWAL und JIAWEI HAN: *Outlier detection for temporal data: A survey*. IEEE Transactions on Knowledge and Data Engineering, 26(9):2250–2267, 2013.
- [9] LIU, FEI TONY, KAI MING TING und ZHI-HUA ZHOU: *Isolation-based anomaly detection*. ACM Transactions on Knowledge Discovery from Data (TKDD), 6(1):1–39, 2012.

- [10] TAN, SWEE CHUAN, KAI MING TING und TONY FEI LIU: *Fast anomaly detection for streaming data*. In: *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet sowie Zitate kenntlich gemacht habe.

Dortmund, den 5. Juli 2020

Muster Mustermann

