technische universität
dortmund

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

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 and der Vielfalt der Anwendungsgebiete. [7] Einige Beispiele für den Nutzen den die Erkennung von Anomalien darstellt 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 Zeitreihenanomalieerkennung 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 Anomalieerkenung 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 Machineërläutert. 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:

- :DDDDD

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

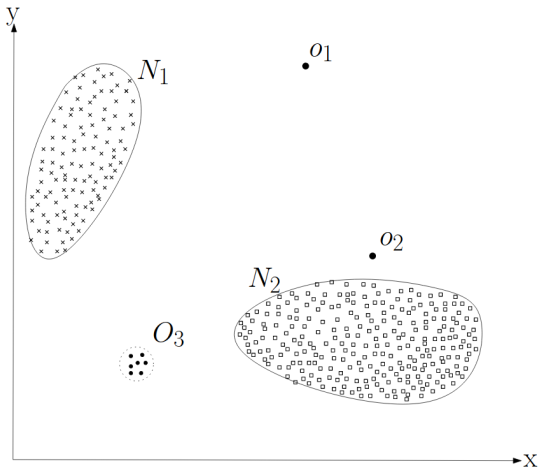


Abbildung 2.1: Ein Beispieldatensatz mit zwei Anomalien o_1 und o_2, sowie eine Punktegruppe O_3 von 7 Anomalien. Die Gruppen N_1 und N_2 stellen die Inliner des Datensatzes da. Quelle: [4]

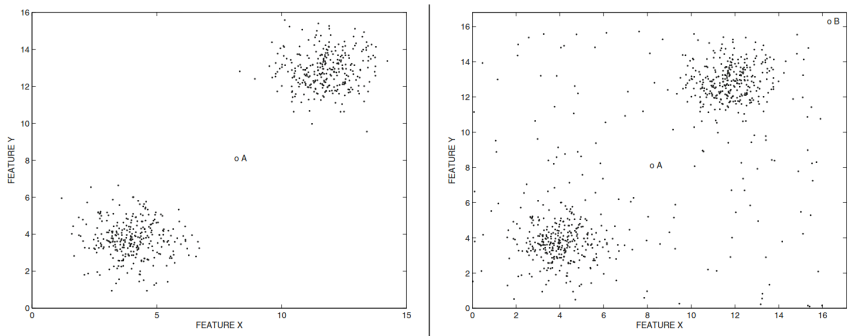
- *Kontextanomalien*: Wenn ein Datenpunkt in einem bestimmten Kontext in seinem Datensatz hervorsticht. Zum Beispiel können bei der Anomalieerkennung auf den Ausgaben einer Person, überdurchschnittlich hohe Ausgaben hohe Ausgaben an einem Feiertag normal sein, im Kontext eines Arbeitstages allerdings 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 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



(a) No noise (b) With noise

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

aber nicht zwingend selber anomal, noch invalidieren sie den Status des Ausreißers als Anomalie. [9]

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 Rauschens, muss dem Anomalieerkennungsverfahren mitgeteilt werden das Punkte mit seinen Merkmalen als anomal 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 anomal 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 Auf-

fälligkeiten miteinander verglichen werden. Anders mag es auch von Nutzen seien, ganze Zeitreihen aus einer Gruppe von Zeitreihen als anomal zu bestimmen. [7]

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 anomal 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 Anomalieklasse es gehört.

2.3.3 Robustheit

Die Robustheit eines Algorithmus beschreibt seine Stabilität gegenüber Anomalien im Trainingsdatensatzes 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 unserem Datensatz angewendeten Verfahren, der **Robust Random Cut Forest** (von hier an RRCF) wie in [6], zuerst in seinen Grundzügen beschrieben und darauf wird auf die im unterlegenen Theoreme eingegangen.

3.1 Vorteile von RRCF

RRCF wird zur Analyse des dieser Arbeit zugrunde legendem Datensatzes benutzt, da dass Verfahren eine Reihe von Vorteilen besitzt [3]:

- *Anwendbarkeit auf Streaming-Daten*: Neue Datenpunkte können in die konstruierten Bäume eingegliedert werden ohne das diese neu aufgebaut werden müssen.
- *Geeignet für hoch dimensionale Daten*: Die angewandte Baumstruktur ist sehr geeignet für das aufnehmen von hochdimensionalen Daten. Da der Algorithmus zwischen wichtigen und unwichtigen Dimensionen unterscheiden kann, wird auch der Einfluss von solchen unwichtigen Dimensionen eingeschränkt.
- *Robust gegenüber Duplikaten*: Duplikate
- *Ausgabe in vorm einer Bewertung*: Eine Bewertende Ausgabe ist nützlich, da

3.2 RRCF Theory

3.2.1 RRCF Aufbau

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

Definition 1

Ein RRCT wird über ein Datensatz S mit j Dimensionen wie folgt generiert:

Tabelle 3.1: Ein Beispiel Datensatz über 3 Dimensionen mit numerischen Werten mit $S = x, y, z$ sowie die von **Definition 1** in Schritt 1 berechnete Wahrscheinlichkeit das S über die jeweilige Dimension als nächstes in Schritt 3 geteilt wird

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

1. Wähle eine Dimension i aus den j Dimensionen. Dabei hat jede Dimension eine Wahrscheinlichkeit proportional zu $\frac{l_i}{\Sigma_j l_i}$, 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 [8], 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.
Als Beispiel würden in dem Datensatz von Tabelle 5.1 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 in Schritt 3 geteilt wird, ausgewählt werden.

In Schritt 2 wird darauf analog zum Isolation Forest Verfahren ein Trennwert X_i uniform aus der Wertespanne aller Datenpunkte der jeweiligen Dimension gewählt.

In Schritt 3 wird der Datensatz S dann in über diesen Trennwert geteilt, sodass $S1$ die Datenpunkte enthält die in Dimension i einen größeren oder gleichen Wert hatten als X_i und S_2 die verbliebenen Datenpunkte, welche in i einen kleineren Wert hatten.

3.2.2 RRCF 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.

Theorem 2

Kapitel 4

Support Vector Machine

Kapitel 5

Tests auf Niederspannungsdaten

In diesem Kapitel wird einer der beiden auf unserem Datensatz angewendeten Verfahren, der **Robust Random Cut Forest** (von hier an RRCF) wie in [6], zuerst in seinen Grundzügen beschrieben und darauf wird auf die im unterlegenen Theoreme eingegangen.

5.1 Vorteile von RRCF

RRCF wird zur Analyse des dieser Arbeit zugrunde legendem Datensatzes benutzt, da dass Verfahren eine Reihe von Vorteilen besitzt [3]:

- *Anwendbarkeit auf Streaming-Daten*: Neue Datenpunkte können in die konstruierten Bäume eingegliedert werden ohne das diese neu aufgebaut werden müssen.
- *Geeignet für hoch dimensionale Daten*: Die angewandte Baumstruktur ist sehr geeignet für das aufnehmen von hochdimensionalen Daten. Da der Algorithmus zwischen wichtigen und unwichtigen Dimensionen unterscheiden kann, wird auch der Einfluss von solchen unwichtigen Dimensionen eingeschränkt.
- *Robust gegenüber Duplikaten*: Duplikate
- *Ausgabe in vorm einer Bewertung*: Eine Bewertende Ausgabe ist nützlich, da

5.2 RRCF Theory

5.2.1 RRCF Aufbau

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

Definition 1

Ein RRCT wird über ein Datensatz S mit j Dimensionen wie folgt generiert:

Tabelle 5.1: Ein Beispiel Datensatz über 3 Dimensionen mit numerischen Werten mit $S = x, y, z$ sowie die von **Definition 1** in Schritt 1 berechnete Wahrscheinlichkeit das S über die jeweilige Dimension als nächstes in Schritt 3 geteilt wird

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

1. Wähle eine Dimension i aus den j Dimensionen. Dabei hat jede Dimension eine Wahrscheinlichkeit proportional zu $\frac{l_i}{\Sigma_j l_i}$, 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 [8], 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.
Als Beispiel würden in dem Datensatz von Tabelle 5.1 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 in Schritt 3 geteilt wird, ausgewählt werden.

In Schritt 2 wird darauf analog zum Isolation Forest Verfahren ein Trennwert X_i uniform aus der Wertespanne aller Datenpunkte der jeweiligen Dimension gewählt.

In Schritt 3 wird der Datensatz S dann in über diesen Trennwert geteilt, sodass $S1$ die Datenpunkte enthält die in Dimension i einen größeren oder gleichen Wert hatten als X_i und S_2 die verbliebenen Datenpunkte, welche in i einen kleineren Wert hatten.

5.2.2 RRCF 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.

Theorem 2

Kapitel 6

Fazit

Anhang A

Weitere Informationen

Abbildungsverzeichnis

Tabellenverzeichnis

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] 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.

[8] 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.

[9] 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 28. Mai 2020

Muster Mustermann