

Forschungsprojekt  
**Ausreißer-Erkennung in Zeitreihen  
mittels Graphen-basierter Algorithmen**

im Studiengang Angewandte Informatik  
der Fakultät Informationstechnik  
Wintersemester 2020/2021

Bahar Uzun  
764647  
Jeremy Kielman  
764097  
Marcus Erz  
762294

**Abgabedatum:** 28. Februar 2021  
**Prüferin:** Prof. Dr. rer. nat. Gabriele Gühring

---

# Kurzfassung

*todo: Kurzfassung erstellen*

**Schlagwörter:** Anomalie-Erkennung, Ausreißer-Erkennung, Netsimile, MIDAS, Random Walk, Graphen-basierte Algorithmen, Zeitreihen

# Inhaltsverzeichnis

|  |    |
|--|----|
| Abbildungsverzeichnis                        | iv |
| Tabellenverzeichnis                          | v  |
| Listings                                     | vi |
| 1 Einleitung                                 | 1  |
| 1.1 Hintergrund . . . . .                    | 1  |
| 1.2 Problemstellung . . . . .                | 2  |
| 1.3 Verwandte Arbeiten . . . . .             | 2  |
| 2 Netsimile                                  | 3  |
| 2.1 Grundlagen . . . . .                     | 3  |
| 2.1.1 Canberra Distance . . . . .            | 3  |
| 3 MIDAS                                      | 4  |
| 3.1 Grundlagen . . . . .                     | 4  |
| 3.1.1 Count-min Sketch . . . . .             | 4  |
| 4 Statische Algorithmen                      | 5  |
| 4.1 IsoMap Basierter Algorithmus . . . . .   | 5  |
| 4.1.1 IsoMap . . . . .                       | 5  |
| 4.1.2 IsoMap Basierter Algorithmus . . . . . | 6  |
| 4.1.3 Implementierung . . . . .              | 6  |
| 4.1.4 Ergebnisse . . . . .                   | 6  |
| 4.2 Perculation . . . . .                    | 7  |
| 5 Kapitelname                                | 8  |
| 5.1 Unterkapitel . . . . .                   | 8  |
| 5.1.1 Unterunterkapitel . . . . .            | 8  |
| Literaturverzeichnis                         | 10 |

# Abbildungsverzeichnis

|     |  |   |
|-----|--|---|
| 4.1 | Funktionsweiße IsoMap . . . . .  | 6 |
| 5.1 | Ego-Netzwerk des Datensatzes . . . . .   | 8 |
| 5.2 | Im Uhrzeigersinn: an erster Variablen ausgerichtet (Variante 1), an zweiter Variablen ausgerichtet (Variante 2), nicht ausgerichtet (Variante 3) und an allen Variablen ausgerichtet (Variante 4). . . . . | 9 |

# Tabellenverzeichnis

# Listings

1

# 1 Einleitung

Im Rahmen der Forschungsprojekt werden verschiedene Algorithmen zur Ausreißer-Erkennung in Graphen erforscht und getestet. Nachfolgend soll die Motivation hinter dieser Thematik erläutert werden.

## 1.1 Hintergrund

*todo: formulieren*

## 1.2 Problemstellung

**todo: Ziele definieren** Das Ziel dieser Forschungsprojekt ist es verschiedene Algorithmen anzuwenden und erste Erkenntnisse aus ihnen zu gewinnen. Dieses Hauptziel, im Zuge des ersten Semesters des Forschungsprojekts, kann wie folgt in drei Teilziele unterteilt werden:

1. Verschaffen eines Überblicks über die existierenden Algorithmen zur Erkennung von Ausreißern in Graphen
2. Die Entwicklung eines Ausreißer-Scores für die zugrundeliegenden Algorithmen
3. Erste Anwendung der verwendeten Graphen-basierten Algorithmen auf Zeitreihen

## 1.3 Verwandte Arbeiten

**todo: related work einfügen**



## 2 Netsimile

todo: In diesem Kapitel werden grundlegende Themen behandelt, die im Rahmen des Forschungsprojekts zum Verständnis der Ausreißer-Erkennung in Graphen gedient haben.

### 2.1 Grundlagen

todo: Einführung in den Algorithmus

#### 2.1.1 Canberra Distance

Einführung

todo: Stichworte sammeln

## 3 MIDAS

todo: In diesem Kapitel werden grundlegende Themen behandelt, die im Rahmen des Forschungsprojekts zum Verständnis der Ausreißer-Erkennung in Graphen gedient haben.

Erst erklären wie der MIDAS funktioniert. Und zum Laufen gebracht mit Graphen über die Zeit ENRON & DARPA. Im Anschluss auf Zeitreihendaten angewendet.

### 3.1 Grundlagen

todo: Einführung in den Algorithmus

#### 3.1.1 Count-min Sketch

Einführung

todo: Stichworte sammeln

## 4 Statische Algorithmen

**todo: Labels für die einzelnen Texte umbenennen** Es ist mit dieser Art von Algorithmen möglich Ausreißer in einer vollständigen und abgeschlossenen Zeitreihe zu identifizieren. Es werden zwei Algorithmen vorgestellt, ein auf Percolation basierender Algorithmus und ein auf IsoMap basierender Algorithmus. Beide Algorithmen wurden dazu entwickelt Ausreißer in unterschiedlichen Typen von Datensätzen zu erkennen (z.B. Videos, Bilder, Netzwerke). Voraussetzung hierfür ist lediglich, dass eine Distanz zwischen unterschiedlichen Elementen des Datensatzes berechnet werden kann (vgl. [Amil et al. 2019](#), S. 2). Im Folgenden werden die Algorithmen, hinsichtlich ihrer Fähigkeit Ausreißer in Zeitreihen zu identifizieren evaluiert.

Für beide Algorithmen gilt, dass die Zeitreihe zunächst in ein Netzwerk umgewandelt werden muss. Hierzu die Formel, welche hierzu verwendet wird:

$$a = b \tag{4.1}$$

$$D_{ij} = \left( \sum_k |v_k^i - v_k^j|^p \right)^{1/p}$$

Formel 1 gibt an, wie ein Element  $ij$  der Distanzmatrix berechnet wird. Insofern für  $p$  gleich zwei eingesetzt wird, wird zwischen den Elementen die euklidische Distanz berechnet. Ein Element der Zeitreihe kann aus mehreren Werten bestehen z.B. bei multivariaten Zeitreihen. Die berechnete Distanzmatrix bildet ein Netzwerk, dabei bilden die Zeitreihen Elemente die Knoten und die Distanzen zwischen ihnen, die Gewichte (vgl. [Amil et al. 2019](#), S. 2-3).

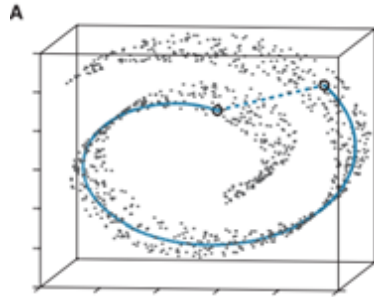
### 4.1 IsoMap Basierter Algorithmus

Der Grundgedanke hinter diesem Ansatz ist, dass Informationen über Ausreißer bei der Reduzierung der Dimensionalität mit dem IsoMap Algorithmus verloren gehen. Insofern versucht wird, die Informationen zu rekonstruieren und mit der ursprünglichen Matrix vergleicht, können große Abweichungen bei Ausreißer Elementen festgestellt werden (vgl. [Amil et al. 2019](#), S. 3).

#### 4.1.1 IsoMap

Beim IsoMap handelt es sich um einen Algorithmus zur nichtlinearen Dimensionsreduktion. Zunächst werden beim IsoMap die Nachbarn eines jeden Punktes über K-Nearest Neighbor bestimmt. Anschließend wird jeder Punkt mit den gefundenen Nachbarn verknüpft, wodurch ein neuer Körper entsteht. Daraufhin wird eine neue Distanzmatrix auf dem entstandenen Körper berechnet. Diese Matrix kann auch als geodätische Distanzmatrix bezeichnet werden und wird im weiteren Verlauf des Algorithmus benötigt. Der Zweck des Ablaufs ist es das nichtlineare

Zusammenhänge in der anschließenden Dimensionsreduktion erhalten bleiben. Die Dimensionsreduktion erfolgt anschließend über Eigenvektor ? (vgl. Tenenbaum et al. 2000, S. 3). **todo: Noch nach Seite für Quelle suchen**



**Abb. 4.1:** Funktionsweise IsoMap

#### 4.1.2 IsoMap Basierter Algorithmus

Mithilfe des IsoMap Algorithmus wurden neue Features berechnet. Im nächsten Schritt wird versucht aus diesen Features die ursprüngliche Distanzmatrix zu rekonstruieren. Nun kann die Distanzmatrix aus [Kap. 4.1.1](#) mit dieser Distanzmatrix verglichen werden. Dazu wird die Pearson Korrelation zwischen den jeweiligen Vektoren der Matrizen berechnet. Für Ausreißer wird erwartet, dass die Ähnlichkeit sehr niedrig ist, da die Informationen über sie bei der Reduktion verloren gehen (vgl. [Amil et al. 2019](#), S. 3).

#### 4.1.3 Implementierung

Da für die Implementierung des Algorithmus viele Berechnungen mit Matrizen durchgeführt werden müssen, wurde hierzu auf Python/Numpy zurückgegriffen. Für den IsoMap Algorithmus existierte eine sehr gute Implementierung in SciKitLearn, deshalb wurde auf diese zurückgegriffen. An den Algorithmus können verschieden Parameter übergeben werden, es handelt sich hierbei um dieselben Parameter, welche auch an den IsoMap Algorithmus übergeben werden können.

#### 4.1.4 Ergebnisse

uqbcqwiu

## 4.2 Percolation

todo: In diesem Kapitel werden grundlegende Themen behandelt, die im Rahmen des Forschungsprojekts zum Verständnis der Ausreißer-Erkennung in Graphen gedient haben.

# 5 Kapitelname

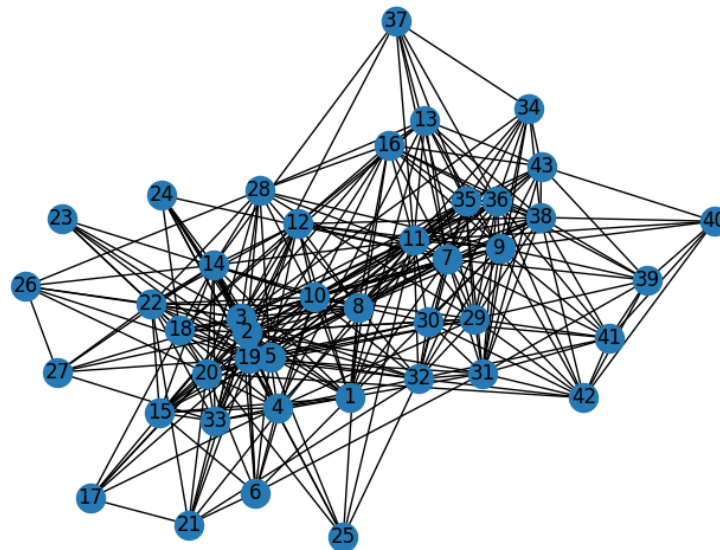
## 5.1 Unterkapitel

### 5.1.1 Unterunterkapitel

**Man** kann mit den labels im Text bezug nehmen mit [Kap. 5.1.1](#). Wenn man zitieren möchte, dann verwendet man am besten (vgl. [Mihajlovic & Petkovic 2001](#), S. 1). Ich denke die Logik wird klar, wenn man es hier betrachtet. Anführungszeichen *im Fließtext* erfolgen mittels "Wort" manchmal benötigt man das space um ein Leerzeichen zu generieren.

#### Fette Überschrift

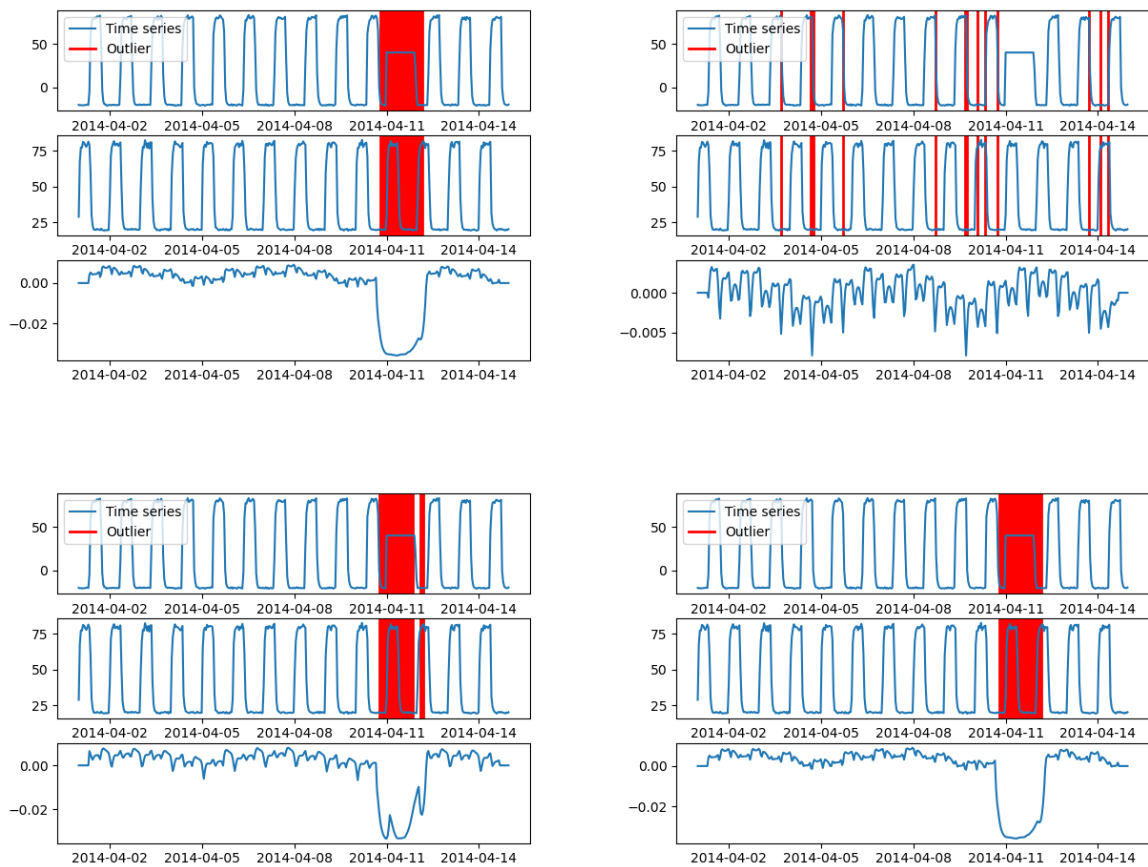
Für Bilder könnt ihr folgendes copy pasten. Die Bilder, die ihr einfügen möchtet, sollten im Ordner fig eingefügt werden. Hier sollte bei includegraphics der Name eingesetzt werden, caption steht für Bildunterschrift und das label muss individuell sein damit man darauf referenzieren kann.



**Abb. 5.1:** Ego-Netzwerk des Datensatzes

Bei mehreren Bildern nebeneinander könnt ihr folgendes verwenden:

Wenn ihr to dos vermerken wollt, nutzt hierfür: **todo: hier ein to do einfügen**



**Abb. 5.2:** Im Uhrzeigersinn: an erster Variablen ausgerichtet (Variante 1), an zweiter Variablen ausgerichtet (Variante 2), nicht ausgerichtet (Variante 3) und an allen Variablen ausgerichtet (Variante 4).

Wenn ihr etwas aufzählen wollt:

1. Verschaffen eines Überblicks über die existierenden Algorithmen zur Erkennung von Ausreißern in Graphen
2. Die Entwicklung eines Ausreißer-Scores für die zugrundeliegenden Algorithmen
3. Erste Anwendung der verwendeten Graphen-basierten Algorithmen auf Zeitreihen

# Literaturverzeichnis

Amil, P., Almeida, N. & Masoller, C. (2019), ‘Outlier mining methods based on graph structure analysis’, *Frontiers in Physics* **7**, 194.

**URL:** <https://www.frontiersin.org/article/10.3389/fphy.2019.00194>

Mihajlovic, V. & Petkovic, M. (2001), *BEISPIEL Dynamic Bayesian Networks: A State of the Art*, Vol. TR-CTIT-34 of *CTIT Technical Report Series*, University of Twente, Netherlands. Imported from CTIT.

Tenenbaum, J. B., Silva, V. d. & Langford, J. C. (2000), ‘A global geometric framework for nonlinear dimensionality reduction’, *Science* **290**(5500), 2319–2323.

**URL:** <https://science.sciencemag.org/content/290/5500/2319>