universität wien

MAGISTERARBEIT | MASTER'S THESIS

Titel | Title

Anomalieerkennung von Energieverbrauchsdaten mittels funktionaler Datenanalyse

verfasst von | submitted by

Johanna Fritz BSc (WU)

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Magistra der Sozial- und Wirtschaftswissenschaften (Mag.rer.soc.oec.)

Wien | Vienna, 2025

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 951
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Magisterstudium Statistik
Betreut von | Supervisor:	Assoz. Prof. Mag. Lukas Steinberger Bakk. BA PhD

Inhaltsverzeichnis

Danksagung

In erster Linie möchte ich mich bei der Firma nista, allen voran bei Benjamin Mörzinger, für die Unterstützung, die Idee sowie die Bereitstellung von Daten und Ressourcen bedanken. Ohne diese wäre diese Arbeit gar nicht erst möglich gewesen. Zudem bedanke ich mich bei dir für die Motivation, dein Vertrauen in mich sowie deinen fachlichen und freundschaftlichen Rat. Ein großer Dank gilt außerdem meinem Betreuer Lukas Steinberger für die Unterstützung meiner Ideen und den hilfreichen fachlichen Input.

Danke an meine Familie und alle Personen, die mich auf meinem Weg hierher begleitet und unterstützt haben, mir die Stärke und Motivation gegeben haben, dranzubleiben und immer für mich da waren.

Besonderer Dank gilt Veronika Hebenstreit für ihren unglaublichen emotionalen Beistand, ihre Geduld, ihr Verständnis und für ihre immer so treffenden weisen Ratschläge, die ich manchmal nicht hören will, aber umso mehr brauche. An Christina Gahn für ihre umfassende moralische Unterstützung, das Initiieren der Zusammenarbeit und die zahlreichen hilfreichen Ratschläge. An Xaver Stadlbauer für die Unterstützung, Motivation und den Beistand über das gesamte Studium hinweg – für literweise Kaffee, dosenweise Pesto und Stunden des gemeinsamen Rechnens und Lernens. Danke für all deine Mühe und Hingabe und dafür, dass du es immer wieder geschafft hast, mir so viel Freude am Studieren zu bereiten.

An Marina Haller und Timea Tóth für die Motivation, die bestärkenden Worte und die liebevollste, ehrlichste Fremdeinschätzung, von der ich nicht wusste, wie sehr ich sie brauche.

Zusammenfassung

In dieser Arbeit wird eine mehrstufige Methodik zur Anomalieerkennung von Smart-Meter-Daten des Wiener Start-ups *nista* vorgestellt. Diese wird in *Python* implementiert, deren Performance mittels einer Simulationsstudie analysiert und deren Eignung für den Anwendungsfall evaluiert. Die Methodik setzt sich aus einem k-Means-Clustering-Schritt und einem clusterweisen Prozess aus dem Bereich der funktionalen Datenanalyse, welcher auf den statistischen Tiefemaßen Total Variation Depth und Modified Shape Similarity basiert, zusammen. Der vorgestellte Prozess konnte in der Simulationsstudie eine sehr gute Performance hinsichtlich einer hohen True- und einer niedrigen False-Positive-Rate aufweisen. Im Vergleich mit einem simpleren Ansatz der Firma *nista* performte die vorgestellte Methode besser. Im Anwendungsfall wurden zwei verschiedene Clustering-Ansätze an den Daten erprobt — ein simpleres multivariates k-Means-Clustering, sowie ein verfeinerter Ansatz mit einer vorhergehenden Dimensionsreduktion. Die Relevanz der gewonnenen Erkenntnisse wurde von der Firma *nista* erhoben. Beide Ansätze konnten für die Firma relevante Anomalien detektieren. Der verfeinerte Ansatz stellte sich durch eine höhere Anzahl relevanter und eine niedrigere Anzahl irrelevanter Anomalien als für den Anwendungsfall als besser geeignet heraus. Abschließend wird noch auf zukünftige Anwendungsbereiche und Forschungsmöglichkeiten sowie eine Reihe von eröffneten Fragestellungen eingegangen.

Abstract

This thesis presents a multi-stage methodology for anomaly detection of smart meter data from the Viennese start-up *nista*. This is implemented in *Python*, its performance is analysed by means of a simulation study and its suitability for the use case is evaluated subsequently. The methodology consists of a k-means clustering step and a cluster-by-cluster process from the field of functional data analysis, which is based on the statistical depth measures Total Va-

riation Depth and Modified Shape Similarity. In the simulation study, the process presented was able to demonstrate very good performance in terms of a high true and low false positive rate. In comparison with a simpler approach from the company *nista*, the method presented performed better. In the use case, two different clustering approaches were tested on the data — a simpler multivariate k-means clustering, as well as a more refined approach including a previous dimension reduction. The relevance of the insights gained was assessed by the company *nista*. Both approaches were able to detect relevant anomalies for the company. The refined approach turned out to be more suitable for the application due to a higher number of relevant and a lower number of irrelevant anomalies. Finally, future areas of application and research possibilities as well as a number of open questions are discussed.

1 Einleitung

Das Wiener Startup *nista* unterstützt industrielle Fertigungsunternehmen mittels Datenanalysen Energie zu sparen. Eine geeignete Methodik zur effizienten und gezielten Erkennung von Anomalien in Energieverbrauchsdaten ist eines der Kernprobleme dieser Analysen. Derzeit werden sehr viele Anomalien erkannt, welche keine oder nur geringe Relevanz für Kund:innen haben. Folglich wird eine Vielzahl falscher Alarme ausgelöst. In dieser Arbeit werden Methoden der funktionalen Datenanalyse hinsichtlich ihrer Fähigkeit, für das Unternehmen und deren Kund:innen relevante Anomalien zu erkennen getestet und evaluiert. Ziel des Unternehmens ist es, sicherzustellen, dass Kund:innen die Warnmeldungen ernst nehmen und ein Vertrauen in die Analysen aufbauen.

Der vorgeschlagene Prozess ist ein mehrstufiges Verfahren zur Anomaliedetektion in univariaten Zeitreihendaten, welches sich aus einem Clusteringschritt, sowie robusten Methoden zur Ausreißererkennung aus dem Bereich der funktionalen Datenanalyse zusammensetzt. In der funktionalen Datenanalyse werden die Messungen der Stromverbräuche nicht als diskrete Datenpunkte, sondern als stetige Funktionen über die Zeit modelliert. Der Anomaliebegriff beschreibt in diesem Kontext Abweichungen von einzelnen Messkurven entweder im Sinne ihrer Form oder ihrer Größenordnung. Demnach wird in sogenannte “Form-” oder “Größenausreißer” unterschieden.

Um die vorgestellten Methoden in die bestehende Infrastruktur des Unternehmens bestmöglich zu integrieren, wurde der Algorithmus als Teil der Arbeit in *Python* implementiert. Ziel der Arbeit ist es, die vorgestellte Methodik hinsichtlich ihrer Eignung zur Erkennung von Anomalien in Stromverbrauchsdaten von Kund:innen von *nista* zu evaluieren.

Im ersten Teil dieser Arbeit werden sowohl die Problemstellung, Anomalieerkennung von Stromverbrauchsdaten, sowie einige grundlegende Konzepte und Hintergründe beleuchtet. Anschließend wird im Kapitel 3 die vorgestellte Methodik, Vorgehensweise und Implementation genauer erklärt und der Algorithmus und dessen Performance mittels einer Simulationsstudie evaluiert. Im Kapitel 4 wird konkret auf den Anwendungsfall und die vorhandenen Daten einge-

gangen. In erster Linie werden die Daten deskriptiv beschrieben und visuell aufbereitet, um einen Überblick über Struktur, Verteilung und etwaige Auffälligkeiten und Besonderheiten zu gewinnen. Anschließend wird die vorgestellte Methodik zur Anomalieerkennung in zwei Ausführungen, einer einfachen Variante und einer Verfeinerung dieser, an den Daten angewandt. Die somit erkannten Anomalien werden von Expert:innen des Unternehmens bewertet und deren Relevanz für den Anwendungsfall erhoben. Im Kapitel 4.4 werden die so gewonnenen Erkenntnisse diskutiert und die zwei Ansätze verglichen.

Kapitel 6 gibt abschließend einen Überblick über offene Fragestellungen, welche im Zuge der Analyse aufgekommen sind, sowie mögliche zukünftige Forschungsfelder und Anwendungsbereiche.

2 Theorie

2.1 Anomalieerkennung von Energieverbrauchsdaten

Die Analyse von Zeitreihendaten und die Erkennung von Anomalien in diesen ist für eine Vielzahl an Anwendungsbereichen relevant. Energieverbrauchsdaten sind einer davon. Einen Überblick über Anomalieerkennung von Zeitreihendaten geben Schmidl et al [11]. Sie evaluieren Methoden, indem sie bestehende Techniken nach Datentyp und Lerntyp kategorisieren und diese vergleichen. Methoden können einerseits nach der Dimension der Daten in univariate und multivariate Zeitreihen, sowie nach Lerntyp in supervised, unsupervised und semi-supervised Learning oder auch nach Methodenfamilie unterteilt werden. [11]

Unsupervised Learning Algorithmen versuchen die Anomalien ohne Vorkenntnisse über die Daten (es gibt keinen expliziten Trainingsschritt) zu erkennen, indem angenommen wird, dass sich diese in deren Form, Verhalten oder Verteilung von den übrigen Daten unterscheiden bzw. seltener vorkommen. Bei supervised Learning muss bereits bekannt sein, ob ein Datenpunkt normal oder abnormal ist, damit der Algorithmus in einem Trainingsschritt dieses Verhalten modellieren kann, um anschließend neue Daten zuzuordnen. [11]

Speziell zur Analyse von Energieverbrauch über längere Zeitintervalle geben Himeur et al [6] eine Übersicht der bestehenden Methoden zur Anomalieerkennung, deren besondere Charakteristika sowie derzeitiger Trends und Aussichten in diesem Bereich. Die aufgezeigten Methoden werden ebenfalls in unsupervised und supervised Learning unterteilt. Hinzu kommen noch die Kategorien Ensemble-Methoden (Boosting, Bagging), Feature Extraction (Zeitreihenanalyse, Distanz basierte Methoden, Dichte basierte Methoden, Graphen basierte Methoden), hybrides Lernen und „andere Techniken“ (Visualisierung, compressive sensing).

Der Anwendungsfall von Energieverbrauchsdaten weist einige domänenspezifische Herausforderungen, Einschränkungen und spezifische Problemstellungen auf. Eine häufig auftretende (auch im Anwendungsfall dieser Arbeit vorhandene) ist das Fehlen von gelabelten Datensätzen bzw. das fehlende Wissen über normale und abnormale Verbrauchsmuster. Des Weiteren

sind Datensätze in diesem Bereich zumeist sehr unbalanciert, d.h. die Anomalien bilden nur einen sehr kleinen Prozentsatz der gesamten Daten. Im Spannungsfeld dazu steht eine konträre Hürde, nämlich dass “abnormales” bzw. “verschwenderisches” Verhalten von Nutzer:innen auch sehr häufig auftreten kann, wenn z.B. eine Kühlschranktür sehr häufig schlecht geschlossen wird oder ein kaum benutztes Gerät ununterbrochen eingeschaltet ist. Dies ist vor allem für Methoden des unsupervised Learnings bzw. Algorithmen, die darauf basieren, dass abnormales Verhalten nur in seltenen Fällen auftritt, relevant, da dieses Phänomen deren Effizienz negativ beeinflussen kann.

Eine weitere Problemstellung in diesem Bereich ist die unklare bzw. fehlende Definition eines passenden Anomaliebegriffs. Die klassische Definition einer Anomalie ist möglicherweise nicht ausreichend, um die möglichen Fälle des Anwendungsbereiches abzudecken bzw. bedarf es zur Erklärung von Auffälligkeiten oft zusätzlicher Informationen, wie z.B. Wetterdaten, Nutzungsmuster oder Betriebsinformationen. Auch die bloße Unterteilung in „normales“ und „abnormales“ Verhalten, wie sie von vielen Algorithmen gehandhabt wird, stellt eine Schwierigkeit dar, da unterschiedliche Verhaltensmuster in Realität differenzierter auftreten.

Zusätzlich zum Fehlen von gelabelten Daten ist die Ressourcenintensivität zur Erstellung dieser, d.h. dass es für Unternehmen zeitintensiv und demnach auch kostspielig ist, vorhandene Daten zu labeln, eine weitere Hürde. Auch das Generieren von simulierten Daten kann schwierig sein, da Anomalien in diesem Bereich sehr differenziert auftreten und daher schwer bis kaum reproduzierbar sind.

Stromdaten und Nutzungsmuster verändern sich mit der Zeit, was dazu führt, dass die einzelnen Beobachtungen nicht unabhängig und identisch verteilt sind. Dieses Problem wird im Bereich des maschinellen Lernens als „Concept-Drift“ bezeichnet und kann dazu führen, dass Methoden für eine gewisse Zeit funktionieren und danach obsolet werden.

Weitere Problemstellungen sind auch das Fehlen von Plattformen zum Vergleich und zur Reproduktion von empirischen Ergebnissen, sowie das Fehlen von einheitlichen Metriken zum Vergleich der Performance der unterschiedlichen Methoden. [6]

Um das Problem des fehlenden Wissens über normales und abnormales Verhalten im Anwendungsfall zu adressieren, wurden die erkannten Anomalien von Expert:innen aus der Branche im Kontext der einzelnen Cluster präsentiert und diese anschließend bewertet. Da so nur die erkannten potenziellen Anomalien im Datensatz betrachtet werden, ist dieser Ansatz weniger zeit- und ressourcenintensiv als ein manuelles Labeln der gesamten Daten. Weiters wurde den Expert:innen die Möglichkeit gegeben, die Auffälligkeiten zu kommentieren, was eine differen-

ziertere Betrachtung als bloß "normales" und "abnormales" Verhalten ermöglicht. Dem kommt hinzu, dass Expert:innen über Hintergrundwissen aus der Branche sowie Betriebsinformationen des Unternehmens verfügen und dieses in ihre Beurteilung mit einfließen lassen können. Da nicht der gesamte Datensatz gelabelt wird, werden die vorgestellten Methoden zusätzlich noch in einer Simulationsstudie auf deren Performance getestet.

2.2 Funktionale Datenanalyse

Der Begriff "Funktionale Datenanalyse"beschreibt die statistische Analyse von Daten, welche in Form von Funktionen, Kurven oder kontinuierlichen Prozessen auftreten, wie z.B. Temperaturmessungen im Laufe der Zeit oder, wie im Anwendungsfall dieser Arbeit, die Messungen von Stromzählern. Die Beobachtungen werden somit, anstatt als diskrete Datenpunkte, als reelle Funktionen über ein Intervall modelliert. Jede Beobachtung ist somit eine reelle Funktion $X_i\left(t\right),\ i = 1, .., n; t \in\ T$ wobei T ein Intervall in $\mathbb{R}$ darstellt. Funktionale Daten sind an sich unendlichdimensional, was aufgrund der Menge an Informationen, die diese beinhalten, sowohl Herausforderungen für Theorie und Berechnung als auch spannende Möglichkeiten für Forschung und Analyse dieser mit sich bringt.

In der Praxis werden diese Funktionen häufig als Realisationen eines eindimensionalen stochastischen Prozesses betrachtet. Dieser Prozess kann jedoch nicht unmittelbar beobachtet werden, da die Daten in der Realität diskret über die Zeit gesammelt werden. Eine allgemeine Annahme ist, dass alle n Datenpunkte auf demselben dichten Zeitraster von geordneten Zeitpunkten $t_1, \ldots, t_p$ aufgezeichnet werden. Bei Aufzeichnungen durch ein Messgerät haben diese Zeitpunkte in der Regel gleiche Abstände, d.h. es gilt $t_j - t_{j-1} = t_{j+1} - t_j$ für alle j.

Für die Analyse funktionaler Daten existieren eine Vielzahl an statistischen Methoden, wobei nicht- und semiparametrische Methoden die vorherrschenden in der Literatur sind. Dies folgt aus der Komplexität dieser hochdimensionalen Daten und der dadurch benötigten Flexibilität. Beispiele für solche Methoden sind die Functional Principal Component Analysis (fPCA), funktionale Regressionsmodelle sowie funktionales Clustering und Classification.[14]

2.2.1 Anomaliebegriff in der funktionalen Datenanalyse

Eine klare und eindeutige Definition einer Anomalie ist eine fortwährende Problematik im Bereich der funktionalen Datenanalyse. In erster Linie kann grob in Größen- und Form-Ausreißer

unterschieden werden. Größenausreißer unterscheiden sich dadurch von den übrigen Daten, dass sie Werte in einer stark unterschiedlichen Größenordnung annehmen. Für Formausreißer ist die Definition komplexer, da eine Vielzahl unterschiedlicher Arten von Formausreißern auftreten können. Generell wird von einem Formausreißer gesprochen, wenn eine Beobachtung ein Muster aufweist, welches sich in irgendeiner Art und Weise von den übrigen unterscheidet, seien es Oszillationen mit unterschiedlichen Frequenzen oder Phasen bis hin zu einem plötzlichen, abrupten Sprung. Diese beiden Arten von Ausreißern schließen sich nicht gegenseitig aus: ein Ausreißer kann sowohl in dessen Größenordnung als auch in der Form andersartig sein (vgl. [7], S. 446). Während Größenausreißer oft leicht, auch visuell, erkannt werden können, ist die Erkennung von Formausreißern etwas komplizierter, da diese in der Vielzahl an Beobachtungen oft nicht auffallen und auch im Sinne der Euklidischen Distanz nicht zwingend stark abweichen müssen (vgl. [1], S. 604).

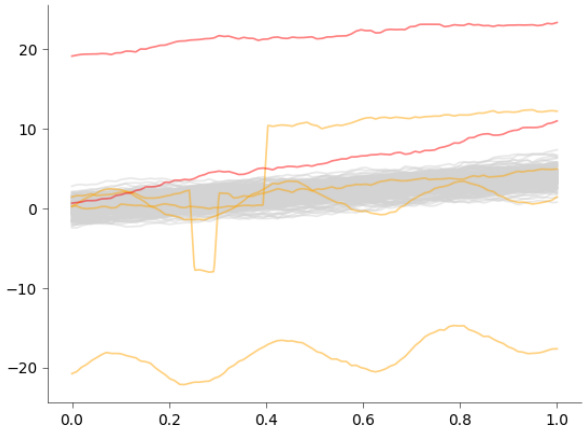

Abbildung 2.1: Beispiel für Form- und Größenausreißer

Abbildung 2.1 illustriert unterschiedliche simulierte Beispiele für Form- Größenausreißer (siehe Kapitel 3.3). Die roten Kurven sind Beispiele für Größenausreißer, die orangen für Formausreißer, wobei die unterste (sinusförmige) Kurve einen Ausreißer, welcher sich sowohl in Größe, als auch in Form unterscheidet, darstellt.

2.2.2 Tiefenmaße und Ausreißergrad

Für robuste Schätzungen und um zentrale Datenpunkte von Ausreißern zu unterscheiden, kommen in der funktionalen Datenanalyse die Konzepte von Datentiefe bzw. Tiefenmaßen (Data Depth) und Ausreißergrad (Outlyingness) zur Anwendung.

Tiefenmaße sind ein Werkzeug aus der nichtparametrischen multivariaten Inferenzstatistik und quantifizieren, wie zentral ein Datenpunkt in einer Verteilung liegt und messen die „Tiefe“ einer Beobachtung im Bezug zum gesamten Datensatz. Je zentraler ein Datenpunkt am Zentrum der Verteilung liegt, umso größer das Tiefenmaß. Somit hat der Median, als zentralster Datenpunkt, den höchsten Wert. Sei F_X eine Verteilung in $\mathbb{R}^d$, dann ordnet ein Tiefenmaß jedem aus dieser Verteilung stammenden $\mathbf{X}$ einen reellen nicht-negativen beschränkten Wert $D(\mathbf{X}, F_X)$ zu. [9]

Der Ausreißergrad verhält sich invers zum Tiefenmaß: Er beschreibt, wie sehr sich ein Punkt von der Struktur der übrigen Daten unterscheidet. Somit haben die zentralsten Punkte die niedrigsten Werte und jene mit der größten Abweichung die höchsten. [3]

Total Variation Depth und Modified Shape Similarity

Total Variation Depth (TVD), oder auch Gesamtvarianztiefe, und Modified Shape Similarity (MSS) sind von Huang und Sun [7] eingeführte Tiefenmaße zur Analyse funktionaler Daten. Die *TVD* definiert sich als Integral eines univariaten Tiefenmaßes und hat die nützliche Eigenschaft, dass sie sich in eine Form- und eine Größenkomponente zerlegen lässt. Die *MSS* definiert sich durch ebendiese Formkomponente. Diese Zerlegung ermöglicht die gezielte Erkennung von Abweichungen und Ausreißern in den Daten nach deren Größen- und Formaspekt. In weiterer Folge wird ein Überblick über die theoretischen Aspekte dieser Maße gegeben. Im Abschnitt 3.2 wird der Prozess zur Erkennung von Ausreißern definiert und die empirische Schätzung von *TVD* und *MSS* genauer beschrieben.

Sei $X(t)$ ein reellwertiger stochastischer Prozess auf dem Interval T in $\mathbb{R}$ mit der Verteilung F_X. Dann definiert sich die TVD einer Funktion f bezüglich F_X in einem ersten Schritt punktweise als:

$$D_f(t) = var\{R_f(t)\} \tag{2.1}$$

$$R_f(t) = \mathbf{1}\{X(t) \leq f(t)\} \tag{2.2}$$

wobei $\mathbf{1}$ die Indikatorfunktion ist. Die funktionale *TVD* definiert sich in weiterer Folge als

$$TVD(f) = \frac{1}{|T|} \int_T D_f(t)dt \tag{2.3}$$

$|T|$ steht für die Länge des Intervalls T, wobei hier auch alternativ eine Gewichtsfunktion verwendet werden kann (siehe [7], S. 447).

Seien t, s zwei Zeitpunkte aus dem Intervall T, wobei $s = t - \Delta$, folgt aus dem Gesetz der totalen Varianz die Zerlegung der *TVD*:

$$D_f(t) = var\{R_f(t)\} \tag{2.4}$$
$$= var[\mathbb{E}\{R_f(t)|R_f(s)\}] + \mathbb{E}[var\{R_f(t)|R_f(s)\}] \tag{2.5}$$

Huang und Sun [7] definieren $var[\mathbb{E}\{R_f(t)|R_f(s)\}]$, als den Teil der Varianz, welcher durch $R_f(s)$ erklärt wird, als die Formkomponente der Kurve und $\mathbb{E}[var\{R_f(t)|R_f(s)\}]$, als jenen Teil, welcher unabhängig von $R_f(s)$ ist, als die Größenkomponente. Macht die Formkomponente einen Großteil der TVD einer Kurve aus, so weist dies darauf hin, dass diese eine ähnliche Form wie die Mehrheit der Kurven in der Stichprobe hat (vgl. [7], S. 447). Die *Shape-Similarity* (*SS*) ergibt sich in weiterer Folge aus dem Verhältnis der Formkomponente einer Kurve zur gesamten Varianztiefe.

$$SS(f;\Delta) = \int_T v(t;\Delta) S_f(t;\Delta) dt \tag{2.6}$$
$$S_f(t;\Delta) = \begin{cases} var[\mathbb{E}\{R_f(t)|R_f(t-\Delta)\}]/D_f(t), & D_f(t) \neq 0 \\ 1, & D_f(t) = 0 \end{cases} \tag{2.7}$$

$R_f(t)$ ist wie oben definiert eine Indikatorfunktion, d.h. wenn $R_f(t) = \mathbf{1}\{X(t) \leq f(t)\} = 1$ spiegelt dies nicht wider wieviel größer der Wert von $f(t)$ ist. $S_f(t;\Delta)$ verhält sich demnach unabhängig von Unterschieden in der Größenordnung der Kurven. Dies motiviert die Wahl der Gewichtsfunktion als $v_f(t;\Delta) = |f(t) - f(t-\Delta)| / \int_T |f(t) - f(t-\Delta)|$, also die normalisierten Änderungen in $f(t)$. Mit dieser Wahl von $v(t;\Delta)$ wird mehr Gewicht auf Zeitintervalle mit größeren Veränderungen in der Größenordnung der Werte gelegt. Die Definition von $S_f(t;\Delta)$ impliziert, dass $0 \leq S_f(t;\Delta) \leq 1$. Im Anwendungsfall würde ein Wert von 0 bedeuten, dass keine Funktion einander in deren Form gleicht und $var[\mathbb{E}\{R_f(t)|R_f(t-\Delta)\}] = var[\mathbb{E}\{R_f(t)\}] = 0$. Eine Shape Similarity von 1 würde implizieren, dass $var[\mathbb{E}\{R_f(t)|R_f(t-\Delta)\}] = var\{R_f(t)\}$. In diesem Fall wäre $X(t)$ ein perfekt korrelierter Prozess und alle Kurven hätten die gleiche Form (vgl. [7], S. 448).

In Definition 2.6 sind kleinere Werte der SS mit einem größeren Ausreißergrad assoziiert. Es kann jedoch passieren, dass für Ausreißer $(f(t), f(t-\Delta))$, welche einen kleinen Wert in der Formkomponente aufweisen, der Wert von $D_f(t)$ im Nenner zu klein ist und somit $SS(f;\Delta)$ nicht klein genug ist, um diesen zu erkennen. Um diesen Formausreißergrad besser zu reflektieren, verschieben Huang und Sun [7] die einzelnen Funktionenpaare $(f(t), f(t-\Delta))$ hin zum Kurvenmedian, sodass $(\tilde{f}(t), \tilde{f}(t-\Delta)) = (f(t), f(t-\Delta)) - [f(t) - median(X(t))]$.

Für jedes Paar $(t-\Delta, t)$ definiert sich $\tilde{f}$ als

$$\tilde{f}(s;\Delta) = \begin{cases} median\{X(s)\}, & s = t \\ f(s) - f(s+\Delta) + median\{X(s+\Delta)\}, & s = t-\Delta \end{cases} \tag{2.8}$$

und $S_{\tilde{f}}(t;\Delta) = var(\mathbb{E}[R_{\tilde{f}}(t)|R_{\tilde{f}}(t-\Delta)])/D_{\tilde{f}}(t)$.

Die *Modified-Shape-Similarity* (*MSS*) definiert sich als

$$MSS(f;\Delta) = \int_T v_f(t;\Delta) S_{\tilde{f}}(t;\Delta) dt \tag{2.9}$$

(vgl. [7], S 446-448)

2.2.3 Funktionale Boxplots

Der funktionale Boxplot ist ein Werkzeug zur Visualisierung von funktionalen Daten, insbesondere deren Zentralität, und kann auch zur Erkennung von funktionalen Ausreißern dienen. Basierend auf einem funktionalen statistischen Tiefenmaß, visualisiert er deskriptive Statistiken des funktionalen Datensatzes: die Mediankurve, die 50 % Central-Region und die maximale Hülle ohne Ausreißer. Ein Vorteil dieser Methode gegenüber anderen in der Literatur erwähnten Visualisierungen zur Ausreißererkennung von funktionalen Daten, wie z.B. dem Functional-Bag-Plot oder dem Functional-Highest-Density-Region-Plot, ist, dass hier die Daten direkt im Funktionenraum abgebildet werden und keine vorhergehende Hauptkomponentenanalyse erforderlich ist. Im Setting eines klassischen Boxplots ist der erste Schritt die Berechnung einer Rang- bzw. Ordnungsstatistik. Im funktionalen Setting werden die Daten mittels Tiefenmaßen, wie z.B. der Total Variation Depth [7], nach ihrer Zentralität geordnet. [12]

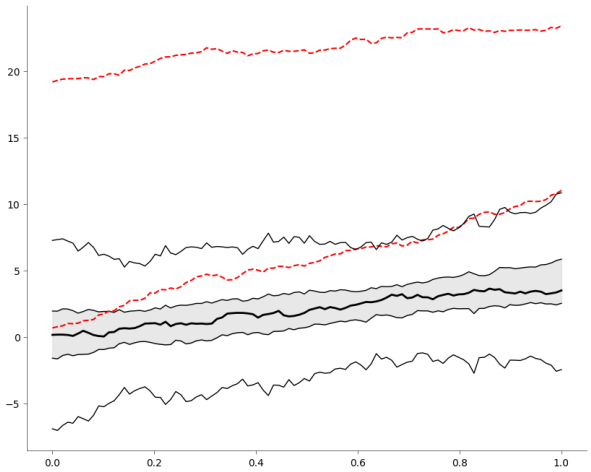

Abbildung 2.2: Beispiel für einen funktionalen Boxplot

In einem klassischen Boxplot repräsentiert die Box die mittleren 50 % der Daten — ein Ansatz, welcher sich im Sinne einer "Central-Region" auf funktionale Daten erweitern lässt. Dieser Bereich oder auch "Band" begrenzt den 50 %-Anteil der zentralsten Kurven der Stichprobe.

Sei X_i eine Stichprobenkurve aus einem funktionalen Datensatz. Dann ist $X_{[i]}$ jene Kurve mit dem i-größten Wert eines funktionalen Tiefenmaßes – in diesem Fall der Total Variation Depth. Somit ist $X_{[1]}$ die zentralste Kurve bzw. die Mediankurve und $X_{[n]}$ die am meisten vom Zentrum abweichende Kurve.

Seien nun

$$Q_1(t) = \min_{r=1,\dots,[n/2]} X_{[r]}(t) \tag{2.10}$$

$$Q_3(t) = \max_{r=1,\dots,[n/2]} X_{[r]}(t) \tag{2.11}$$

$$IQR(t) = Q_3(t) - Q_1(t) \tag{2.12}$$

wobei $[n/2]$ die kleinste natürliche Zahl größer oder gleich $n/2$ ist. Dann definiert sich die 50 % -"Central-Region" als:

$$C_{0.5}(t) := [Q_1(t), Q_3(t)]. \tag{2.13}$$

Die Grenze dieser Region ist demnach das funktionale Pendant zur Umrandung der Box eines klassischen Boxplots und die Fläche repräsentiert den Interquartilsabstand. Die maximale Hül-

le, das Pendant zu den Antennen eines klassischen Boxplots, wird bestimmt, indem man die Fläche der "Central-Region" um einen Faktor F (meist 1.5) mal ihrer Spannweite erweitert. Sie definiert sich demnach als:

$$C_{max}(t) := [Q_1(t) - F \cdot IQR(t), Q_3(t) + F \cdot IQR(t)]. \tag{2.14}$$

Eine Kurve X_i ist ein Ausreißer im Sinne eines funktionalen Boxplots, wenn:

$$\exists t \in T \text{ so dass } X_i(t) < Q_1(t) - F \cdot IQR(t) \text{ oder } X_i(t) > Q_3(t) + F \cdot IQR(t). \tag{2.15}$$

Abbildung 2.2 zeigt ein illustratives Beispiel für einen funktionalen Boxplot, basierend auf dem funktionalen Tiefenmaß TVD mit Faktor $F = 1.5$. Die schwarze Linie in der Mitte repräsentiert die Mediankurve. Die hellgraue Fläche die "Central-Region", die schwarzen Linien zeigen die maximale Hülle und rot strichliert sind die nach der Regel in Formel 2.15 erkannten Ausreißer gekennzeichnet. [12]

2.2.4 Clustering von funktionalen Daten

Um funktionale Daten zu Clustern, werden typischerweise Konzepte, welche bei vektorwertigen multivariaten Daten Anwendung finden, auf funktionale Daten erweitert. Die häufigsten Methoden darunter sind hierarchisches und k-Means-Clustering. Bei funktionalen Daten ist die Anwendung von k-Means-Algorithmen eine weit verbreitete Methode (vgl. [14], S. 276). In weiterer Folge wird nun konkret auf diesen Ansatz eingegangen.

Für eine Stichprobe von funktionalen Daten $X_i\,(t)\,; i = 1, \ldots, n$ wird versucht, eine Menge an Cluster-Zentren $\mu^c; c = 1, \ldots, k$ zu finden, unter der Annahme, dass k Cluster existieren, welche die Summe der quadrierten Abstände zwischen den X_i und den zugehörigen Cluster-Zentren, welche mit deren Cluster-Labels $\{C_i; i = 1, \ldots n\}$ verbunden sind, minimiert. Im funktionalen Clustering wird die Mittelwertfunktion eines Clusters als dessen Zentrum betrachtet. Als Distanzmaß wird eine geeignete funktionale Distanz d gewählt, in der Praxis oft die L^2 Norm. Demnach werden n Beobachtungen in k Gruppen aufgeteilt, sodass

$$\sum_{c=1}^{k} \sum_{i:C_i=c} d^2\left(X_i, \mu_n^c\right) \tag{2.16}$$

über alle möglichen Mengen von Funktionen $\{\mu_n^c : c = 1, \ldots, k\}$ minimiert wird, wobei

$\mu_n^c(t) = \sum_{i=1}^n X_i(t)\mathbf{1}_{\{C_i=c\}}/N_c$ und $N_c = \sum_{i=1}^n \mathbf{1}_{\{C_i=c\}}$ ist. Die in der Praxis diskret aufgezeichneten funktionalen Daten machen das vorliegende Optimierungsproblem analog zu jenem im multivariaten Kontext. Gängige Ansätze zur Lösung dieses Problems im funktionalen Kontext bestehen darin, die hochdimensionalen Daten in einem niedrigdimensionalen Raum abzubilden. Ein Ansatz hier wäre z.B. die Daten mittels Basisfunktionen, z.B. B-Splines oder Fourier-Basisfunktionen, zu approximieren und den k-Means-Algorithmus auf die Komponenten der entsprechenden Basisdarstellung anzuwenden. Eine weitere Herangehensweise wäre, die Dimensionen durch eine funktionale Hauptkomponentenanalyse (siehe 2.2.5) zu reduzieren und den k-Means-Algorithmus auf die fPC-Scores anzuwenden (vgl. [14], S. 276).

2.2.5 Funktionale Hauptkomponentenanalyse

Die funktionale Hauptkomponentenanalyse bzw. Functional Principal Component Analysis (fPCA) ist eine zentrale Technik in der funktionalen Datenanalyse. Sie dient zur Dimensionsreduktion und kann helfen, charakteristische Merkmale in den Daten zu erkennen. In einem ersten Schritt wird hier das ursprüngliche Konzept der Principal Component Analysis (PCA) aus der multivariaten Datenanalyse eingeführt und dieses anschließend für den funktionalen Kontext erweitert.

PCA in der multivariaten Datenanalyse

Der zugrundeliegende Ansatz einer PCA ist es, Linearkombinationen der Variablenwerte zu betrachten:

$$z_i = \sum_{j=1}^{p} \beta_j x_{ij} \ , i = 1, \ldots, N \tag{2.17}$$

β_j sind hier Gewichtskoeffizienten und x_{ij} die beobachteten Werte. Bei einer PCA wird nun schrittweise eine Menge an Gewichtskoeffizienten gesucht, welche die Varianz der z_i maximieren:

1. Finde einen Vektor $\beta_1 = (\beta_{11}, \ldots, \beta_{p1})$ für welchen die Linearkombination $z_{i1} = \sum_j \beta_{j1} x_{ij}$ das größtmögliche quadratische Mittel $N^{-1} \sum_i z_{i1}^2$ aufweist, unter der Bedingung, dass $\sum_j \beta_{j1}^2 = \|\beta_1\|^2 = 1$ gilt.
2. Wiederhole diesen Schritt, wenn möglich bis zur Obergrenze, der Anzahl der Variablen p. Für den m-ten Schritt wird nun ein Vektor β_m mit Komponenten β_{jm} und neue Werte

$z_{im} = \langle \beta_m, x_i \rangle$, welche das größte quadratische Mittel unter der Bedingung $\|\beta_m\| = 1$ haben, sowie die $m-1$ weiteren Bedingungen: $\sum_j \beta_{jk}\beta_{jm} = \langle \beta_k, \beta_m \rangle = 0 \ , k < m$, also dass jeder neue Gewichtsvektor orthogonal zum Vorhergehenden ist.

Die Motivation hinter dem ersten Schritt ist es, die stärkste Ausprägung der Varianz in den Daten zu identifizieren. Die Orthogonalitätsbedingung verhindert Redundanzen in der Darstellung. In jedem Schritt wird die Varianz, gemessen im quadratischen Mittel der z_i, abnehmen. Die Werte dieser Linearkombinationen z_{im} werden Principal-Component-Scores (PC-Scores) genannt und können zur Beschreibung von charakteristischen Eigenschaften in den Daten dienen. (vgl. [10], S. 86-87)

PCA in der funktionalen Datenanalyse

Im funktionalen Kontext werden die Variablen nicht mehr als Vektoren, sondern als Funktionen $f_i(t)$ betrachtet. Die Gewichte werden zu Gewichtsfunktionen $\beta_i(t)$ und die Summen zu Integralen über das Intervall T. Die funktionalen PC-Scores definieren sich wie folgt:

$$z_i = \int_T \beta(t) f_i(t) dt = \langle \beta, f_i \rangle \tag{2.18}$$

Im ersten Schritt einer fPCA wird eine Gewichtsfunktion β_1 gewählt, welche $N^{-1}\sum_i z_{i1}^2 = N^{-1}\sum_i \langle \beta_1, f_i \rangle^2$ unter der Einheitsnorm-Bedingung $\|\beta_1\|^2 = \int \beta_1(t)^2 dt = 1$ maximiert. Analog zur multivariaten PCA gilt für die weiteren Gewichtsfunktionen β_m die Orthogonalitätsbedingungen $\langle \beta_k, \beta_m \rangle = 0 \ , k < m$ (vgl. [10], S. 87-88).

Dieses funktionale Optimierungsproblem kann praktisch über mehrere Ansätze gelöst werden. Gängige Methoden sind die Diskretisierung der Funktionen durch Unterteilung des Intervalls T in ein gleichmäßiges Raster, oder die Approximation durch Linearkombinationen von Basisfunktionen, wie z.B. Fourier-Basisfunktionen oder B-Splines (vgl. [10], S. 99-100).

3 Methodik

Dieses Kapitel gibt einen detaillierten Einblick in die methodische Vorgehensweise dieser Arbeit. In erster Linie wird der gesamte Prozess zur Ausreißererkennung vorgestellt und illustriert. Hierbei wird sowohl der Prozess an sich als auch dessen Implementierung, die Vorgehensweise im Anwendungsfall sowie die abschließende Evaluierung der Ergebnisse beschrieben.

Der in Abbildung 3.1 beschriebene mehrstufige Prozess lässt sich in zwei Hauptphasen unterteilen. In der ersten Phase erfolgt ein Clustering der Daten in verschiedene Typen von Verbrauchskurven. Die Einbindung dieser Phase in den Prozess lässt sich durch empirische Beobachtungen im Datensatz des Anwendungsfalls sowie Erfahrungswerten und inhaltlicher Expertise des Unternehmens *nista* begründen. Clustering-Algorithmen an sich sind ein weit verbreiteter Ansatz zur Ausreißererkennung aus dem Bereich des unsupervised Learnings. [6] In diesem Kontext jedoch dient dieser Schritt zur Typisierung der Kurven. So können zuerst Betriebszustände identifiziert werden, welche gegebenenfalls noch von Betriebsexpert:innen validiert werden können, bevor innerhalb dieser die Ausreißererkennung erfolgt.

Der zugrundeliegende Algorithmus ist ein k-Means Clustering (siehe 2.2.4), welches auf zwei Arten erfolgt: Einerseits werden die einzelnen Tage als 96-dimensionale Vektoren betrachtet und der Algorithmus direkt auf diese angewandt. Andererseits wird vorab eine fPCA durchgeführt und anschließend der k-Means-Algorithmus auf die somit ermittelten Scores der ersten zwei Hauptkomponenten angewandt. In beiden Fällen wird die euklidische Distanz als Distanzmaß verwendet.

Nachdem die vorhandenen Daten in separate Cluster gruppiert wurden, erfolgt nun die zweite Phase des Prozesses. In dieser wird in jedem einzelnen der zuvor erkannten Cluster ein Anomaliedetektionsverfahren durchgeführt. Dieses Verfahren wird in Abschnitt 3.2 genauer erklärt und basiert auf den statistischen Tiefen- und Ähnlichkeitsmaßen *Total Variation Depth* und *Modified Shape Similarity*, welche im Abschnitt 2.2.2 eingeführt wurden. In dieser Phase sind die empirischen Faktoren F_{MSS} und F_{TVD}, welche die Grenzen zur Erkennung von Form- bzw. Größenausreißern bestimmen, zu wählen.

Der gesamte Prozess wurde im Rahmen dieser Arbeit in *Python* implementiert, was in Abschnitt 3.2.2 beschrieben wird.

In Kapitel 3.3 wird die vorgestellte Methodik und deren Implementation in einer Simulationsstudie evaluiert. Analysiert wird die Performance der Anomaliedetektion mittels TVD und MSS, abhängig von der Wahl der empirischen Faktoren. Zusätzlich wird auch die Stärke der Abweichungen der Anomalien vom simulierten Hauptmodell variiert. Weiters wird auch der Einfluss des Clusteringschrittes auf die Performance untersucht und schlussendlich ein Vergleich zu einem bestehenden Ansatz der Firma *nista* gezogen. Gemessen wird die Performance anhand der True-Positive-Rate (TPR) und False-Positive-Rate (FPR). Die TPR definiert sich als der Anteil der erkannten wahren Anomalien an der tatsächlichen Anzahl an Anomalien im Datensatz. Die FPR beschreibt hingegen den Anteil der fälschlicherweise als Anomalien erkannten Messungen an den tatsächlich negativen Fällen. [4]

Im nächsten Schritt wird in Kapitel 4 auf den konkreten Anwendungsfall eingegangen und die vorgestellten Methoden an einem realen Datensatz angewandt. Bei diesem Datensatz ist im Gegensatz zur Simulation nicht im Vorfeld bekannt, welche und wie viele der Messungen "tatsächliche", relevante Anomalien sind. Um die Ergebnisse beurteilen zu können, werden die in diesem Schritt erkannten Anomalien von Expert:innen des Unternehmens *nista* bewertet und deren Relevanz erhoben. Die so gewonnenen Ergebnisse werden im darauffolgenden Kapitel diskutiert.

Abbildung 3.1 gibt einen Überblick über den vorgestellten Prozess.

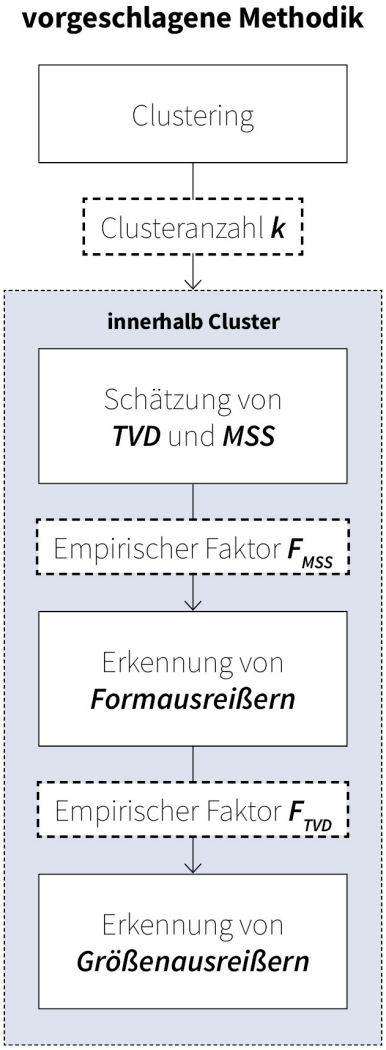

Abbildung 3.1: Methodenübersicht

3.1 Heuristischer Ansatz von *nista*

Eine der von *nista* verwendeten Methoden zur Erkennung von Einsparungspotenzialen in den Daten ist ein mehrstufiger Prozess basierend auf Erfahrungswerten und Beobachtungen in den Daten. In einem ersten Schritt werden die einzelnen Datenpunkte mittels k-Means Clustering gruppiert. Ein Datenpunkt versteht sich hier als ein Vektor $X_i = (x_{i1}, \ldots, x_{im}) \in \mathbb{R}^m$ welcher alle m Messungen eines Stromzählers eines Kalendertages beinhaltet. Im Anwendungsfall wird

in 15-Minuten-Intervallen gemessen, d.h. $m = 96$. Das Unternehmen verwendet hier einen k-Means Algorithmus (siehe 2.16) und als Distanzmaß die euklidische Distanz.

Um eine optimale Anzahl an Clustern (k) festzulegen, wird der Algorithmus für $k \in \{2, \dots, 12\}$ trainiert und dann für jedes k die Summe der quadrierten Abstände (SSD) zu den Clusterzentren errechnet. Dies führt zu einer Verlaufskurve mit den errechneten Summen, abhängig von k, für welche nun der Wert mit dem maximalen Gefälle errechnet wird. Das darauf basierende optimale k wird nun für die Bildung der Cluster verwendet. Nun wird für jeden Cluster ein optimaler Tag $X_{i_j^*}$ für $j = 1, \dots, k$ ermittelt. Dieser definiert sich als der Tag innerhalb eines Clusters C_j mit dem geringsten Energieverbrauch.

$$X_{i_j^*} = \arg\min_{i \in C_j} \sum_{l=1}^{m} x_{il} \tag{3.1}$$

Nun wird für jedes X_i die Distanz zum zugehörigen Clusteroptimum bestimmt.

$$D_i = \sum_{l=1}^{m} |x_{il} - x_{i_j^*,l}|, i \in C_j \tag{3.2}$$

Im nächsten Schritt wird für jeden Cluster eine Schranke $\bar{D}_{C_j}$ für die Distanzen festgesetzt. Gilt nun für einen Tag i im Cluster C_j, dass $D_i > \bar{D}_{C_j}$, so wird dieser als Ausreißer markiert.

Für die Anwendung dieses Algorithmus in der Simulationsstudie in Abschnitt 3.3 wird dieser etwas angepasst, um den gegebenen Anforderungen besser zu entsprechen. Anstatt des in 3.1 beschriebenen Clusteroptimums wird die Mediankurve herangezogen, da ansonsten Größenausreißer mit einem negativen Offset als Optimum erkannt werden. Als Schranken werden Perzentile der Distanzwerte D_i verwendet.

3.2 Ausreißererkennung mittels *TVD* und *MSS*

Der Prozess zur separaten Erkennung von Größen- und Formausreißern mittels *TVD* und *MSS*, welcher in der vorgestellten Methodik anschließend an das Clustering in jedem einzelnen Cluster erfolgt, wurde von Huang und Sun eingeführt [7]. Er setzt sich aus folgenden Schritten zusammen:

1. Für jede Kurve, also für jedes X_i werden die *TVD* und *MSS* geschätzt (die Schätzung von *TVD* und *MSS* wird in Abschnitt 3.2.1 beschrieben)
2. Mittels eines klassischen Boxplot der geschätzten *MSS* werden Formausreißer identifiziert. Ein X_i ist dann ein Formausreißer, wenn gilt $MSS(X_i) < F_{MSS} * IQR$ wobei IQR der Interquartilsabstand ist und F_{MSS} manuell gewählt werden kann.
3. Die erkannten Formausreißer werden nun aus der Stichprobe entfernt und basierend auf der *TVD* wird ein Funktionaler Boxplot erstellt, um Größenausreißer nach der F_{TVD} * "Central-Region"-Regel zu identifizieren. (siehe 2.2.3) [7]

Abbildung 3.2 illustriert den oben beschriebenen Prozess.

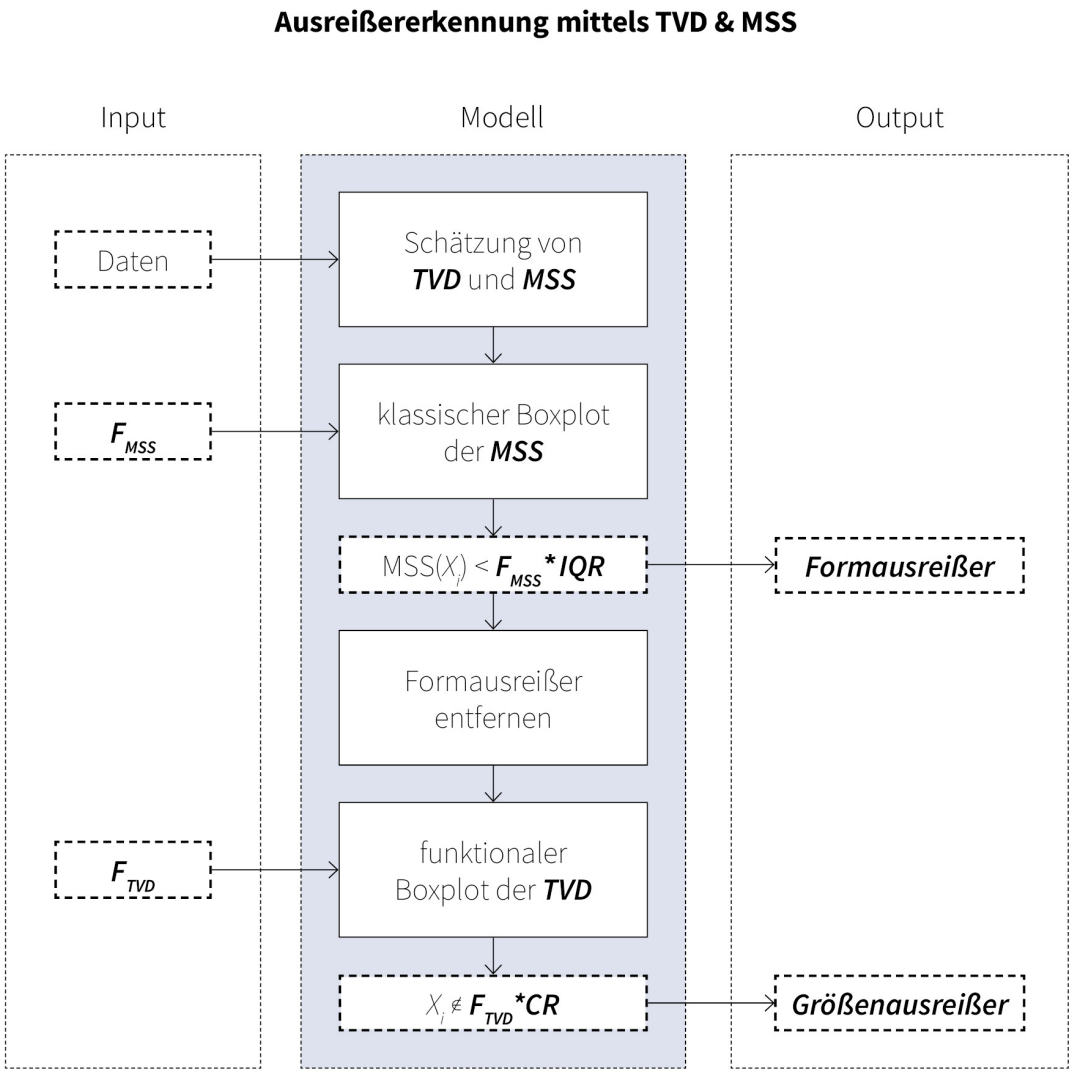

Abbildung 3.2: Prozess zur Erkennung von Ausreißern mittels *TVD* und *MSS*

3.2.1 Schätzung von *Total-Variation-Depth* und *Shape-Similarity*

Die in 2.2.2 eingeführten Definitionen für TVD und MSS beziehen sich auf Funktionen auf einem Intervall in $\mathbb{R}$. In der Praxis liegen die Daten jedoch in Form von diskreten Messungen vor. Seien $X_i(t_j), i = 1, \ldots, n, j = 1, \ldots m$ nun n Messungen zu m Zeitpunkten, sind die empirische *TVD* und *SS* folgendermaßen definiert:

$$\widehat{TVD}(X_i) = \frac{1}{m}\sum_{j=1}^{m} \hat{D}_{X_i}(t_j), \tag{3.3}$$

$$\widehat{SS}(X_i) = \sum_{j=2}^{m} \hat{v}_{X_i}(t_j)\hat{S}_{X_i}(t_j; t_{j-1}) \tag{3.4}$$

$D_{X_i}(t)$ wurde in Kapitel 2.2.2 als $var\{\mathbf{1}\{X(t) \leq X_i(t)\}\}$ definiert. Empirisch schätzen wir diese Varianz durch $\hat{D}_{X_i}(t_j) = \hat{p}_{X_i}(t_j)\{1-\hat{p}_{X_i}(t_j)\}$, wobei $\hat{p}_{X_i}(t_j) = \#\{s : X_s(t_j) \leq X_i(t_j)\}/n$, also der Anteil der Beobachtungen $X_s(t_j)$, die unter $X_i(t_j)$ liegen.

Für die *Shape-Similarity* definiert sich die empirische Gewichtsfunktion als

$$\hat{v}_{X_i}(t_j) = \frac{|X_i(t_j) - X_i(t_{j-1})|}{\sum_{r=2}^{m} |X_i(t_r) - X_i(t_{r-1})|} \tag{3.5}$$

In Kapitel 2.2.2 hatten wir die Shape Similarity punktweise als den Anteil der Varianztiefe, welcher in Abhängigkeit von dem Zeitpunkt $t - \Delta$ ist – die sogenannte Formkomponente der TVD – im Verhältnis zur Gesamtvarianztiefe definiert:

$$S_{X_i}(t;\Delta) = \begin{cases} \dfrac{var[\mathbb{E}\{\mathbf{1}\{X(t) \leq X_i(t)\}|\mathbf{1}\{X(t-\Delta) \leq X_i(t-\Delta)\}\}]}{D_{Xi}(t)}, & D_{X_i}(t) \neq 0 \\ 1, & D_{X_i}(t) = 0 \end{cases} \tag{3.6}$$

Den Schätzer für $D_{X_i}(t)$ hatten wir bereits oben definiert. Die Formkomponente formulieren wir für $\Delta = t_j - t_{j-1}$ folgendermaßen um:

$$\begin{aligned} & var[\mathbb{E}\{\mathbf{1}\{X(t_j) \leq X_i(t_j)\}|\mathbf{1}\{X(t_{j-1}) \leq X_i(t_{j-1})\}\}] \\ = {} & var[\mathbb{P}\{X(t_j) \leq X_i(t_j) \mid \mathbf{1}\{X(t_{j-1}) \leq X_i(t_{j-1})\}] \\ = {} & \mathbb{E}[\mathbb{P}^2\{X(t_j) \leq X_i(t_j) \mid \mathbf{1}\{X(t_{j-1}) \leq X_i(t_{j-1})\}] \\ & - \mathbb{E}^2[\mathbb{P}\{X(t_j) \leq X_i(t_j) \mid \mathbf{1}\{X(t_{j-1}) \leq X_i(t_{j-1})\}] \end{aligned}$$

Den zweiten Term, $\mathbb{E}^2[\mathbb{P}\{X(t_j) \leq X_i(t_j) \mid \mathbf{1}\{X(t_{j-1}) \leq X_i(t_{j-1})\}]$, schätzen wir durch $\hat{p}^2_{X_i}(t_j)$ und den ersten Term formulieren wir wie folgt um:

$$
\begin{aligned}
&\mathbb{E}[\mathbb{P}^2\{X(t_j) \leq X_i(t_j) \mid \mathbf{1}\{X(t_{j-1}) \leq X_i(t_{j-1})\}] \\
&= \mathbb{P}\{\mathbf{1}\{X(t_{j-1}) \leq X_i(t_{j-1})\} = 0\}\mathbb{P}^2\{X(t_j) \leq X_i(t_j) \mid \mathbf{1}\{X(t_{j-1}) \leq X_i(t_{j-1})\} = 0\} \\
&\quad + \ \mathbb{P}\{\mathbf{1}\{X(t_{j-1}) \leq X_i(t_{j-1})\} = 1\}\mathbb{P}^2\{X(t_j) \leq X_i(t_j) \mid \mathbf{1}\{X(t_{j-1}) \leq X_i(t_{j-1})\} = 1\} \\
&= \mathbb{P}\{X(t_{j-1}) > X_i(t_{j-1})\}\mathbb{P}^2\{X(t_j) \leq X_i(t_j) \mid X(t_{j-1}) > X_i(t_{j-1})\} \\
&\quad + \ \mathbb{P}\{X(t_{j-1}) \leq X_i(t_{j-1})\}\mathbb{P}^2\{X(t_j) \leq X_i(t_j) \mid X(t_{j-1}) \leq X_i(t_{j-1})\}
\end{aligned}
$$

Sei nun

$$
\hat{p}_{X_i}(t_j, t_{j-1}^-) = \#\{s : X_s(t_j) \leq X_i(t_j), X_s(t_{j-1}) \leq X_i(t_{j-1})\}/n, \tag{3.7}
$$

und

$$
\hat{p}_{X_i}(t_j, t_{j-1}^+) = \#\{s : X_s(t_j) \leq X_i(t_j), X_s(t_{j-1}) > X_i(t_{j-1})\}/n, \tag{3.8}
$$

dann schätzen wir $\mathbb{E}[\mathbb{P}^2\{X(t_j) \leq X_i(t_j) \mid \mathbf{1}\{X(t_{j-1}) \leq X_i(t_{j-1})\}]$ durch

$$
\frac{\hat{p}_{X_i}^2(t_j, t_{j-1}^-)}{\hat{p}_{X_i}(t_{j-1})} + \frac{\hat{p}_{X_i}^2(t_j, t_{j-1}^+)}{1 - \hat{p}_{X_i}(t_{j-1})} \tag{3.9}
$$

wenn gilt $\hat{p}_{X_i}(t_{j-1}) \neq 1$. Wenn $\hat{p}_{X_i}(t_{j-1}) = 1$, fällt der zweite Term weg. Somit ergibt sich der Schätzer für $S_{X_i}(t_j; t_{j-1})$:

$$
\hat{S}_{X_i}(t_j; t_{j-1}) = \begin{cases}
1, & \hat{p}_{X_i}(t_j) = 1, \\
\dfrac{\hat{p}_{X_i}^2(t_j, t_{j-1}^-)/\hat{p}_{X_i}(t_{j-1}) - \hat{p}_{X_i}^2(t_j)}{\hat{p}_{X_i}(t_j)\{1 - \hat{p}_{X_i}(t_j)\}}, & \hat{p}_{X_i}(t_j) \neq 1, \hat{p}_{X_i}(t_{j-1}) = 1, \\
\dfrac{\left\{\dfrac{\hat{p}_{X_i}^2(t_j, t_{j-1}^-)}{\hat{p}_{X_i}(t_{j-1})} + \dfrac{\hat{p}_{X_i}^2(t_j, t_{j-1}^+)}{1 - \hat{p}_{X_i}(t_{j-1})}\right\} - \hat{p}_{X_i}^2(t_j)}{\hat{p}_{X_i}(t_j)\{1 - \hat{p}_{X_i}(t_j)\}}, & \hat{p}_{X_i}(t_j) \neq 1, \hat{p}_{X_i}(t_{j-1}) \neq 1,
\end{cases} \tag{3.10}
$$

(Vgl. [7], S. 457)

3.2.2 Implementierung in Python

Für eine effiziente Integration der vorgestellten Methodik in die bestehende Infrastruktur von *nista* wurde der von Huang und Sun [7] eingeführte Prozess zur Ausreißererkennung, sowie die Berechnung der Schätzungen für *Total-Variation-Depth* und *Modified-Shape-Similarity* in *Python* implementiert. Huang und Sun [7] stellten eine Implementation ihrer Methodik in *R* vor, an welcher sich der Algorithmus in *Python* orientiert.

Der Algorithmus nutzt die Funktionalitäten der *Python*-Libraries "numpy"[5], "matplotlib"[8] sowie "scikit-fda"[13]. "Scikit-fda" ist eine *Python*-Library, welche eine Vielzahl an Methoden und Visualisierungsmöglichkeiten für funktionale Datenanalyse bereitstellt. Eine dieser ist die Organisation der Daten als ein sogenanntes *FDataGrid*-Objekt. Dieses beinhaltet einerseits die Messwerte in Form eines mehrdimensionalen Arrays, sowie ein eindimensionales Array der Messzeitpunkte. Alle implementierten Funktionalitäten arbeiten mit dieser Datenstruktur, um die zusätzliche Verwendung von Methoden der *scikit-fda*-Library zu erleichtern.

Der implementierte Code bietet zum einen die Möglichkeit, die Indizes der erkannten Ausreißer zu erhalten. Zusätzlich können auch ein klassischer Boxplot der *MSS*, in welchem Formausreißer farblich gekennzeichnet werden, sowie ein funktionaler Boxplot der *TVD*, ebenfalls mit farblicher Kennzeichnung der Ausreißer, erstellt werden. Beide dieser Visualisierungen bieten die Möglichkeit den empirischen Faktor, sowohl für *TVD*, als auch für *MSS*, manuell zu variieren.

3.3 Simulationsstudie

Bevor die in *Python* implementierte Methode an den Datensätzen des Unternehmens *nista* zur Anwendung kommt, wird die Funktionalität an simulierten Daten erprobt. In einem ersten Schritt wird nur die Funktionsweise der Ausreißererkennung mittels *TVD* und *MSS* getestet. Analog zu Dai und Genton [2], wurden unterschiedliche Form- und Größenausreißer simuliert:

1. **Hauptmodell:**

$$X(t) = 4t + e(t) \tag{3.11}$$

   Wobei $e(t)$ ein Gauss-Prozess mit $\mu = 0$ und Kovarianzfunktion $\gamma(s,t) = exp\{-|t-s|\}$ ist und $t \in T = [0,1]$.

2. **Größenausreißer**

$$X_{Magnitude}(t) = 4t + 20 + e(t) \tag{3.12}$$

Dieser Ausreißer verhält sich analog zum Hauptmodell, jedoch mit einem konstanten Offset von $+20$.

3. **Formausreißer (Shift)**

$$X_{Shift}(t) = 4t + 10 \cdot \mathbf{1}_{t>0.4} + e(t) \tag{3.13}$$

Dieser Ausreißer hat einen konstanten Offset von $+10$, allerdings nur ab dem Zeitpunkt $t > 0.4$.

4. **Formausreißer (Peak)**

$$X_{Peak}(t) = 4t - 10 \cdot \mathbf{1}_{0.25<t<0.3} + e(t) \tag{3.14}$$

Dieser Ausreißer hat einen negativen Offset zwischen den Zeitpunkten 0,25 und 0,3.

5. **Formausreißer (Sinusoidal)**

$$X_{Sinus}(t) = 4t + 2 \cdot sin(18t) + e(t) \tag{3.15}$$

Dieser Ausreißer unterscheidet sich durch einen sinusoidalen Verlauf von der Form der Kurven des Hauptmodells.

6. **Formausreißer (Slope)**

$$X_{Slope}(t) = 10t + e(t) \tag{3.16}$$

Dieser Ausreißer unterscheidet sich durch eine andere Steigung von den Kurven des Hauptmodells.

7. **Form- & Größenausreißer**

$$X_{Magnitude+Shape}(t) = 4t + 2 \cdot sin(18t) - 20 + e(t) \tag{3.17}$$

Dieser Ausreißer unterscheidet sich sowohl hinsichtlich seiner Form, durch einen sinusoidalen Verlauf, als auch durch seine Größenordnung, durch einen Offset von -20 von den Kurven des Hauptmodells.

Simuliert wurden $n = 200$ Kurven des Hauptmodels mit den 6 unterschiedlichen Ausreißern auf gleichmäßigen Intervallen im Bereich $[0, 1]$ mit $p = 100$ Zeitpunkten. Zur Simulation wurde die *Python*-Library "scifit-fda" [13], sowie "numpy"[5] verwendet. Abbildung 3.3 zeigt die Kurven des Hauptmodells in grau und die Ausreißer in schwarz.

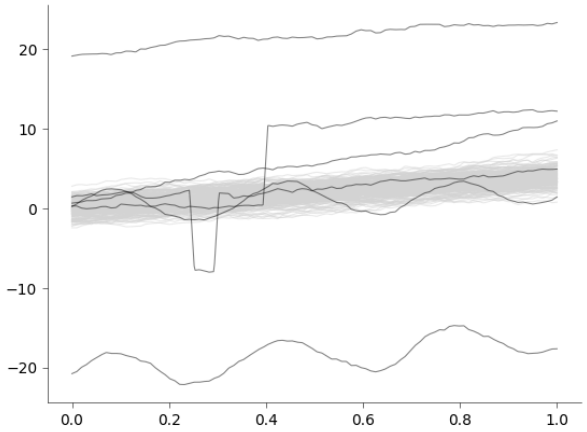

Abbildung 3.3: Simulierte Daten

An diesen simulierten Daten wurde in einem ersten Schritt der in *Python* nach Huang und Sun [7] implementierte Algorithmus zur Ausreißererkennung (siehe 3.2.2) angewandt. Als empirischen Faktor für die MSS wurde analog zu [7] $F_{MSS} = 3$ festgelegt. Für die TVD wurde $F_{TVD} = 1.5$ gewählt. Abbildung 3.4 zeigt die durch den Algorithmus erkannten Ausreißer. Formausreißer sind Orange eingefärbt und Größenausreißer in rot. Abbildung 3.5 zeigt den klassischen Boxplot mit den errechneten Werten der MSS. Die erkannten Formausreißer sind in rot markiert. Abbildung 3.6 zeigt den funktionalen Boxplot basierend auf der errechneten TVD, nachdem die Formausreißer entfernt wurden. Die Mediankurve ist in schwarz dargestellt, die Central-Region ist der hellgraue Bereich. Die maximale Hülle ohne Ausreißer wird durch die schwarzen Linien angezeigt. Die beiden Größenausreißer sind in rot dargestellt. Wie in 3.4 gut zu sehen ist, wurden alle simulierten Form- und Größenausreißer korrekt als solche erkannt.

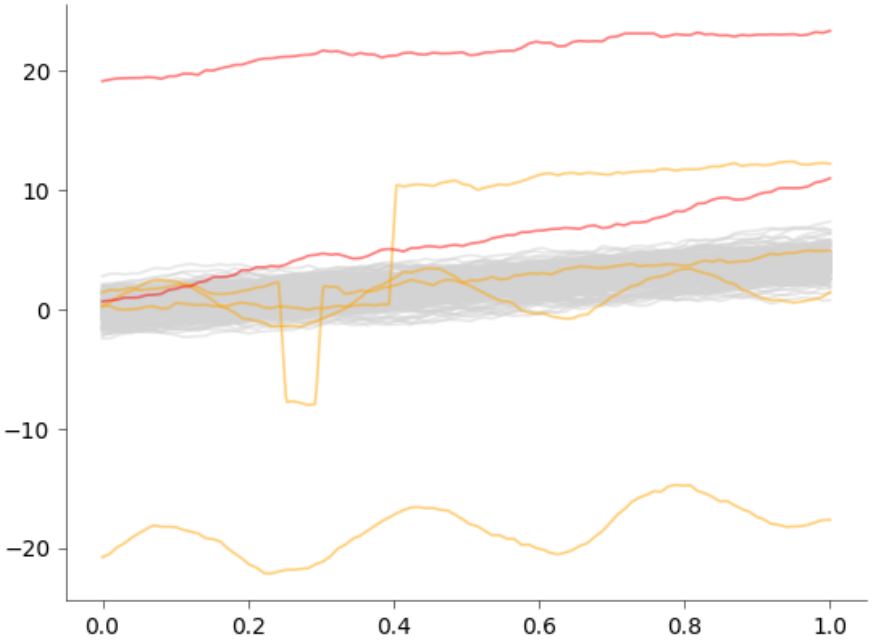

Abbildung 3.4: Erkannte Ausreißer mittels TVD & MSS

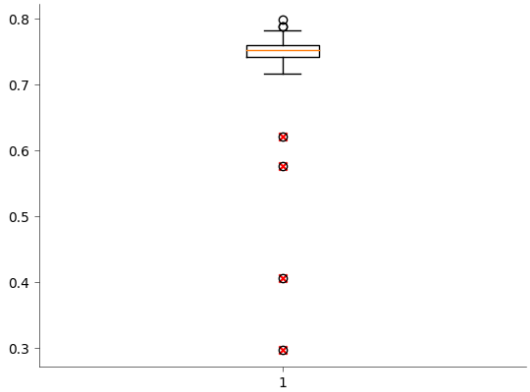

Abbildung 3.5: Boxplot der MSS

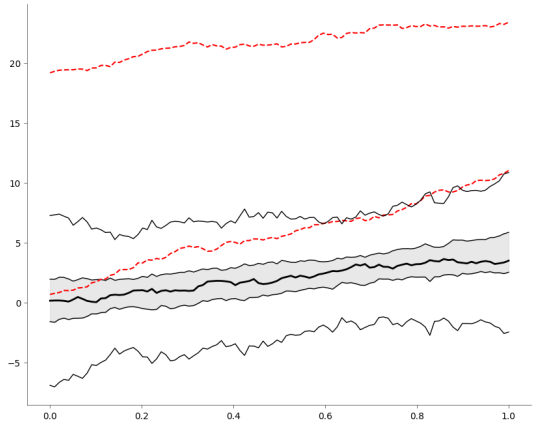

Abbildung 3.6: Funktionaler Boxplot der simulierten Kurven

3.3.1 Wahl der empirischen Faktoren

Um die Auswirkungen der Werte von F_{MSS} und F_{TVD} genauer zu analysieren, wurden Variationen dieser an den in Kapitel 3.3 angeführten simulierten Modellen angewandt. Die Werte der beiden Faktoren wurden jeweils in 0,5-er Schritten von 1 bis 3 variiert. Für jede dieser Kombination dieser Werte wurden die $n = 206$ Kurven (200 des Hauptmodells, 6 Ausreißer) $N_{SIM} = 50$

mal simuliert und über die jeweils errechneten True-Positive- und False-Positive-Rates gemittelt. Die Ergebnisse dieser Simulation sind in Abbildungen 3.7 und 3.8 zu sehen.

Abbildung 3.7: TPR der empririschen Faktoren

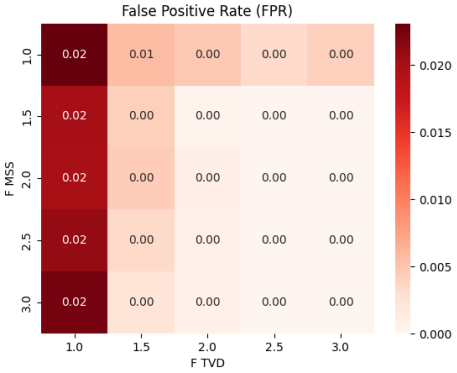

Abbildung 3.8: FPR der empririschen Faktoren

Hier kann man sehen, dass sich in dieser Studie höhere Werte für F_{MSS} negativ auf die TPR auswirken, bei den Werten für F_{TVD} dieser Effekt jedoch nur minimal auftritt. Die höchste TPR wird bei $F_{MSS} = 1$ und $F_{TVD} < 3$ erzielt. Was die FPR betrifft, so lässt sich hier ein gegenteiliger Effekt beobachten. Generell ist diese sehr nahe bei 0 mit einem Maximum von 0.02 für $F_{TVD} = 1$, wobei eine Veränderung von F_{MSS} kaum einen Effekt auf diese zu haben scheint. Hinsichtlich dieser Metriken wird bei $F_{MSS} < 1.5$ und $F_{TVD} \in [2, 2.5]$ ein Optimum erzielt mit einer TPR von 0.99 und einer FPR von 0.

3.3.2 Abweichungen der Ausreißer vom Hauptmodell

Eine weitere Frage, welche sich zur Performance der vorgestellten Methodik stellt, ist, inwiefern sich diese verhält, je nachdem wie stark die simulierten Ausreißer vom Hauptmodell abweichen. Um dieser auf den Grund zu gehen, wurden die Abweichungen der Ausreißerkurven vom Hauptmodell jeweils mit einem Faktor F_{OF} skaliert und die Simulation aus dem vorhergehenden Abschnitt 3.3.1 für drei unterschiedliche Werte für $F_{OF} = [0.3, 0.65, 1]$ durchgeführt ($F_{OF} = 1$ entspricht in diesem Fall der ursprünglichen Vorgehensweise analog zu Abschnitt 3.3.1). Abbildungen 3.9 - 3.11 veranschaulichen die dabei simulierten Datensätze und Abbildungen 3.12-3.15 die ermittelten Werte der TPR und FPR der vorgestellten Methodik für die unterschiedlich starken Abweichungen.

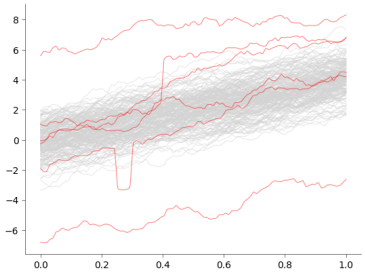

Abbildung 3.9: Simulierte Daten mit $F_{OF} = 0.3$

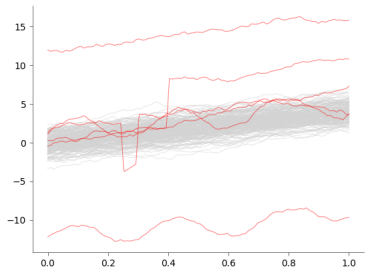

Abbildung 3.10: Simulierte Daten mit $F_{OF} = 0.65$

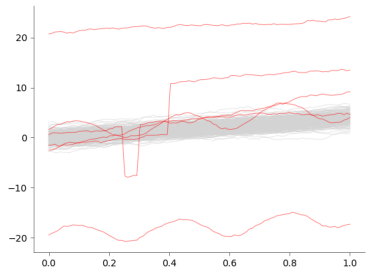

Abbildung 3.11: Simulierte Daten mit $F_{OF} = 1$

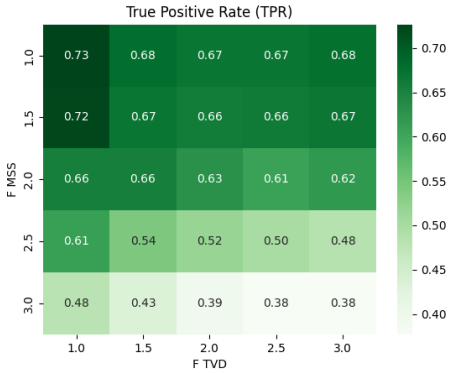

Abbildung 3.12: TPR $F_{OF} = 0.3$

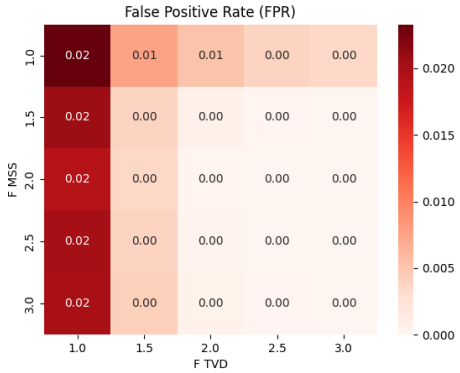

Abbildung 3.13: PFR $F_{OF} = 0.3$

Abbildung 3.14: TPR $F_{OF} = 0.65$

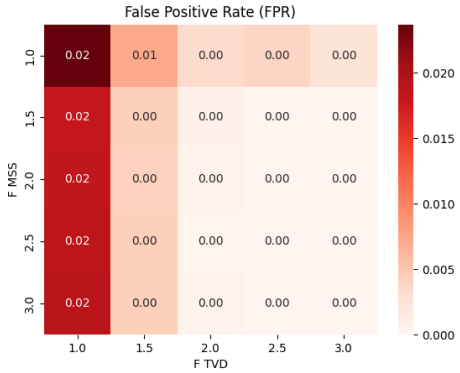

Abbildung 3.15: PFR $F_{OF} = 0.65$

Hier ist zu beobachten, dass je weniger stark die Ausreißer vom Hauptmodell abweichen, umso niedriger ist die TPR, über alle Faktorkombinationen hinweg. Die FPR hingegen verhält sich bei $F_{OF} = 0.65$ gleich wie bei $F_{OF} = 1$ und nur minimal höher bei $F_{OF} = 0.3$.

3.3.3 Simulation inklusive Clusteringschritt

Aufbauend auf den ersten Schritt der Simulationsstudie wurden in weiterer Folge zu dem simulierten Datensatz noch ein zweiter "Cluster" — also zusätzliche Daten aus einem abweichenden Hauptmodell — simuliert. Dieser beinhaltet ebenfalls 200 Stichproben und 4 Ausreißer. Abbildung 3.16 zeigt die simulierten Kurven, wobei die Ausreißer in schwarz gekennzeichnet sind.

1. **Hauptmodell 2:**

$$X(t) = t + 6 * sin(6t) + 20 + e(t) \tag{3.18}$$

   Wobei $e(t)$ ein Gauss-Prozess mit $\mu = 0$ und Kovarianzfunktion $\gamma(s,t) = exp\{-|t-s|\}$ ist.

2. **Größenausreißer**

$$X_{Magnitude2}(t) = t + 6 \cdot sin(6t) + 25 + e(t) \tag{3.19}$$

   Dieser Ausreißer verhält sich analog zum Hauptmodell, jedoch mit einem konstanten Offset von $+5$.

3. **Formausreißer (Amplitude)**

$$X_{Amplitude}(t) = t + 13 \cdot sin(6t) + 20 + e(t) \tag{3.20}$$

   Dieser Ausreißer weicht vom Hauptmodell durch eine höhere Amplitude ab.

4. **Formausreißer (Frequenz)**

$$X_{Frequenz}(t) = t + 6 \cdot sin(9t) + 20 + e(t) \tag{3.21}$$

   Dieser Ausreißer weicht vom Hauptmodell durch eine höhere Frequenz ab.

5. **Formausreißer (Shift)**

$$X_{Shift2}(t) = t + 6 \cdot sin(9t) + 20 + 5 \cdot \mathbf{1}_{t>0.4} + e(t) \tag{3.22}$$

Dieser Ausreißer weicht vom Hauptmodell durch einen Shift um $+5$ ab dem Zeitpunkt $t > 0.4$ ab.

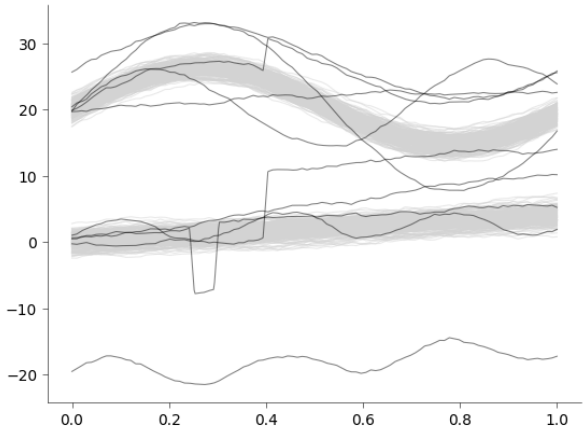

Abbildung 3.16: Simulierte Daten mit zwei Clustern

Zuerst wurde eine Anomalieerkennung ohne den Clusteringschritt durchgeführt. Die empirischen Faktoren F_{MSS} und F_{TVD} wurden beide mit 1.5 sensibel gewählt. Wie Abbildungen 3.17-3.19 illustrieren, wurde nur einer der Ausreißer mittels MSS erkannt und keiner durch den funktionalen Boxplot. Hier ist zu beobachten, dass der Algorithmus fast alle Kurven als sehr ähnlich einstuft, also dass die MSS hier beinahe ausschließlich Werte sehr nahe an 1 schätzt. Zusätzlich ist hier gut zu sehen, dass die errechnete maximale Hülle ohne Ausreißer im Funktionalen Boxplot einen relativ großen Bereich abdeckt und sich alle simulierten Ausreißer innerhalb dieses Bereichs befinden, und diese folglich nicht als solche erkannt werden.

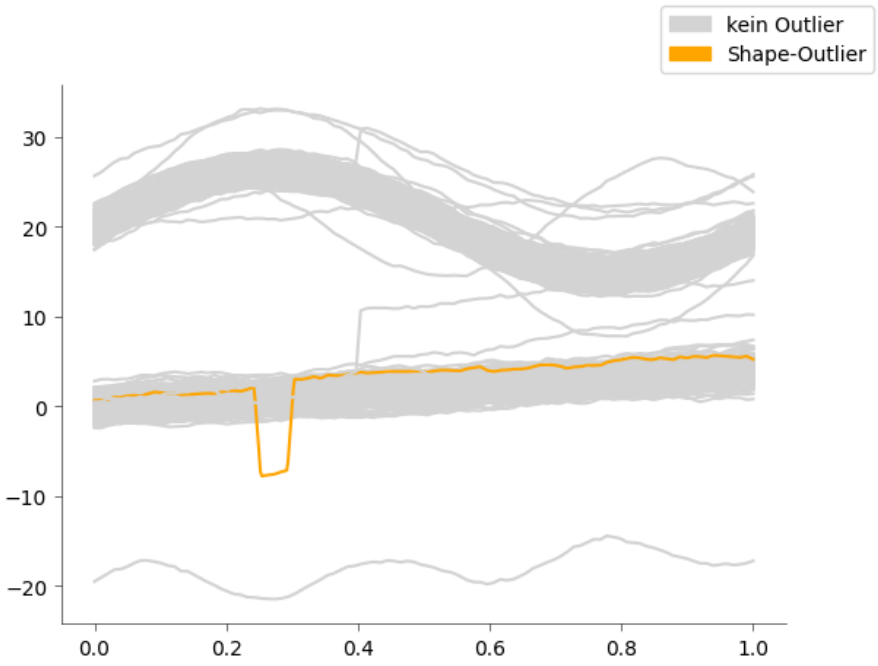

Abbildung 3.17: Erkannte Ausreißer mittels TVD & MSS ohne Clustering-Schritt

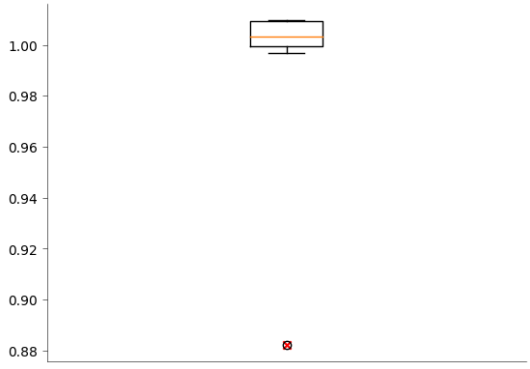

Abbildung 3.18: Boxplot der MSS ohne Clustering-Schritt

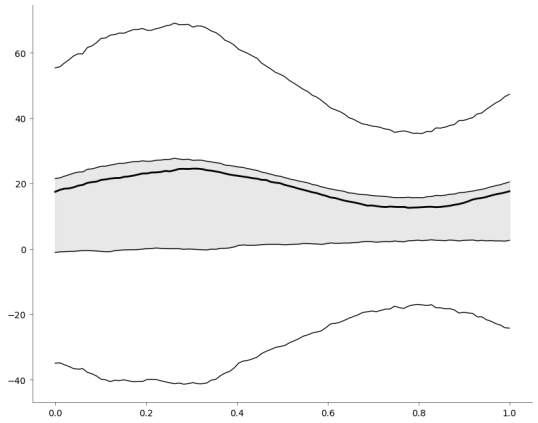

Abbildung 3.19: Funktionaler Boxplot ohne Clustering-Schritt

Als nächster Schritt wurde vor dem Verfahren zur Ausreißererkennung nach Huang und Sun [7], entsprechend der in dieser Arbeit vorgestellten Methodik, ein k-Means-Clustering der Daten mit k=2 durchgeführt. Dieses teilt die simulierten Kurven in zwei Gruppen, welche in den Abbildungen 3.20 - 3.21 zu sehen sind.

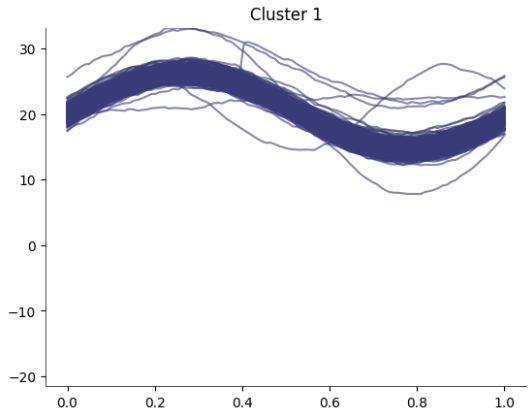

Abbildung 3.20: Simulierter Cluster 1

Abbildung 3.21: Simulierter Cluster 2

Anschließend an das Clustering folgt, wie in Kapitel 3 beschrieben, in jedem dieser Cluster separat die Anomalieerkennung mittels TVD und MSS. Wie in den Abbildungen 3.22 - 3.23 zu sehen ist, wurden hier alle simulierten Anomalien erkannt. Als Werte für F_{TVD} und F_{MSS} wurden anfangs 1,5 gewählt und diese Wahl in weiterer Folge verfeinert. In diesem Fall führten Werte von 1,5 für F_{TVD} und 1,8 für F_{MSS} zu einem optimalen Ergebnis, d.h. dass alle Ausreißer korrekt als solche erkannt werden und keine der Kurven aus einem der Hauptmodelle fälschlicherweise als Ausreißer angenommen werden. Abbildungen 3.23 - 3.27 zeigen die Ergebnisse des Prozesses mit den optimierten empirischen Faktoren.

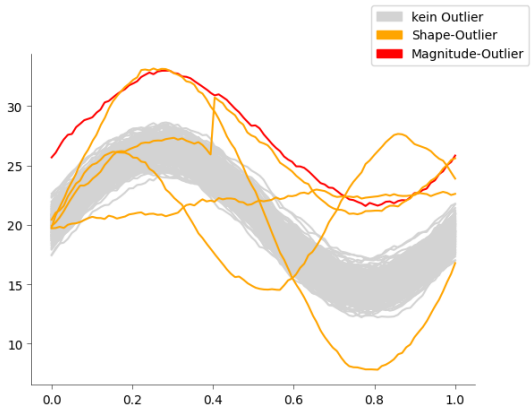

Abbildung 3.22: Ausreißer Cluster 1

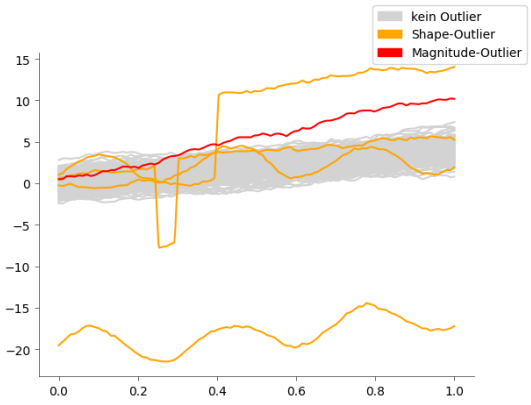

Abbildung 3.23: Ausreißer Cluster 2

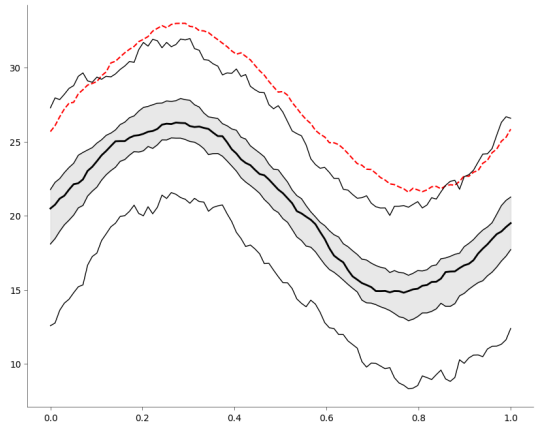

Abbildung 3.24: Funktionaler Boxplot Cluster 1

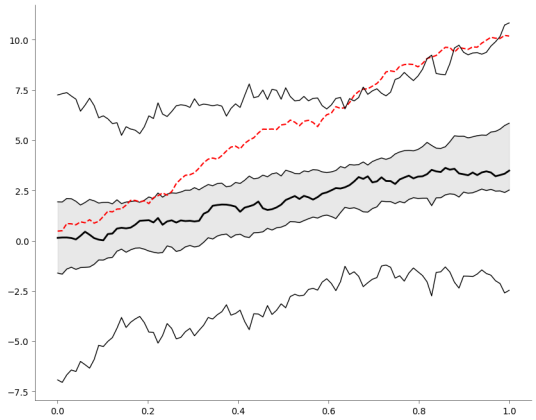

Abbildung 3.25: Funktionaler Boxplot 2

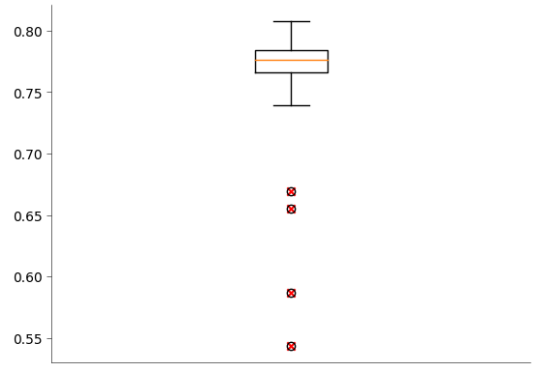

Abbildung 3.26: MSS-Boxplot Cluster 1

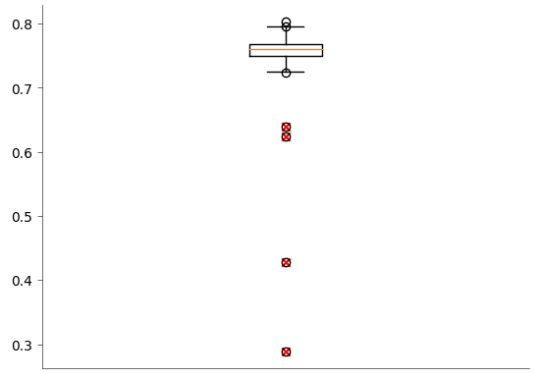

Abbildung 3.27: MSS-Boxplot Cluster 2

Performanceverlgeich: Clustering vs. kein Clustering

Analog zu der in Abschnitt 3.3.1 definierten Frage zur Performance der Methodik für unterschiedliche Kombinationen von empirischen Faktoren, stellt sich diese auch hinsichtlich des Vergleichs bei simulierten “Clustern” bzw. bei zwei bestehenden Hauptmodellen. Diesbezüglich wurde die in Abschnitt 3.3.3 vorgestellte Simulation ebenfalls als Monte-Carlo-Simulation mit $N_{SIM} = 50$ bei variierenden empirischen Faktoren jeweils mit und ohne dem Clusteringschritt durchgeführt. Die Ergebnisse sind in Abbildungen 3.28 - 3.29 dargestellt. Zieht man die Kombination beider Metriken in Betracht, so kann beobachtet werden, dass das Hinzufügen des Clusterings bei jeglicher Wahl der empirischen Faktoren zu einem besseren Ergebnis führt.

Abbildung 3.28: TPR ohne Clusteringschritt

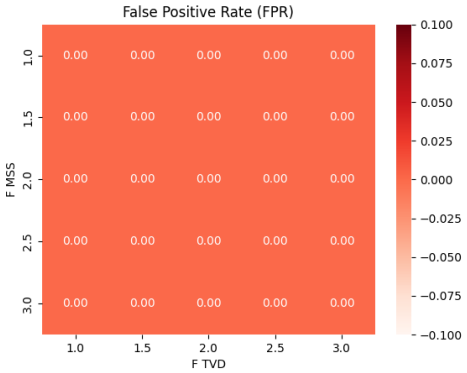

Abbildung 3.29: FPR ohne Clusteringschritt

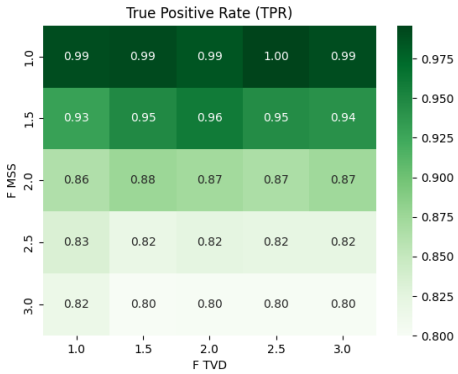

Abbildung 3.30: TPR mit Clusteringschritt

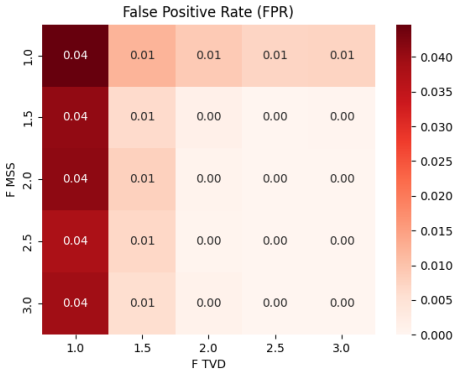

Abbildung 3.31: FPR mit Clusteringschritt

3.3.4 Methodenvergleich

Um die Performance der vorgestellten Methode im Kontext zu anderen Ansätzen zu bewerten, wird diese nun mit einem “naiven”, auf einer heuristischen Methode der Firma *nista* basierendem Ansatz verglichen. Dazu wurden analog zu Abschnitt 3.3.3 Datensätze mit $n = 410$ Kurven mit 10 Ausreißern und zwei Hauptmodellen aus welchen jeweils 200 Kurven simuliert wurden herangezogen.

Der “naive” Ansatz erfolgt, wie in Kapitel 3.1 beschrieben, ebenfalls mit einem Clusteringschritt und einer anschließenden Anomalieerkennung innerhalb der Cluster. In diesem Schritt kann der Parameter $\bar{D}$, eine Schranke ab welcher potenzielle Ausreißer als solche erkannt werden,

variiert werden. Abbildungen 3.32 - 3.33 zeigen die Ergebnisse der vorgestellten Methodik (analog zu Kapitel 3.3.3) und 3.34 zeigt die True- und False-Positive-Rate des *nista*-Algorithmus, abhängig von der Wahl des Parameters $\bar{D}$. Hier ist gut ersichtlich, dass dieser, bei entsprechender Anpassung des Parameters, ebenfalls eine niedrige FPR aufweist, die TPR jedoch ein Maximum bei knapp über 0,8 hat. Insgesamt kann gesagt werden, dass die vorgestellte Methode eine bessere Performance, sowohl hinsichtlich der TPR als auch der FPR, als der “naive” Ansatz erzielt.

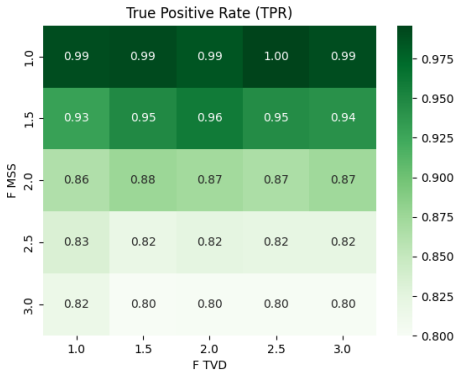

Abbildung 3.32: TPR vorgestellte Methode

Abbildung 3.33: FPR vorgestellte Methode

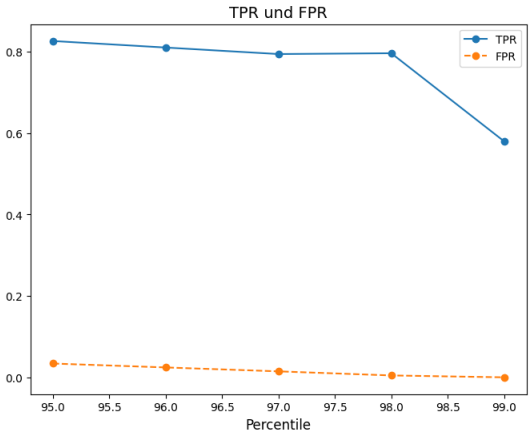

Abbildung 3.34: TPR & FPR “naive” Methode

4 Anwendungsfall: Smartmeter Daten von *nista*

4.1 Beschreibung der Daten

In diesem Kapitel werden die für die Analyse verwendeten Daten vorgestellt und detailliert beschrieben. Die Daten stammen aus einer Produktionsstätte eines industriellen Fertigungsunternehmens und wurden von *nista* für diese Arbeit zur Verfügung gestellt. Es handelt sich um eindimensionale Zeitreihendaten des Energieverbrauchs- bzw. Smart-Meter-Stromzählerdaten. Diese wurden in 15-Minuten-Intervallen gesammelt und bilden für jeden Zeitpunkt den Stromverbrauch in Kilowatt (kW) ab. Aufgrund der Natur der Messwerte nehmen die Daten nur positive Werte an, d.h. es gibt keinen "negativen Stromverbrauch". Als Zeitraum der Analyse wurde das Kalenderjahr 2023, somit das zum Zeitpunkt der Analyse aktuellste vollständige Kalenderjahr, gewählt. Die Daten haben keine fehlenden Werte.

Für die Analyse werden die Verbrauchskurven der einzelnen Tage analysiert und jeder Tag als eine Funktion $X_i(t_j), i = 1, \ldots, n, j = 1, \ldots, m$, wobei $n = 365$ und $m = 96$, modelliert.

Zuerst wird ein Überblick über die Verteilung der Werte, die Struktur der einzelnen Kurven und allgemeine Auffälligkeiten gegeben. Anschließend wird genauer auf einzelne Wochentage, Feiertage und saisonale Besonderheiten eingegangen.

Das Histogramm in 4.2 gibt einen Überblick über die Verteilung der Werte des gesamten Datensatzes. Hier kann man gut sehen, dass der häufigste Wert bei ca. 100 kW liegt. Eine weitere Spitze ist bei 1100 kW zu erkennen.

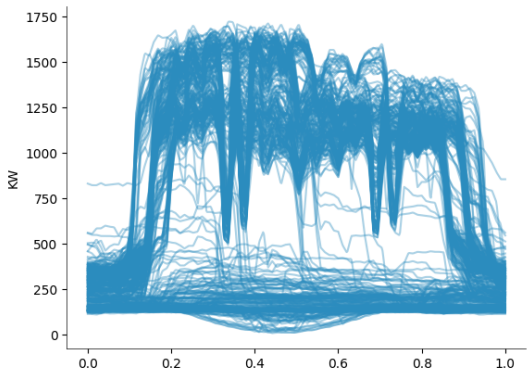

Abbildung 4.1: Verbrauchskurven

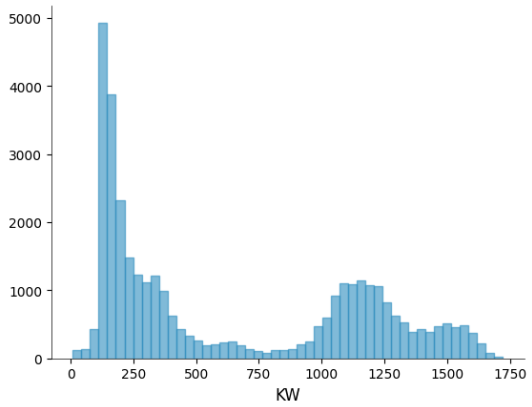

Abbildung 4.2: Verteilung der Werte

Gruppiert man die Tage in Wochentage und Wochenenden und Feiertage, so ergeben sich zwei unterschiedliche Verteilungen. An den Wochentagen hat die Verteilung zwei Modi, einmal bei ca. 250 kW und einmal bei 1200 kW. An Wochenenden und Feiertagen lässt sich eine uni-modale Verteilung erkennen, der häufigste Wert liegt bei ca. 150 kW. Hier kann man sehen, dass vereinzelt auch ungewöhnlich hohe Werte erfasst werden.

Um einen Einblick in die Unterschiede in den Kurvenstrukturen zu erhalten, werden in Abbildung 4.1 die Verlaufskurven der einzelnen Tage als Liniendiagramme dargestellt. Die X-Achse stellt die einzelnen Tageszeitpunkte dar, die Y-Achse die zugehörigen Messwerte. Jede dieser Linien stellt die Messungen eines Kalendertages dar. Hier kann man erkennen, dass je nach Tag unterschiedliche Typen von Verbrauchsmustern auftreten. Mit freiem Auge sind grob zwei unterschiedliche Strukturen zu erkennen: Einerseits gibt es Tage, an welchen der Verbrauch am Anfang des Tages stark ansteigt, dann auf einem gewissen Level schwankt und anschließend mit Ende des Tages wieder stark abfällt. Andererseits gibt es auch Tage, an welchen der Verbrauch konstant niedrig, um die 100 kW bleibt. Es lässt vermuten, dass es sich bei ersterem um Werktage und bei letzterem um Wochenenden oder Feiertage (diese werden in weiterer Folge als "Ruhetage"bezeichnet) handelt. Auf Basis dieser Vermutung werden nun in den Abbildungen 4.3 und 4.5 die Kurven gruppiert nach Werktagen und Ruhetagen dargestellt. Hier ist klar erkennbar, dass Ruhetage eine unterschiedliche Struktur im Vergleich zu den Werktagen haben. Einige der Verläufe an den Werktagen ähneln jenen der Ruhetage, was vermuten lässt, dass es sich hierbei um betriebsspezifische Ruhetage handelt, bzw. um Tage, an welchen die Firma ihre Produktion teilweise oder komplett heruntergefahren hat.

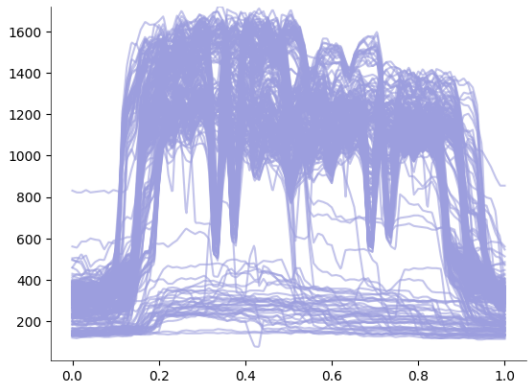

Abbildung 4.3: Verlaufskurven der Werktage

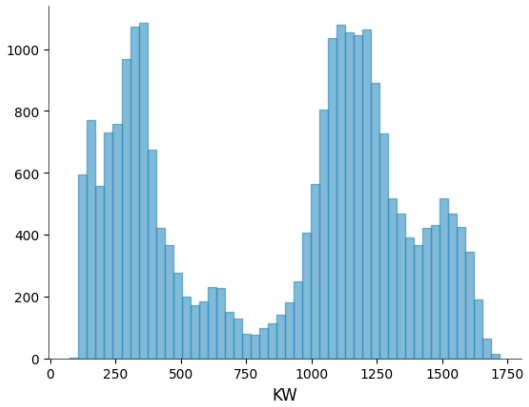

Abbildung 4.4: Wochentage

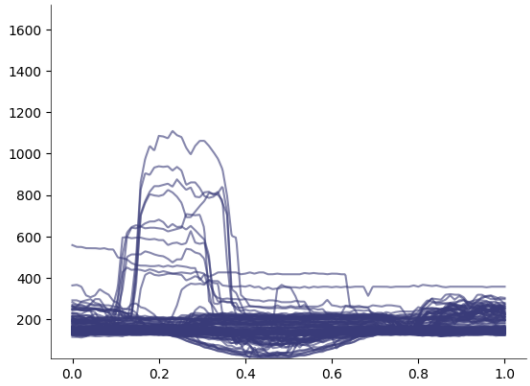

Abbildung 4.5: Verlaufskurven der Ruhetage

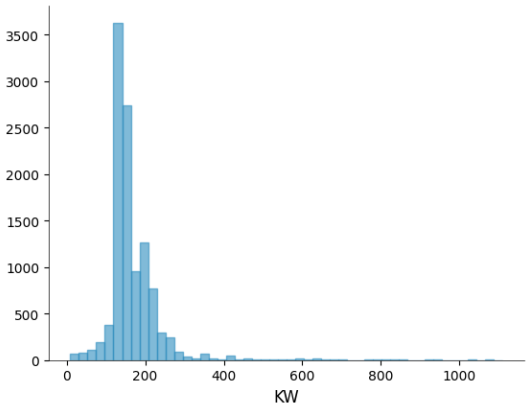

Abbildung 4.6: Wochenende & Feiertage

Eine weitere Frage, die sich stellt, ist, wie und ob sich der Gesamtverbrauch eines Tages im Laufe des Kalenderjahres verändert. Um hier einen groben Überblick zu bekommen, werden zunächst die Messungen eines Tages aufsummiert und diese anschließend in einem Streudiagramm visualisiert. Abbildung 4.7 zeigt die Werte farblich nach Wochentag aufgeschlüsselt, Abbildung 4.8 unterscheidet farblich, ob es sich um einen Werk- oder Ruhetag handelt.

Hier lassen sich grob zwei Gruppierungen erkennen. Die Einteilung in Werk- und Ruhetag bietet, was sich auch aus den Verbrauchskurven vermuten lässt, auf den ersten Blick eine grob passende Gruppierung der Werte. Die Tage mit sehr niedrigem Gesamtverbrauch verhalten sich im Verlauf des Jahres einigermaßen konstant. Bei jenen mit höherem Verbrauch, welche als Werk- bzw. Betriebstage angenommen werden, ist ein sinuskurvenartiger Verlauf zu beobachten, welcher in der ersten Jahreshälfte ansteigt, ca. im April/Mai seinen Höhepunkt erreicht und

ab dann zu sinken beginnt, bis er im Oktober seinen Tiefpunkt erreicht und wieder zu steigen beginnt.

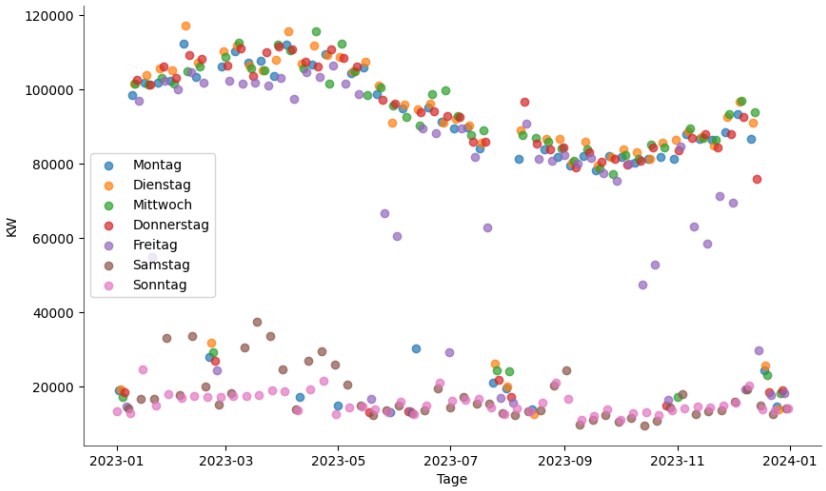

Abbildung 4.7: Tagessummen nach Wochentag

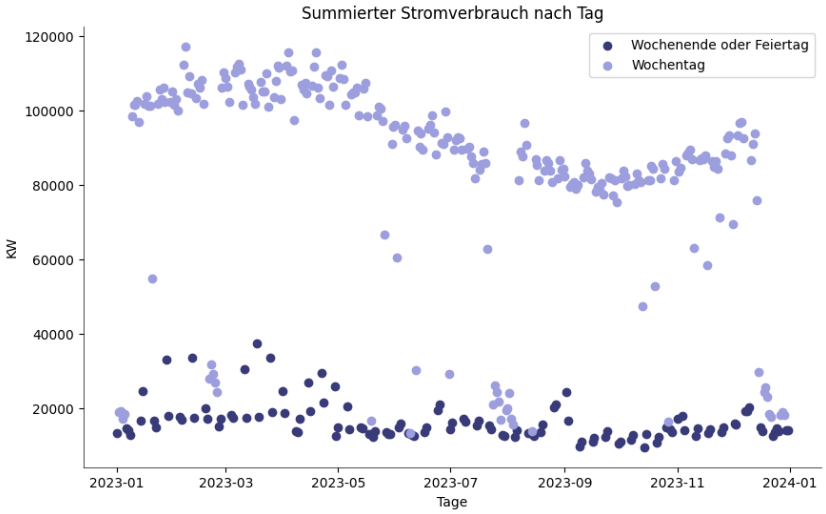

Abbildung 4.8: Tagessummen Werk- und Ruhetage

4.1.1 Funktionale Hauptkomponentenanalyse der Daten

Um einen genaueren Einblick in die Struktur der Daten zu erhalten, werden in weiterer Folge die Ergebnisse einer funktionalen PCA (sieh Kapitel 2.2.5) diskutiert. Diese wurde mit den diskreten Daten ohne vorherige Basistransformation durchgeführt. Abbildung 4.9 zeigt die ersten zwei

Hauptkomponenten (PC1 und PC2). Diese erklären in Summe über 95 % der Varianz in den Daten, die PC1 alleine 91,3 % und PC2 3,92 %.

Abbildung 4.10 zeigt die Mittelwertsfunktion der Daten und den Effekt, welcher durch Addieren und Subtrahieren eines geeigneten Vielfachen der jeweiligen fPC-Scores entsteht. Diese Visualisierung kann helfen, diese Effekte näher zu betrachten und zu interpretieren (vgl. [10], S 93). PC1 scheint in dieser Darstellung die Schwankungen innerhalb der Lastspitzen der Tage zu beschreiben und PC2 jene der Einbrüche im Laufe des Tages.

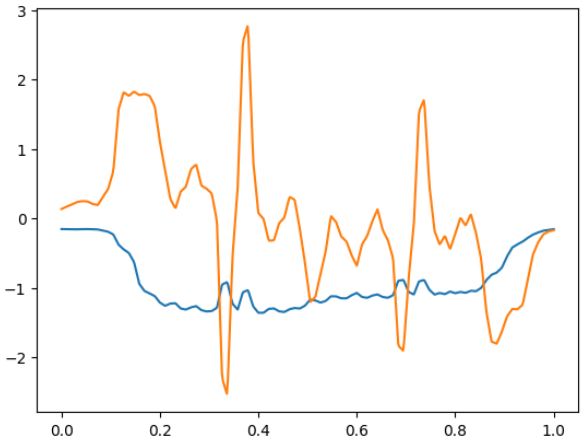

Abbildung 4.9: PC1 und PC2

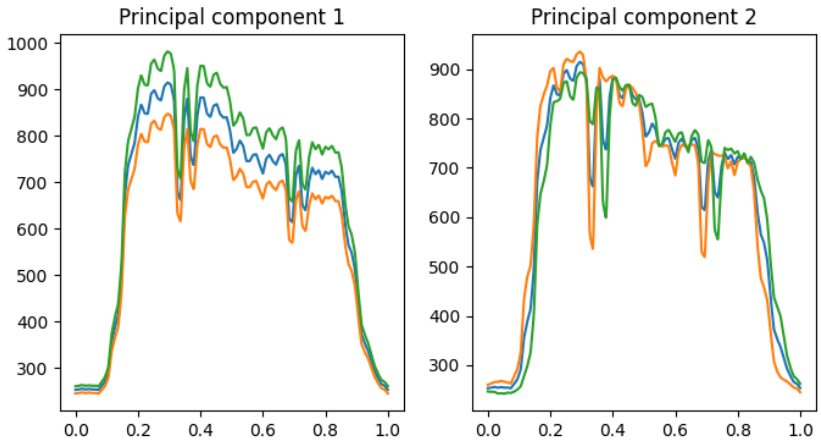

Abbildung 4.10: PC1 und PC2, Variation um die Mittelwertfunktion

Abbildung 4.11 zeigt einen Scatterplot der fPC-Scores der ersten zwei Hauptkomponenten. Jeder Punkt repräsentiert einen Tag und jede Achse des Diagramms die jeweiligen PC-Scores.

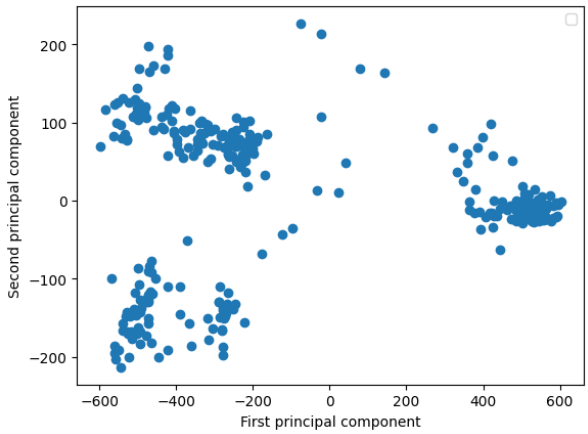

Abbildung 4.11: Scatterplot der fPC-Scores

Anhand dieser Visualisierung lassen sich mehrere Gruppierungen in den Datenpunkten erkennen. In Abbildungen 4.12 und 4.13 wurde noch eine zusätzliche Dimension, Wochentage bzw. Werk- und Ruhetage, hinzugefügt. Bei Betrachtung der Wochentage lassen sich kaum offensichtliche Strukturen erkennen, jedoch bei Werk- und Ruhetagen fällt auf, dass Ruhetage allesamt Teil des Clusters ganz rechts auf der X-Achse sind.

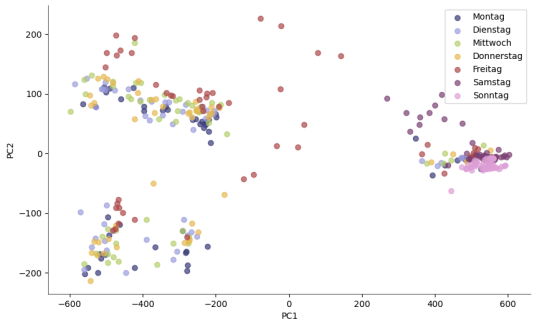

Abbildung 4.12: fPC-Scores nach Wochentagen

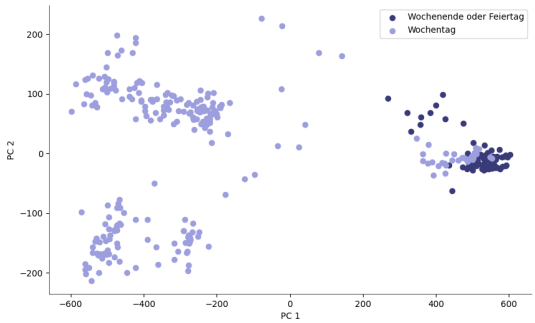

Abbildung 4.13: fPC-Scores nach Ruhe- und Werktagen

4.2 Clustering der Daten

Eine explorative erste Analyse der Daten sowie die Ergebnisse der Simulationsstudie legen nahe, dass eine Betrachtung der Daten in zuvor definierten Gruppen ein sinnvoller erster Schritt für die weitere Vorgehensweise und Erkennung von abnormalen Verbrauchsmustern ist. Da die Verbrauchsmuster der einzelnen Tage sehr heterogen sind, ist die Annahme, dass ein Weglassen dieses Schrittes zu einer weniger gezielten Erkennung von Anomalien führt.

Zusätzlich haben Erfahrungswerte der Firma *nista* gezeigt, dass ein vorheriges Clustering hilfreich für weitere Analysen ist. Analog zu bestehenden Ansätzen der Firma wird nun in einem ersten Schritt ein Clustering der Tage mittels k-Means durchgeführt und die Ergebnisse dieses Schrittes hier diskutiert und visualisiert. Als Verfahren für das Clustering werden zwei verschiedene Ansätze für k-Means-Algorithmen angewendet. Der erste ist ein klassischer multivariater k-Means-Algorithmus (Ansatz 1), analog zur bisherigen Vorgehensweise der Firma. Die zweite Methode stellt eine Verfeinerung der ersten dar und wendet den k-Means-Algorithmus auf die ersten zwei Hauptkomponenten der in Abschnitt 4.1.1 diskutierten fPCA an.

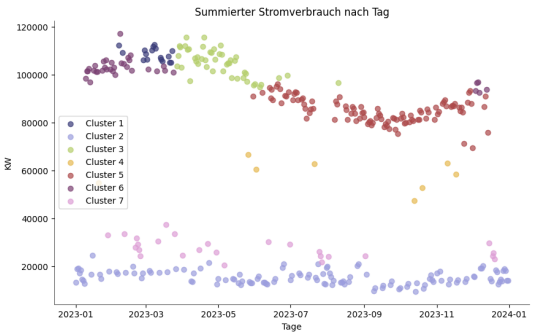

Abbildung 4.14: Tagessummen nach Cluster: Ansatz 1

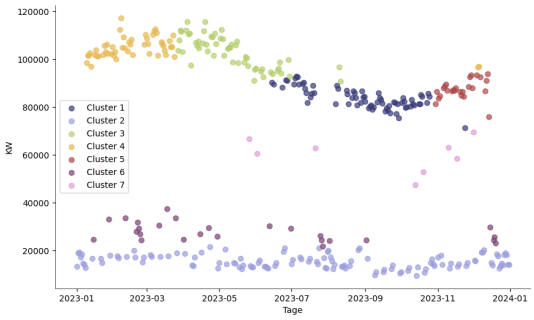

Abbildung 4.15: Tagessummen nach Cluster: Ansatz 2

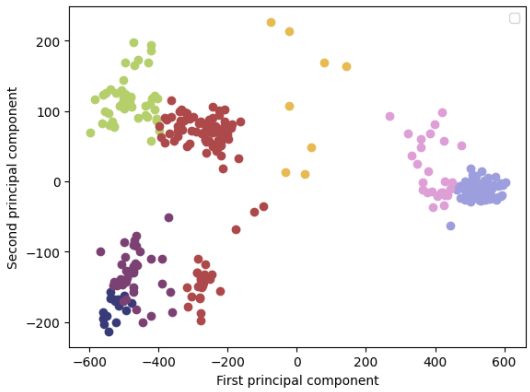

Abbildung 4.16: PC1 & PC2 nach Cluster: Ansatz 1

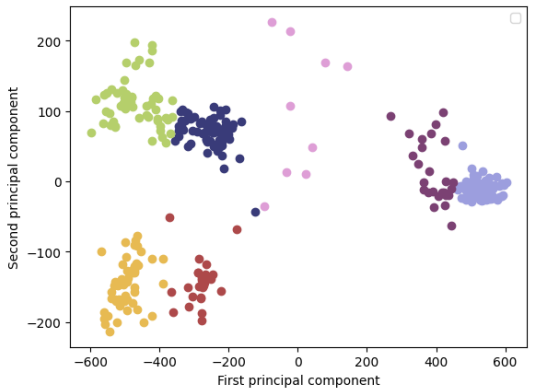

Abbildung 4.17: PC1 & PC2 nach Cluster: Ansatz 2

Die “optimale” Anzahl an Clustern k wurde durch die Methode von *nista* (siehe Kapitel 3.1) als $k = 7$ festgelegt und wird in weiterer Folge für beide Ansätze angewandt. Abbildungen 4.14 - 4.17 stellen den Datensatz reduziert auf zwei Dimensionen, einerseits durch Bildung der Summen der einzelnen Tage, andererseits durch die ersten zwei fPC-Scores, dar und bilden die Clus-

ter durch unterschiedliche Farben ab. Hier zeigen sich einerseits viele Ähnlichkeiten, teilweise fast identische Cluster, als auch geringfügige Unterschiede in den Gruppierungen.

Verlausfkurven der einzelnen Cluster nach Ansatz 1

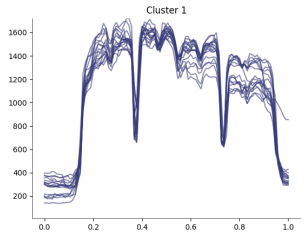

Abbildung 4.18: Verlaufskurven Cluster 1

Abbildung 4.19: Verlaufskurven Cluster 2

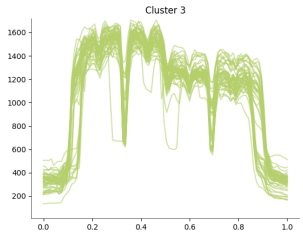

Abbildung 4.20: Verlaufskurven Cluster 3

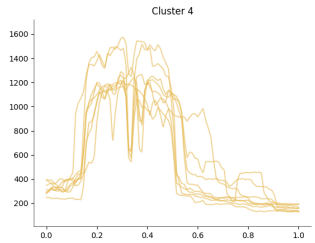

Abbildung 4.21: Verlaufskurven Cluster 4

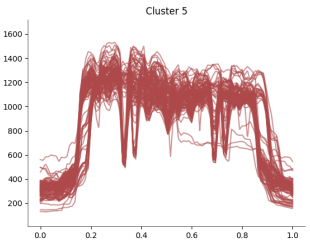

Abbildung 4.22: Verlaufskurven Cluster 5

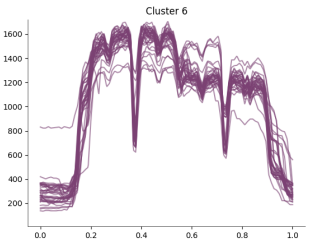

Abbildung 4.23: Verlaufskurven Cluster 6

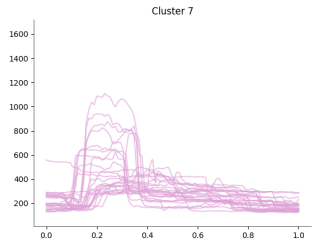

Abbildung 4.24: Verlaufskurven Cluster 7

Verlausfkurven der einzelnen Cluster nach Ansatz 2

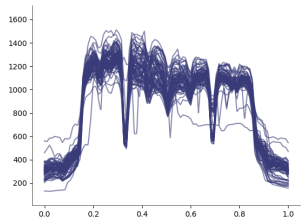

Abbildung 4.25: Verlaufskurven Cluster 1

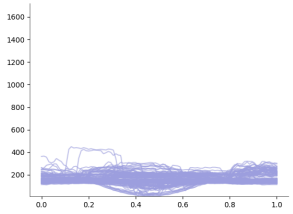

Abbildung 4.26: Verlaufskurven Cluster 2

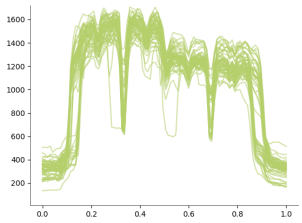

Abbildung 4.27: Verlaufskurven Cluster 3

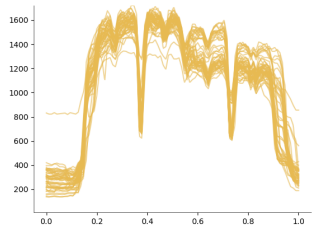

Abbildung 4.28: Verlaufskurven Cluster 4

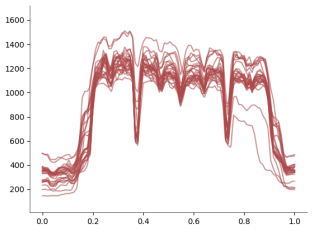

Abbildung 4.29: Verlaufskurven Cluster 5

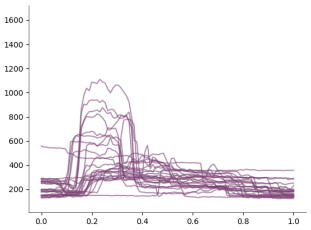

Abbildung 4.30: Verlaufskurven Cluster 6

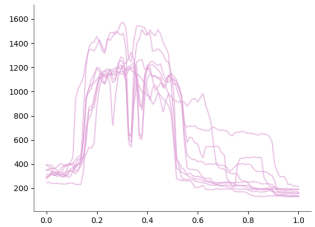

Abbildung 4.31: Verlaufskurven Cluster 7

4.3 Clusterweise Anomalieerkennung mittels *TVD* und *MSS*

Nachdem in Schritt 1 (4.2) die Cluster ermittelt wurden, erfolgt im nächsten Schritt innerhalb der einzelnen Cluster die Erkennung von Form- und Größenausreißern mittels *TVD* und *MSS*. In jedem Cluster wurden die empirischen Faktoren mit einem sehr sensiblen, minimalen Wert

festgesetzt, um eine maximale Zahl an erkannten Anomalien zu erzielen. In weiterer Folge wurde deren Relevanz von Expert:innen von *nista* evaluiert.

In diesem Kapitel werden die Ergebnisse der zwei verschiedenen Ansätze sowie etwaige Besonderheiten und Auffälligkeiten in einzelnen Clustern diskutiert und visualisiert. Kapitel 4.4 fasst die Ergebnisse zusammen und zieht einen Vergleich zwischen den zwei Clustering-Ansätzen.

Clustering-Ansatz 1: Multivariates Clustering mit k-Means

Auffälligkeiten beim ersten Ansatz sind einerseits die in Cluster 5 erkannten Formausreißer, sowie die geringe Clustergröße in Cluster 4, welche analog zu Cluster 7 bei Ansatz 2 auftritt, in welchem die Hälfte der Kurven als Größenausreißer identifiziert werden.

Im 5. Cluster wurden 20 Formausreißer identifiziert, welche durch *nista* als nicht relevant eingestuft wurden. Bei diesen handelt es sich laut dem Unternehmen um eine Umstellung im Schichtbetrieb, welche sich durch eine Verschiebung der "normalen" Verbrauchskurve entlang der Zeitachse äußert. Dies ist zwar ein "untypisches" Verhalten im Vergleich zu den restlichen Kurven innerhalb des Clusters und entspricht demnach auch der theoretischen Definition einer Formanomalie, ist jedoch für den konkreten Anwendungsfall nicht relevant. Zusätzlich ist noch anzumerken, dass dieses Muster zwar von dem Rest der Kurven abweicht, aber trotzdem vergleichsweise häufig auftritt. Dieser Fall illustriert das in der Einleitung erwähnte Spannungsfeld zwischen der Häufigkeit einer abnormalen Beobachtung und der Wirksamkeit von unsupervised-Learning-Algorithmen. Im zweiten Ansatz fallen diese Kurven in eine andere Gruppierung und diese Problematik tritt nicht auf, was im folgenden Abschnitt 4.3 betrachtet wird.

Zeitweise treten starke Schwankungen in den Daten auf, d.h. der Stromverbrauch steigt oder sinkt stark innerhalb kurzer Zeit. In Cluster 3 passiert dies beispielsweise am Anfang und am Ende des Tages. Variiert diese Steigung innerhalb eines Clusters nun stärker, so kann dies dazu führen, dass die Central-Region in diesem Bereich verhältnismäßig groß ausfällt (siehe z.B. Abbildung 4.40) und der funktionale Boxplot in diesen Bereichen weniger sensibel auf Abweichungen reagiert. In diesem Fall sollte die MSS etwaige Ungenauigkeiten ausgleichen, da die Gewichtsfunktion so gewählt wurde, dass diese Bereiche einen stärkeren Einfluss haben (siehe Kapitel 2.2.2).

Die maximale Hülle ist symmetrisch um die Central-Region definiert, was im Anwendungsfall teilweise kritisch betrachtet werden kann. Beispielsweise in Cluster 2 und 7 umfasst diese einen Bereich mit negativen Werten, welche im Anwendungsfall nicht vorkommen können.

Was zusätzlich zu beobachten ist, sind die unterschiedlichen Wertebereiche der MSS je nach Cluster. In den Clustern 3,5 und 6 sind diese beispielsweise durchwegs sehr niedrig, mit einem Median bei ca. 0.3, während sich dieser bei den restlichen Clustern bei ca. $0.6 - 0.7$ befindet.

Cluster 1

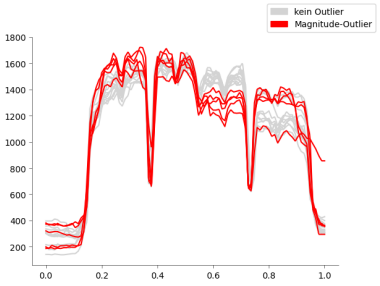

Abbildung 4.32: Ausreißer in Cluster 1

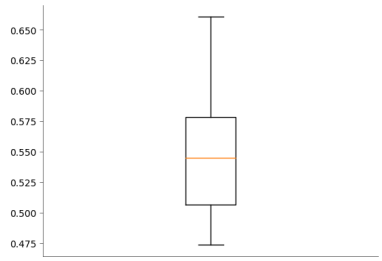

Abbildung 4.33: Boxplot der *MSS* in Cluster 1

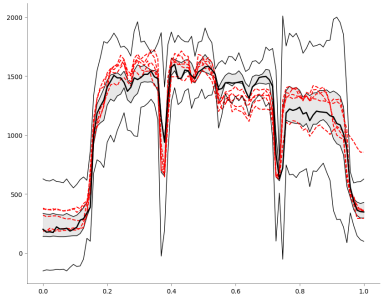

Abbildung 4.34: Funktionaler Boxplot von Cluster 1

Cluster 2

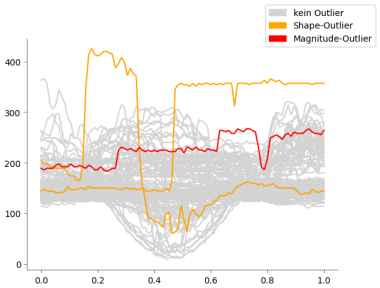

Abbildung 4.35: Ausreißer in Cluster 2

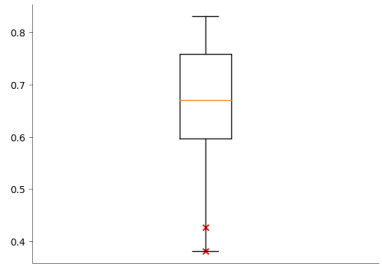

Abbildung 4.36: Boxplot der *MSS* in Cluster 2

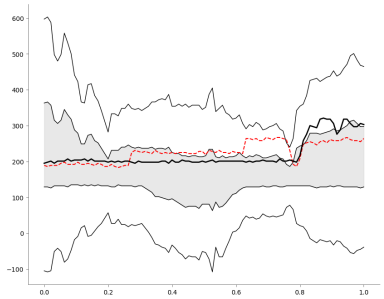

Abbildung 4.37: Funktionaler Boxplot von Cluster 2

Cluster 3

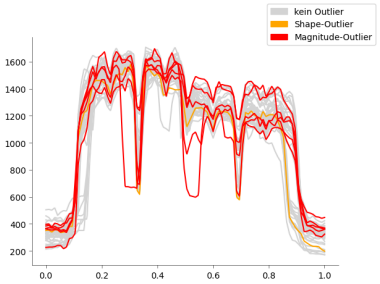

Abbildung 4.38: Ausreißer in Cluster 3

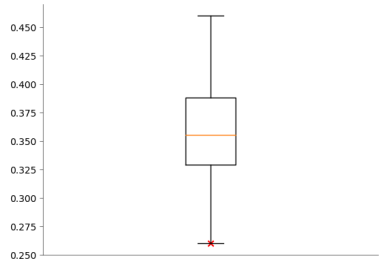

Abbildung 4.39: Boxplot der *MSS* in Cluster 3

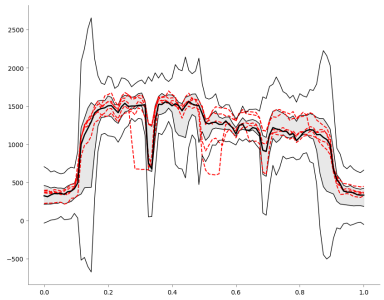

Abbildung 4.40: Funktionaler Boxplot von Cluster 3

Cluster 4

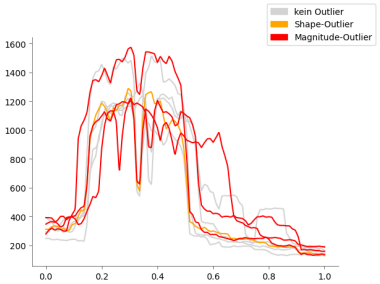

Abbildung 4.41: Ausreißer in Cluster 4

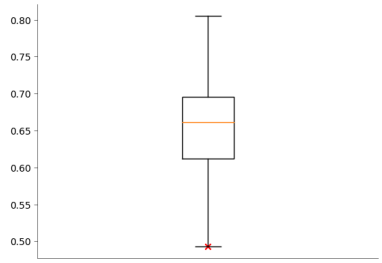

Abbildung 4.42: Boxplot der *MSS* in Cluster 4

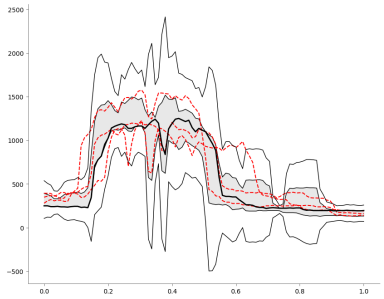

Abbildung 4.43: Funktionaler Boxplot von Cluster 4

Cluster 5

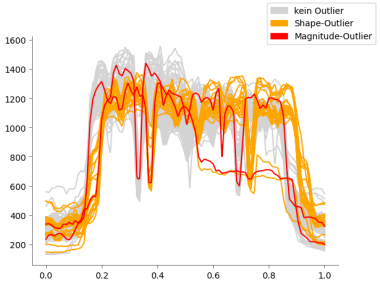

Abbildung 4.44: Ausreißer in Cluster 5

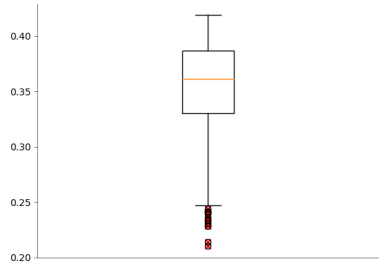

Abbildung 4.45: Boxplot der *MSS* in Cluster 5

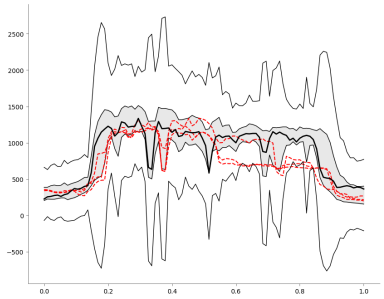

Abbildung 4.46: Funktionaler Boxplot von Cluster 5

Cluster 6

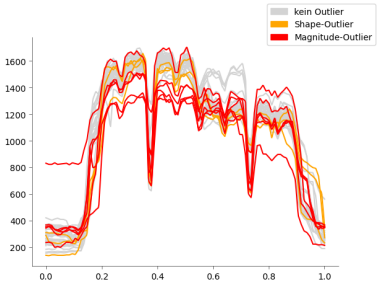

Abbildung 4.47: Ausreißer in Cluster 6

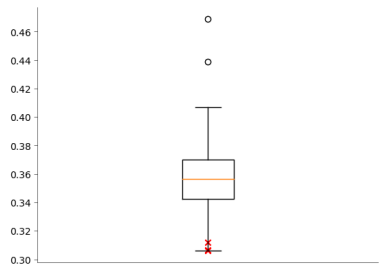

Abbildung 4.48: Boxplot der *MSS* in Cluster 6

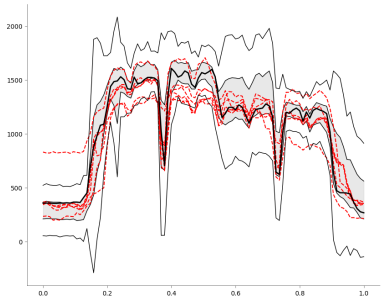

Abbildung 4.49: Funktionaler Boxplot von Cluster 6

Cluster 7

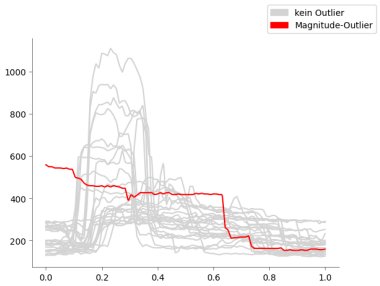

Abbildung 4.50: Ausreißer in Cluster 7

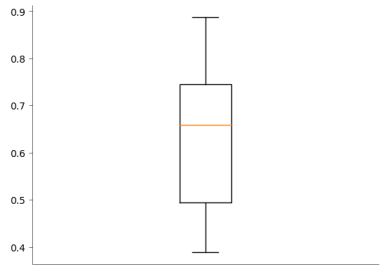

Abbildung 4.51: Boxplot der *MSS* in Cluster 7

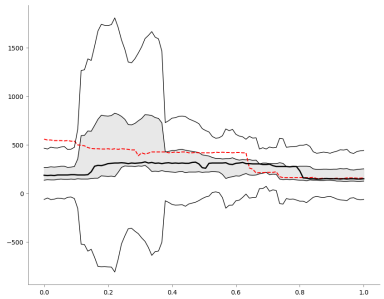

Abbildung 4.52: Funktionaler Boxplot von Cluster 7

Clustering-Ansatz 2: k-Means-Clustering der fPC-Scores

Ähnlich wie im ersten Ansatz, weist ein Teil der Cluster durchwegs niedrigere Werte der MSS auf. So liegt der Median dieser in den Clustern 1,3,4 und 5 unter jenen der übrigen.

In Cluster 7, welcher beinahe identisch zu Cluster 4 von Ansatz 1 ist, tritt dieselbe Problematik einer zu geringen Clustergröße auf, wie bereits im vorhergehenden Abschnitt 4.3 beschrieben. Auch die starken Schwankungen der maximalen Hülle können hier beobachtet werden.

Die Kurven, welche unter Ansatz 1 Cluster 5 bildeten, sind in diesem Ansatz anders aufgeteilt. Der im 1. Ansatz als Formanomalie detektierte Wechsel im Schichtbetrieb ist hier Teil eines anderen Clusters (siehe Abbildung 4.17 und 4.29) und wird in diesem nicht als Formanomalie erkannt. Dies spricht dafür, dass im gegebenen Anwendungsfall Clustering-Ansatz 2 zu einer besseren Gruppierung der Daten führt.

Cluster 1

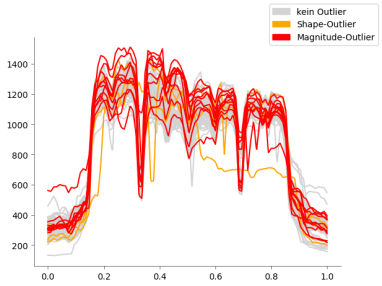

Abbildung 4.53: Ausreißer in Cluster 1

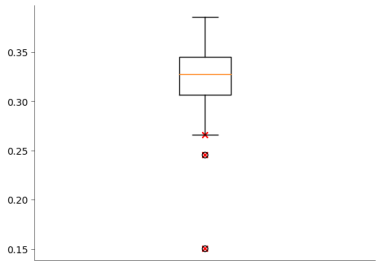

Abbildung 4.54: Boxplot der *MSS* in Cluster 1

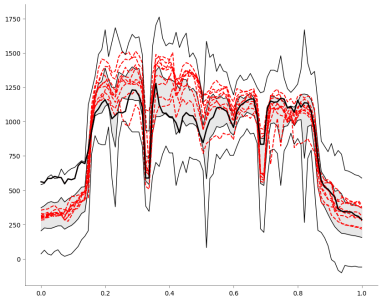

Abbildung 4.55: Funktionaler Boxplot von Cluster 1

Cluster 2

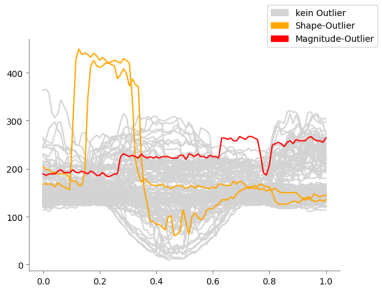

Abbildung 4.56: Ausreißer in Cluster 2

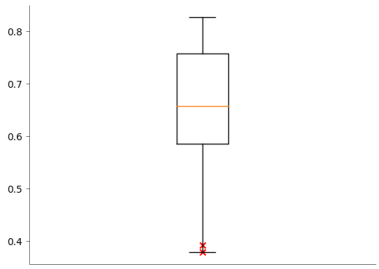

Abbildung 4.57: Boxplot der *MSS* in Cluster 2

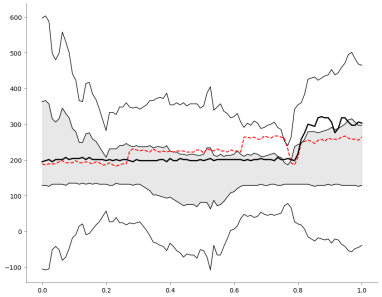

Abbildung 4.58: Funktionaler Boxplot von Cluster 2

Cluster 3

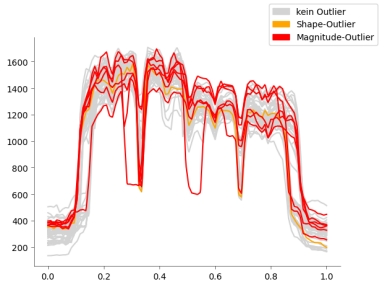

Abbildung 4.59: Ausreißer in Cluster 3

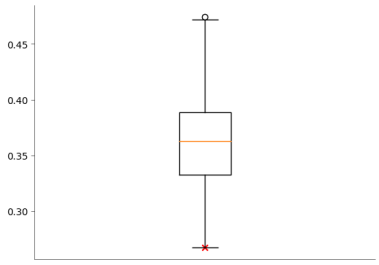

Abbildung 4.60: Boxplot der *MSS* in Cluster 3

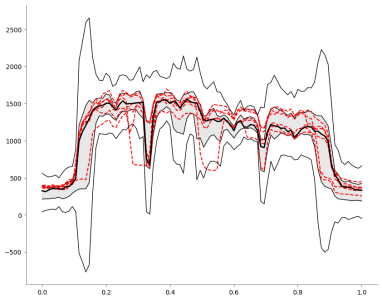

Abbildung 4.61: Funktionaler Boxplot von Cluster 3

Cluster 4

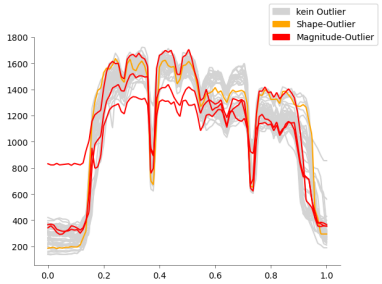

Abbildung 4.62: Ausreißer in Cluster 4

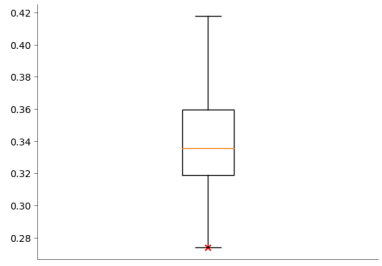

Abbildung 4.63: Boxplot der *MSS* in Cluster 4

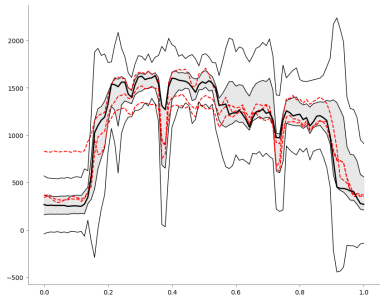

Abbildung 4.64: Funktionaler Boxplot von Cluster 4

Cluster 5

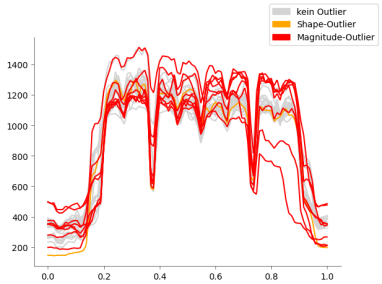

Abbildung 4.65: Ausreißer in Cluster 5

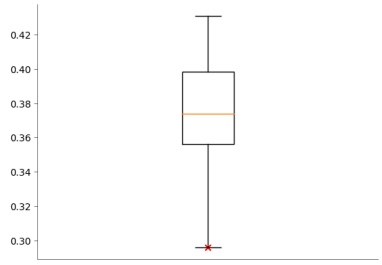

Abbildung 4.66: Boxplot der *MSS* in Cluster 5

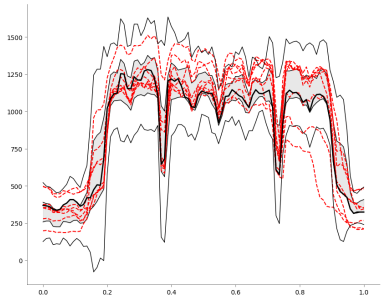

Abbildung 4.67: Funktionaler Boxplot von Cluster 5

Cluster 6

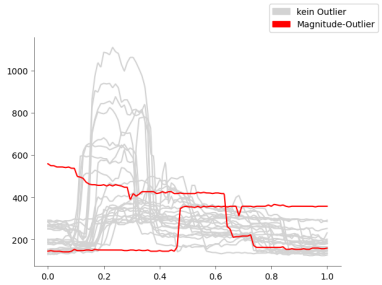

Abbildung 4.68: Ausreißer in Cluster 6

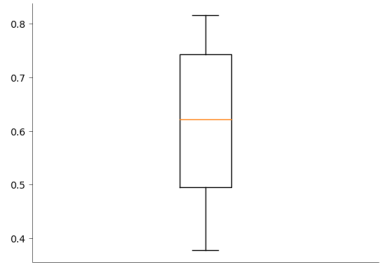

Abbildung 4.69: Boxplot der *MSS* in Cluster 6

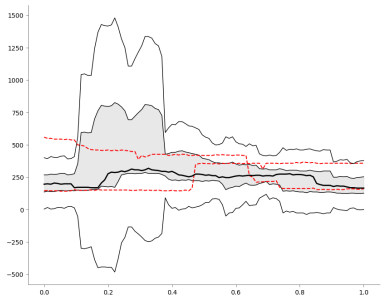

Abbildung 4.70: Funktionaler Boxplot von Cluster 6

Cluster 7

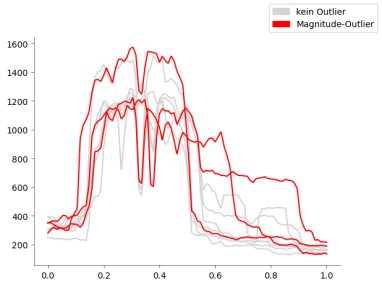

Abbildung 4.71: Ausreißer in Cluster 7

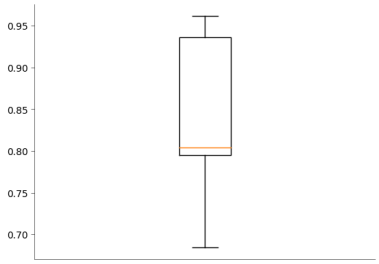

Abbildung 4.72: Boxplot der *MSS* in Cluster 7

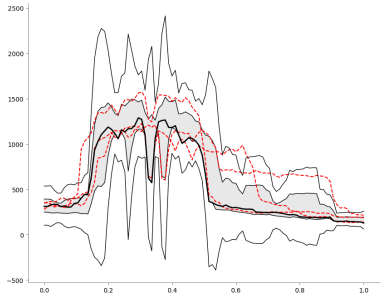

Abbildung 4.73: Funktionaler Boxplot von Cluster 7

4.4 Ergebnisse

In diesem Kapitel werden die Ergebnisse der Analyse zusammengefasst und diskutiert. Tabelle 4.1 zeigt die Summe der erkannten Ausreißer im Anwendungsfall, unter Verwendung von Ansatz 1 und Tabelle 4.2 jene unter Ansatz 2. Diese sind jeweils sowohl nach Cluster als auch zusätzlich nach Art der Ausreißer gegliedert. Was hier bei diesen Ergebnissen zu beachten ist, ist dass diese Auswertung ohne eine Optimierung der Parameter F_{MSS} und F_{TVD} erfolgt ist. Die Evaluierung der Ausreißer erfolgte im Kontext des Clusters, d.h. den Expert:innen von *nista* wurden die potenziellen Ausreißerkurven innerhalb der zuvor erkannten Cluster präsentiert.

In Summe wurden mit Clustering-Ansatz 1 im vorhandenen Datensatz 54 potenzielle Ausreißer erkannt, davon 31 Form- und 23 Größenausreißer. Von der Firma wurden unter diesen 22 als relevant klassifiziert, 7 der Form- und 15 der Größenausreißer.

Mit dem zweiten Clustering-Ansatz wurden in Summe 41 potenzielle Anomalien erkannt, 8 Form- und 33 Größenausreißer, von welchen insgesamt 28 relevant für den Anwendungsfall sind. Es wurden somit mehr relevante und weniger irrelevante Anomalien als durch den ersten Ansatz erkannt. Vor allem die Anzahl an falsch positiven Formausreißern konnte von 24 auf 4 reduziert werden. Insgesamt wurde die Anzahl an falsch positiven Anomalien von 32 auf 13 reduziert und die der “relevanten” Anomalien von 22 auf 28 erhöht, ohne die Parameter F_{MSS} und F_{TVD} zu optimieren.

Cluster	Ausreißer-Typ	Erkannte Ausreißer	Relevante Ausreißer
1	Shape Outlier	0	0
	Magnitude Outlier	5	3
	Gesamt	**5**	**3**
2	Shape Outlier	2	2
	Magnitude Outlier	1	1
	Gesamt	**3**	**3**
3	Shape Outlier	1	0
	Magnitude Outlier	5	4
	Gesamt	**6**	**4**
4	Shape Outlier	1	0
	Magnitude Outlier	3	0
	Gesamt	**4**	**0**
5	Shape Outlier	24	4
	Magnitude Outlier	2	1
	Gesamt	**26**	**5**
6	Shape Outlier	3	1
	Magnitude Outlier	6	5
	Gesamt	**9**	**6**
7	Shape Outlier	0	0
	Magnitude Outlier	1	1
	Gesamt	**1**	**1**
Summe	Shape Outlier	31	7
	Magnitude Outlier	23	15
	Gesamt	**54**	**22**

Tabelle 4.1: Ergebnisse Clustering-Ansatz 1

Cluster	Ausreißer-Typ	Erkannte Ausreißer	Relevante Ausreißer
1	Shape Outlier	3	2
	Magnitude Outlier	10	7
	Gesamt	**13**	**9**
2	Shape Outlier	2	2
	Magnitude Outlier	1	1
	Gesamt	**3**	**3**
3	Shape Outlier	1	0
	Magnitude Outlier	6	6
	Gesamt	**7**	**6**
4	Shape Outlier	1	0
	Magnitude Outlier	3	3
	Gesamt	**4**	**3**
5	Shape Outlier	1	0
	Magnitude Outlier	8	4
	Gesamt	**9**	**4**
6	Shape Outlier	0	0
	Magnitude Outlier	2	2
	Gesamt	**2**	**2**
7	Shape Outlier	0	0
	Magnitude Outlier	3	1
	Gesamt	**3**	**1**
Summe	Shape Outlier	8	4
	Magnitude Outlier	33	24
	Gesamt	**41**	**28**

Tabelle 4.2: Ergebnisse Clustering-Ansatz 2

5 Fazit

In dieser Arbeit wurde ein mehrstufiges Verfahren zur Ausreißererkennung, basierend auf einem k-Means-Clustering und einem robusten Detektionsverfahren von Form- und Größenausreißern aus dem Bereich der funktionalen Datenanalyse vorgestellt, dieses in *Python* implementiert, dessen Performance mittels einer Simulationsstudie analysiert und anschließend dessen Relevanz für den Anwendungsfall an Smart-Meter-Daten des Start-ups *nista* evaluiert.

Die Simulationsstudie verfolgte in erster Linie das Ziel, die Funktionalität und Wirksamkeit des nach Huang und Sun [7] implementierten *Python*-Algorithmus zu validieren und dessen Performance hinsichtlich der True- und False-Positive-Rate in Abhängigkeit der empirischen Faktoren F_{MSS} und F_{TVD} zu evaluieren. Zusätzlich wurde der Grad der Abweichung der Ausreißer vom simulierten Hauptmodell variiert und dessen Einfluss auf die Performance des Algorithmus analysiert. Zudem wurde der Mehrwert des Clustering-Schrittes für die gezielte Anomaliedetektion bei Datensätzen mit unterschiedlichen Gruppen von Verbrauchsmustern evaluiert. Letztendlich wurde die Performance der vorgestellten Methode mit einer "naiven" Methode, basierend auf einer Herangehensweise des Unternehmens *nista*, verglichen. Die Studie hat ergeben, dass die vorgestellte Methodik eine bessere Performance im Sinne einer höheren True-Positive-Rate und einer niedrigeren False-Positive-Rate als der "naive" Ansatz aufweisen konnte. Die Simulation von Daten aus zwei unterschiedlichen Hauptmodellen hat gezeigt, dass der vorgestellte Ansatz eine bessere Performance durch Hinzufügen des Clusteringschrittes erzielt, als wenn dieser nicht durchgeführt wird.

Im Anwendungsfall wurde die vorgestellte Methodik mit zwei verschiedenen Clusteringansätzen angewandt und die pro Ansatz erkannten potenziellen Anomalien von der Firma als relevant oder nicht relevant eingestuft. Durch den ersten, simpleren Clusteringansatz konnten 22 relevante und 32 irrelevante Anomalien erkannt werden. Der zweite, verfeinerte Ansatz detektierte 28 relevante und nur 13 irrelevante Ausreißer. Vor allem die Anzahl der irrelevanten Formausreißer konnte durch den verfeinerten Clusteringansatz reduziert werden. Dies führt zu der Schlussfolgerung, dass sich Ansatz 2 im Sinne einer höheren Zahl relevanter und einer niedrigeren Zahl irrelevanter besser für den Anwendungsfall eignet.

Nach Rückmeldung des Unternehmens bieten sowohl die erkannten Anomalien als auch die zur Verfügung gestellte Visualisierung dieser einen hilfreichen Input und vielversprechenden Ansatz zur Anomalieerkennung von Energieverbrauchsdaten und somit einen Mehrwert für das Unternehmen und dessen Kund:innen. Die Implementation in *Python* bietet außerdem die Möglichkeit einer schnellen und unkomplizierten Integration in die bestehende Infrastruktur. Was nach dieser Methodik unklar bleibt, ist die Anzahl der Anomalien, welche durch den Algorithmus nicht erkannt werden konnten. Die vorherige Klassifizierung dieser befand sich außerhalb des Rahmens dieser Arbeit.

6 Ausblick

Diese Arbeit hat gezeigt, dass die vorgestellte Methodik sowohl gezielt simulierte Anomalien erkennen als auch relevante Erkenntnisse für das Unternehmen *nista* im vorhandenen Datensatz liefern kann. Gleichzeitig hat diese Analyse einige Fragen aufgeworfen, Optimierungspotenziale aufgezeigt bzw. Impulse für weitergehende Forschung gegeben, welche in diesem Kapitel erörtert werden.

In erster Linie bieten Verfeinerungen der vorgestellten Methodik eine Vielzahl an Potenzialen. *Total Variation Depth* und *Modified Shape Similarity* sind beide mit einer Gewichtsfunktion definiert. Verfeinerungen dieser könnten im Bezug auf den Anwendungsfall weiter untersucht werden. Weitere Möglichkeiten bietet die Optimierung der empirischen Faktoren im Anwendungsfall, was auch aus einem wirtschaftlichen Standpunkt unternehmensseitig relevant ist. Zusätzlich können alternative Verfahren aus der funktionalen Datenanalyse zur Erkennung von Form- und Größenausreißern, wie z.B. der Magnitude-Shape-Plot [2] oder dem Outliergramm [1] getestet und deren Performance sowie Vor- und Nachteile verglichen werden. Dasselbe gilt für alternative Clusteringsverfahren bzw. eine Vertiefung der Clusteranalyse.

Bei der durchgeführten Analyse war die Anzahl der nicht erkannten Anomalien nicht bekannt. Ein vorhergehendes manuelles Labeln der Datensätze und eine anschließende Analyse dieser kann zusätzliche Möglichkeiten zur Forschung bieten. Einerseits könnte die vorgestellte Methodik für den Anwendungsfall genauer evaluiert werden. Andererseits würde sich dadurch auch der Raum an möglichen Methoden erhöhen, da nun in diesem Fall detaillierteres Wissen über normale und abnormale Verbrauchsmuster vorhanden wäre.

Wie in Kapitel 2 erwähnt, ist der “Concept-Drift”, also die Veränderung der Datengrundlage mit der Zeit, eine der Problemstellungen in der Analyse von Stromdaten. Daher könnte die Analyse dieser Methodik über einen längeren Zeitraum ebenfalls ein interessanter Forschungsgegenstand sein. Ein weiterer Ansatz wäre die Einbindung von Nutzer:innen, bzw. Feedback von Kund:innen in die Modellentwicklung — so können einerseits die Parameter des Modells kontinuierlich angepasst, und zusätzlich auch Feedback über die Nutzer:innenfreundlichkeit des

Modells erhalten werden.

Die Entwicklung eines standardisierten Produkts und die damit verbundene Optimierung der Nutzer:innenerfahrung bietet ebenfalls Möglichkeiten für weitere Forschung. Zusätzlich könnte auch die Nutzer:innenfreundlichkeit von der Analytiker:innenseite des Unternehmens evaluiert werden, d.h. wie hilfreich die einzelnen Visualisierungen, wie z.B. der Boxplot oder die Verteilung der Schätzwerte für die *MSS* und der Funktionale Boxplot aus dieser Perspektive sind. Bei diesem Ansatz könnte auch die Entwicklung eines interaktiven Analysetools im Vordergrund stehen. Hier könnte man zusätzlich die Frage stellen, inwieweit Nutzer:innen in diesen Prozess involviert werden können bzw. sollen, und ob das Wissen über die Hintergründe der Anomalieerkennung einen Mehrwert für Kund:innen bietet. Es könnte z.B. erhoben werden, inwiefern ein Wissen über die Methodik und die Visualisierung der Ergebnisse das Vertrauen in die Relevanz der Analysen seitens der Kund:innen stärkt.

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[1] Ana Arribas-Gil and Juan Romo. Shape outlier detection and visualization for functional data: the outliergram. *Biostatistics*, 15(4):603–619, October 2014.

[2] Wenlin Dai and Marc G. Genton. Multivariate Functional Data Visualization and Outlier Detection. *Journal of Computational and Graphical Statistics*, 27(4):923–934, October 2018. Publisher: ASA Website.

[3] Wenlin Dai and Marc G. Genton. Directional outlyingness for multivariate functional data. *Computational Statistics Data Analysis*, 131:50–65, 2019. High-dimensional and functional data analysis.

[4] Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006. ROC Analysis in Pattern Recognition.

[5] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[6] Yassine Himeur, Khalida Ghanem, Abdullah Alsalemi, Faycal Bensaali, and Abbes Amira. Artificial intelligence based anomaly detection of energy consumption in buildings: A review, current trends and new perspectives. *Applied Energy*, 287:116601, April 2021.

[7] Huang Huang and Ying Sun. A Decomposition of Total Variation Depth for Understanding Functional Outliers. *Technometrics*, 61(4):445–458, October 2019. Publisher: ASA Website.

[8] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[9] Sara López-Pintado and Juan Romo. On the Concept of Depth for Functional Data. *Jour-*

nal of the American Statistical Association, 104(486):718–734, June 2009. Publisher: ASA Website.

[10] James O Ramsay. *Functional data analysis*. Springer series in statistics. Springer, New York, NY [u.a.], 1997.

[11] Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. Anomaly detection in time series: a comprehensive evaluation. *Proc. VLDB Endow.*, 15(9):1779–1797, May 2022. Publisher: VLDB Endowment.

[12] Ying Sun and Marc G. Genton. Functional Boxplots. *Journal of Computational and Graphical Statistics*, 20(2):316–334, January 2011. Publisher: ASA Website.

[13] Romain Tavenard, Johann Faouzi, Gilles Vandewiele, Felix Divo, Guillaume Androz, Chester Holtz, Marie Payne, Roman Yurchak, Marc Rußwurm, Kushal Kolar, and Eli Woods. Tslearn, a machine learning toolkit for time series data. *Journal of Machine Learning Research*, 21(118):1–6, 2020.

[14] Jane-Ling Wang, Jeng-Min Chiou, and Hans-Georg Müller. Functional Data Analysis. *Annual Review of Statistics and Its Application*, 3(Volume 3, 2016):257–295, 2016. Publisher: Annual Reviews Type: Journal Article.