h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

Hochschule Darmstadt

– Fachbereich Informatik–

Konzeption und Evaluation eines Systems zur Outlier Detection für Web Analytics Daten im Kontext eines Wordpress Plugins

Abschlussarbeit zur Erlangung des akademischen Grades

Bachelor of Science (B.Sc.)

vorgelegt von

André Brücke

Matrikelnummer: *****

Referent	:	Prof. Dr. Gunter Grieser
Korreferent	:	Prof. Dr.-Ing. Michael von Rüden

André Brücke: *Konzeption und Evaluation eines Systems zur Outlier Detection für Web Analytics Daten im Kontext eines Wordpress Plugins,* © 02. März 2020

ERKLÄRUNG

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe.

Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht.

Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen.

Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Darmstadt, 02. März 2020

André Brücke

ABSTRACT

The design of an outlier detection system includes at least preprocessing of the input data, feature selection, choice of a suitable outlier detection algorithm and configuration of its parameters.

These steps are dependent on the application area, as well as the properties of the input data. For a new outlier detection system, an individual concept must therefore be created and evaluated. The main contribution of this thesis is the exemplary design of an outlier detection system for web analytics data, as well as the evaluation of the resulting concept. Another contribution is the development of a generator for synthetic data, which simulates the input data by creating transition matrices, and allows the controlled insertion of different types of outliers. It was shown that the basic statistical properties of the input data are being maintained. The generated datasets were used to support the productive web analytics data during design and evaluation. Empirical experiments during the design phase showed the Local Outlier Factors superior precision when compared to the k-Nearest-Neighbor algorithm and the Histogram Based Outlier Score. During the evaluation a weakness of the system was revealed, which occurred during failures lasting several weeks, and significantly decreased the systems accuracy. Such failures have to be detected and labelled when preprocessing the datasets. With this taken into account, the overall system achieved an improvement in accuracy on all productive datasets, when compared to the standard configuration. For two out of three datasets, the probability of a randomly selected outlier being assigned a higher outlier score than a normal data point, is greater than 99%. For the third data set, this probability is above 92%. The process used to create the system design can be applied to systems of other application domains as well.

ZUSAMMENFASSUNG

Der Entwurf eines Systems zur Outlier Detection umfasst mindestens die Vorverarbeitung der Eingabedaten, die Selektion der Features, die Wahl eines geeigneten Outlier Detection Algorithmus und die Konfiguration der Parameter dieses Algorithmus. Diese Schritte hängen nicht nur vom Anwendungsgebiet, sondern auch von den Eigenschaften der Eingabedaten ab. Für ein neuartiges System muss daher ein individuelles Konzept erstellt und bewertet werden. Der Hauptbeitrag dieser Arbeit ist die beispielhafte Konzeption eines Outlier Detection Systems für Web Analytics Daten, sowie die Evaluation des entstandenen Konzepts. Ein weiterer Beitrag ist die Entwicklung eines Generators für synthetische Daten, der die Eingabedaten mithilfe einer Übergangsmatrix nachbildet, und das gezielte Einfügen verschiedener Arten von Outliern ermöglicht. Es wurde gezeigt, dass die grundlegenden statistischen Eigenschaften der Eingabedaten beibehalten werden. Die generierten Datensätze wurden zur Ergänzung der produktiven Web Analytics Daten bei der Konzeption und Evaluation eingesetzt. In den empirischen Experimenten im Rahmen der Konzeption setzte sich der Local Outlier Factor gegen den k-Nearest-Neighbor Algorithmus und den Histogram Based Outlier Score durch. Die Evaluation konnte eine Schwachstelle des Systems aufdecken, die bei mehrwöchigen Ausfällen auftrat und die Genauigkeit des Systems signifikant verschlechterte. Derartige Ausfälle müssen bereits bei der Vorverarbeitung der Datensätze erkannt und markiert werden. Wenn dies berücksichtigt wurde, erzielte das Gesamtsystem auf allen produktiven Datensätzen eine Verbesserung der Genauigkeit im Vergleich zur standardmäßigen Konfiguration. Für zwei der drei Datensätze ist die Wahrscheinlichkeit, dass ein zufällig gewählter Outlier einen höheren Outlier Score zugewiesen bekommt als ein normaler Datenpunkt, größer als 99%. Auf dem dritten Datensatz beträgt diese Wahrscheinlich über 92%. Das Vorgehen beim Entwurf des Systems kann auch auf Systeme außerhalb des Web Analytics Bereichs übertragen werden.

INHALTSVERZEICHNIS

ABBILDUNGSVERZEICHNIS

TABELLENVERZEICHNIS

1 EINLEITUNG

1.1 PROBLEMSTELLUNG UND MOTIVATION

Ein zentrales Problem bei der Analyse von Daten aus realen Systemen ist der Umgang mit Verunreinigungen und unerwarteten Werten. Solche unerwarteten Einträge werden als Outlier bezeichnet.

Am genauesten können Outlier von Personen mit Expertenwissen auf dem jeweiligen Gebiet identifiziert werden, da diese beispielsweise Hintergrundinformationen nutzen und Zusammenhänge erkennen können. Je größer ein Datensatz jedoch wird, und je häufiger er um neue Einträge ergänzt wird, desto weniger praktikabel ist eine solche manuelle Identifikation.

Die automatische Erkennung von Outliern ist daher für verschiedene Anwendungsgebiete interessant und wird aktiv erforscht. So wird Outlier Detection unter anderem dazu genutzt, unauthorisierte Zugriffe auf Netzwerke zu erkennen, Sensordaten zu validieren und in medizinischen Anwendungen die Diagnose von Krankheiten zu unterstützen [23].

Auch im Web Analytics Bereich sind in den Daten Verunreinigungen von DoS Angriffen, Systemtests, oder Robots und Webcrawlern zu erwarten. Soll auf Basis von Web Analytics Daten der Einfluss von Marketing Kampagnen auf eine Unternehmens-Website abgeschätzt werden, sind nur die Einträge realer Besucher von Interesse, während andere Einträge zu Fehleinschätzungen führen können. Angriffe auf die Website sind dagegen aus sicherheitstechnischer Perspektive interessant. In beiden Fällen müssen Outlier vom Normalverhalten unterschieden werden. Da anzunehmen ist, dass sich Verunreinigungen signifikant von den Einträgen regulärer Besucher unterscheiden, können hierfür Methoden der Outlier Detection eingesetzt werden.

Die Entwicklung eines Systems zur Outlier Detection bringt eine Reihe von Entscheidungen mit sich. In der Regel ist mindestens zu entscheiden, welche Daten relevant sind, wie die Daten vorverarbeitet werden, welcher Algorithmus zur Erkennung eingesetzt wird und wie dessen Eingabeparameter gewählt werden. Diese Entscheidungen sind stark von den vorliegenden Datensätzen und dem jeweiligen Anwendungsgebiet abhängig und können nur bedingt verallgemeinert werden. Um ein neues System zur Erkennung von Outliern in Web Analytics Daten zu implementieren, muss daher im Vorfeld ein Konzept entwickelt werden, das anhand von konkreten produktiven Daten evaluiert werden kann.

1.2 ZIELSETZUNG

Im Rahmen der Arbeit soll das Konzept eines Systems entstehen, das Outlier in Web Analytics Daten zuverlässig und automatisiert erkennt. Das System soll bei der OdiSys GmbH zur Datenanalyse in einem Data Warehouse eingesetzt werden und die Daten für die Online Marketing Erfolgskontrolle des Unternehmens vorbereiten. Um das Risiko zu minimieren, reale Besucher irrtümlich zu entfernen oder Outlier nicht als solche zu erkennen, ist eine möglichst hohe Genauigkeit des Systems gewünscht. Außerdem soll das System, nachdem es in Betrieb genommen wurde, so wenig Konfigurationsaufwand wie möglich erfordern.

Der Fokus dieser Arbeit liegt ausschließlich auf der Identifikation der Outlier. Die Interpretation der erkannten Outlier ist genau wie die Bereinigung der Daten kein Bestandteil der Arbeit.

1.3 AUFBAU DER ARBEIT

Die Arbeit gliedert sich in sieben Kapitel. Während dieses Kapitel eine Einführung in die Grundthematik der Arbeit bietet, führt Kapitel 2 wichtige Begriffe und theoretische Grundlagen ein, gefolgt von einem Überblick über Outlier Detection Algorithmen und Metriken zur Bewertung der Genauigkeit dieser Algorithmen. Kapitel 3 stellt die Erkenntnisse verwandter Arbeiten vor.

In Kapitel 4 wird beschrieben, wie die Datensätze für die Konzeption und Evaluation der Algorithmen ausgewählt und vorverarbeitet wurden. Es wird ein Generator für synthetische Daten vorgestellt, der reale Eingabedaten nachbildet und gezielt Outlier einfügen kann. Kapitel 5 behandelt die Konzeption des Systems anhand von empirischen Experimenten. Das entstandene Konzept eines Gesamtsystems wird in Kapitel 6 evaluiert. Schließlich werden die Erkenntnisse in Kapitel 7 zusammengefasst und ein Ausblick auf die zukünftige Arbeit gegeben.

2 GRUNDLAGEN

2.1 ALLGEMEINE BEGRIFFE UND KONZEPTE

2.1.1 *Outlier und Noise*

Der Begriff Outlier ist in der Literatur nicht eindeutig bestimmt. Aggarwal und Yu definieren Outlier als Datenpunkte, die sich basierend auf einem bestimmten Maß signifikant vom Rest der Daten unterscheiden [2]. Hawkins definiert Outlier als Beobachtungen, die so stark von anderen Beobachtungen abweichen, dass der Verdacht besteht, dass sie durch einen anderen Mechanismus erzeugt wurden [16].

Ab wann eine Abweichung als signifikant betrachtet wird, ist vom Anwendungsgebiet abhängig und kann nicht allgemeingültig definiert werden.

Noise ist ein Phänomen in den Daten, das für den Analysten nicht von Interesse ist, sondern die Datenanalyse behindert [8]. Ob Datenpunkte als Noise betrachtet werden, hängt demnach vom Ziel der Datenanalyse ab. Möchte man beispielsweise die Reichweite einer Website anhand von Besucherstatistiken analysieren, können DoS Angriffe als Noise klassifiziert werden. Möchte man dagegen die Sicherheit dieser Website einschätzen, sind DoS Angriffe Gegenstand der Analyse und somit kein Noise.

2.1.2 *Dimensionalität*

Ein Datensatz besteht aus Datenpunkten, auch Einträge oder Instanzen genannt. Jeder Datenpunkt kann wiederum ein oder mehrere Features bzw. Attribute haben. Daten mit mehreren Features werden auch als multivariat bezeichnet, während Daten mit nur einem Feature univariat genannt werden [8]. Einen multivariaten Datensatz kann man sich als n-dimensionalen Raum vorstellen, welcher durch die Features aufgespannt wird. Die einzelnen Datenpunkte liegen dann innerhalb dieses Raumes.

2.1.3 *Cluster*

Clustering bezeichnet die Gruppierung eines Datensatzes in Kategorien, genannt Cluster. Während verschiedene Definitionen für den Begriff Cluster existieren, werden Cluster häufig über ihre innere Homogenität und externe Separierung definiert [24]. Also sind sich zwei Datenpunkte möglichst ähnlich, wenn sie sich innerhalb desselben Clusters befinden, und unähnlich, wenn sie sich in verschiedenen Clustern befinden.

Zwischen Clustering und Outlier Detection besteht eine komplementäre Beziehung, da Datenpunkte, die keinem Cluster zugeordnet werden können, vom Rest der Daten abweichen [1].

2.1.4 *Label*

Von einem gelabelten Datensatz spricht man, wenn die einzelnen Datenpunkte eines Datensatzes den zwei Klassen Outlier und Normalverhalten zugeordnet sind. Bei produktiven Daten muss diese Zuordnung manuell vorgenommen werden, indem Expertenwissen über die dahinter liegenden Prozesse ausgenutzt wird. Daher stellt die Verfügbarkeit von gelabelten Daten zum Training und zur Validierung von Outlier Detection Verfahren oft eine Herausforderung dar [8].

2.1.5 *Eigenschaften von Zeitreihen*

Zeitreihen sind eine spezielle Form der Daten, bei der die Datenpunkte eine zeitliche Abfolge darstellen. Eine Zeitreihe kann verschiedene Muster beinhalten, darunter Trend und Saison. Bei der Zerlegung von Zeitreihen werden diese Muster in eigene Komponenten aufgeteilt. Die Trendkomponente beinhaltet langfristige systematische Veränderungen im Niveau der Zeitreihe, während die Saisonkomponente zeitlich bedingte Schwankungen umfasst, die sich in regelmäßigen Zeitabständen mit ungefähr gleich bleibendem Muster wiederholen [9].

2.1.6 *Markov-Prozesse*

Ein Prozess ist ein Markov-Prozess, wenn die Auftrittswahrscheinlichkeit eines Zustands zu einem bestimmten Zeitpunkt aus Informationen über die vorherigen Zustände abgeleitet werden kann [21]. Eine Markov-Kette ist ein Modell zur Repräsentation eines solchen Prozesses, bei dem Zustandswechsel zu aufeinander folgenden Zeitschritten erfolgen [6].

Mit einer Übergangsmatrix wird für jede mögliche Kombination von Zuständen einer Markov-Kette beschrieben, wie wahrscheinlich ein Zustandswechsel von einem in den anderen Zustand ist.

Dabei bestimmt die Ordnung der Markov-Kette, wie viele Zustände in Kombination betrachtet werden, also von wie vielen vorherigen Zuständen ein Zustand abhängig ist. Bei einer Markov-Kette der ersten Ordnung ist jeder Zustand ausschließlich von seinem jeweiligen Vorgänger abhängig.

2.2 GRUNDLAGEN DER OUTLIER DETECTION

2.2.1 *Arten von Outliern*

Je nach Sicht auf den Datenraum können zwei verschiedene Arten von Outliern unterschieden werden. Globale Outlier sind Datenpunkte, die von allen Clustern signifikant abweichen.

Lokale Outlier treten bei globaler Betrachtung des Datenraums zwar in der Nähe eines Clusters auf, zeigen aber eine Abweichung, wenn sie relativ zu der entsprechenden Nachbarschaft betrachtet werden [5].

2.2.2 *Arten des Trainings*

Es existieren drei verschiedene Möglichkeiten für das Training von Outlier Detection Algorithmen. Beim Supervised Training liegen vollständig gelabelte Datensätze vor, die sowohl Outlier, als auch Normalverhalten enthalten. Aus diesen Daten werden in der Trainingsphase oft Prognosemodelle für die zwei Klassen erstellt, mit denen ungesehene Datenpunkte in der Testphase abgeglichen werden [8].

Beim Semi-Supervised Training liegt ein Datensatz vor, der das Normalverhalten repräsentiert. In der Trainingsphase wird entsprechend nur ein Prognosemodell für das Normalverhalten erstellt. Datenpunkte, die von diesem Modell abweichen, werden in der Testphase als Outlier gekennzeichnet.

Unsupervised Training benötigt keinerlei gelabelte Trainingsdaten. Stattdessen werden Outlier über Eigenschaften wie der Distanz zu anderen Datenpunkten oder der Auftrittshäufigkeit erkannt. Die Funktionsfähigkeit ist nur dann gewährleistet, wenn der Outlier-Anteil in den Daten deutlich geringer ist, als der Anteil normaler Datenpunkte [8]. Eine weitere Besonderheit ist, dass bei der Unsupervised Outlier Detection nicht zwischen Trainings- und Testdatensatz unterschieden wird [13].

2.2.3 *Arten der Ausgabe*

Für Outlier Detection Algorithmen sind zwei Arten der Ausgabe denkbar. Einige Algorithmen weisen den Datenpunkten direkt ein Label zu, das sie entweder als Outlier, oder als Normalverhalten kennzeichnet. Dieser Ansatz ist bei der Supervised Outlier Detection üblich, während Semi-Supervised und Unsupervised Outlier Detection Algorithmen den einzelnen Datenpunkten stattdessen meist einen Score zuweisen [13]. Dieser Score gibt an, mit welcher Wahrscheinlichkeit es sich bei dem Datenpunkt um einen Outlier handelt. Als Ausgabe für einen Datensatz ergibt sich somit ein Outlier Ranking, das eine Ordnung vom wahrscheinlichsten Outlier bis zum unwahrscheinlichsten Outlier angibt. Ein solches Outlier Ranking kann durch die Definition eines Schwellenwerts in Labels umgewandelt werden.

2.3 OUTLIER DETECTION ALGORITHMEN

2.3.1 *K-Nearest-Neighbor (k-NN)*

Die Erklärungen dieses Absatzes orientieren sich an der Arbeit von Ramaswamy et al. [19], in welcher der k-Nearest Neighbor Algorithmus vorgestellt wurde. k-NN ist ein Outlier Detection Algorithmus, der auf Distanzen zwischen Datenpunkten basiert. Da Outlier per Definition vom Rest der Daten abweichen, treten sie im Datensatz außerhalb von dichten Clustern auf. Somit ist der Abstand eines Datenpunktes von seinen nächsten im Datenraum angesiedelten Nachbarn ein geeignetes Maß, um zu bestimmen, mit welcher Wahrscheinlichkeit es sich dabei um einen Outlier handelt.

k-NN errechnet für jeden Datenpunkt eine Distanz zu seinem k-ten nächsten Nachbarn, wobei k der einzige Eingabeparameter des Algorithmus ist. Aus den ermittelten Werten kann ein Ranking von Datenpunkten nach absteigender Distanz erstellt werden. Je höher ein Datenpunkt im Ranking steht, desto wahrscheinlicher handelt es sich um einen Outlier.

Die einfachste Variante des Algorithmus berechnet für jeden Datenpunkt die Distanzen zu allen anderen Datenpunkten. Hierzu können verschiedene Distanz-Metriken für multidimensionale Räume eingesetzt werden. Da die auf verschachtelten Schleifen basierende Variante $O(n^2)$ Berechnungen durchführen muss, wurden verschiedene Optimierungen vorgestellt. Da die Maximierung der Performance nicht im Fokus der Arbeit steht, werden die Varianten an dieser Stelle nicht näher erläutert.

Eine von Angiulli et al. vorgeschlagene Variante des k-NN [3] berücksichtigt nicht nur die einzelne Distanz eines Datenpunktes zu seinem k-ten nächsten Nachbarn, sondern den Durchschnitt der Distanzen aller k nächsten Nachbarn. Hierzu kann entweder der Mittelwert, oder der Median der Distanzen verwendet werden.

2.3.2 *Local Outlier Factor (LOF)*

Die Erklärungen dieses Absatzes orientieren sich an der Arbeit von Breunig et al. [5], in welcher der Local Outlier Factor vorgestellt wurde. LOF berechnet für jeden Datenpunkt einen Outlier Score basierend auf der Dichte benachbarter Datenpunkte. Dadurch kann der Algorithmus auf Datensätzen mit Clustern unterschiedlicher Dichte und zur Erkennung von lokalen Outliern eingesetzt werden.

Der Algorithmus nutzt k-NN, um die *MinPts* nächsten Nachbarn eines Datenpunktes zu bestimmen, wobei *MinPts* der einzige Eingabeparameter des Algorithmus ist und die untere Schranke an Datenpunkten darstellt, die in eine Nachbarschaft miteinbezogen werden. Haben mehrere Datenpunkte dieselbe Distanz wie der k-te nächste Nachbar, werden diese zusätzlich in die Nachbarschaft aufgenommen.

Zur Berechnung des Local Outlier Factors müssen zunächst zwei Distanzen definiert werden. Die k-Distanz $k\text{-}dist(o)$ ist die Distanz eines Datenpunktes o zu seinem k-ten nächsten Nachbarn, die Reachability Distance $reach\text{-}dist_k(p, o)$ eines Datenpunktes p in Bezug auf den Datenpunkt o ist das Maximum aus der Distanz beider Punkte und der k-Distanz von o. Abbildung 2.1 veranschaulicht beide Distanzen für $k = 4$.

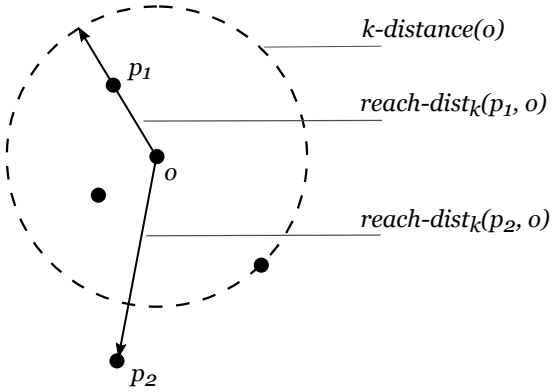


Abbildung 2.1: Veranschaulichung von k-Distanz und Reachability Distance für $k = 4$, nach Breunig et al. [5]

Die Reachability Distance wird zur Berechnung der Local Reachability Density (LRD) eingesetzt. Die LRD ist das Inverse des Durchschnitts aller Reachability Distances einer durch *MinPts* gegebenen Nachbarschaft.

Zur Berechnung des Local Outlier Factors eines Punktes p wird schließlich das Verhältnis aus der LRD von p und den LRDs der nächsten Nachbarn berechnet, und erneut der Durchschnitt für die Nachbarschaft gebildet. Dies sorgt dafür, dass Punkte mit dünn besiedelten Nachbarschaften, die sich in der Nähe von Punkten mit dicht besiedelten Nachbarschaften befinden, einen hohen LOF zugewiesen bekommen.

2.3.3 *Histogram Based Outlier Score (HBOS)*

HBOS ist ein statistisches Verfahren, das Outlier Scores mithilfe von Histogrammen berechnet. Die Erklärungen dieses Abschnitts sind angelehnt an die Arbeit von Goldstein et al. [12].

HBOS basiert auf der Annahme, dass Outlier in einem Datensatz mit deutlich geringerer Wahrscheinlichkeit auftreten, als Datenpunkte, die dem Normalverhalten entsprechen. Histogramme werden eingesetzt, um die Auftrittshäufigkeit der einzelnen Datenpunkte anzunähern. Bei multivariaten Eingabedaten wird die Unabhängigkeit der Features angenommen und jedes Feature einzeln betrachtet, was die Performance auf Kosten der Genauigkeit erhöht.

HBOS erstellt zunächst für jedes Feature ein Histogramm. Für numerische Features kann die Breite der Bins (Klassen) des Histogramms entweder statisch oder dynamisch bestimmt werden. Für die statischen Histogramme wird der Wertebereich eines Features in k Bins gleicher Breite eingeteilt, wo-

bei k der einzige Eingabeparameter des Algorithmus ist. Die Höhe eines Bins ist gegeben durch die Auftrittshäufigkeit der Datenpunkte, die in diesen Wertebereich fallen.

Abbildung 2.2 veranschaulicht die Probleme, die bei der Nutzung der statischen Breite entstehen können, anhand eines Beispiels aus dem Web Analytics Bereich. Durch hohe Besucherzahlen, die nur vereinzelt in den Daten auftreten, wird der Wertebereich gestreckt. So deckt jeder der acht Bins knapp 50 Werte ab. Im niedrigen Wertebereich, in den die große Mehrheit der Werte fallen, kann die Dichte nur noch schlecht angenähert werden.

Dieses Problem wird von der dynamischen Breite gelöst, indem nicht der Wertebereich, sondern die einzelnen Werte jedes Features aufgeteilt werden. Dazu werden die Werte sortiert und im Normalfall in k gleich große Bereiche aufgeteilt. Für den Sonderfall, dass ein Wert mehr als k-mal auftritt, kann der entsprechende Bereich jedoch größer als andere Bereiche sein. Die Breite der Bins ist durch den ersten und den letzten Wert des Bereichs gegeben.

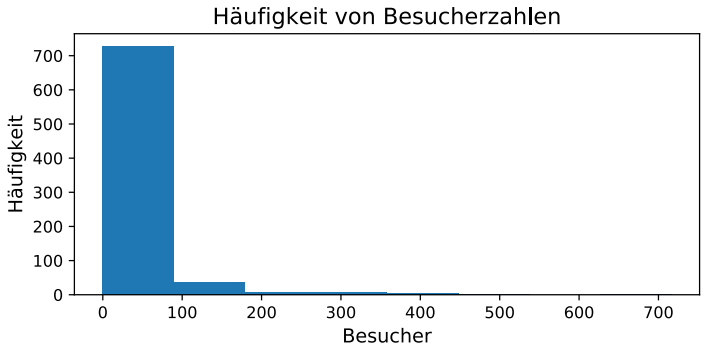


Abbildung 2.2: Veranschaulichung der Einschränkungen statischer Histogramme bei Wertebereichen mit großen Lücken, anhand eines Histogramms mit 8 Bins

Nachdem die erstellten Histogramme normalisiert wurden, sodass die maximale Höhe $1,0$ beträgt, gibt das Inverse dieser Höhe den Score an. Die Scores der einzelnen Features werden schließlich multipliziert, um den Gesamtscore eines Datenpunktes zu erhalten.

Goldstein et al. merken an, dass der von ihnen entwickelte Algorithmus globale Outlier verlässlich erkennen kann, aber schwach bei der Erkennung von lokalen Outliern abschneidet.

2.4 KATEGORISIERUNG DER BEHANDELTEN ALGORITHMEN

Die Kategorisierung von Outlier Detection Algorithmen kann je nach Quelle variieren. Die Entwickler des im praktischen Teil verwendeten Frameworks unterteilen die implementierten Algorithmen in die vier Kategorien Proximity, Linear Model, Ensembling und Neural Net [25]. Die nachfolgend beschriebenen Algorithmen fallen nach dieser Einteilung in die Kategorie Proximity.

Algorithmen dieser Kategorie definieren Outlier aufgrund ihrer Nähe oder Ähnlichkeit zu anderen Datenpunkten.

Goldstein et al. unterteilen die in ihrem Artikel verglichenen Verfahren nach deren grundlegendem Funktionsprinzip und der Art von Outliern, die erkannt werden. Abbildung 2.3 zeigt die Einordnung der in dieser Arbeit verwendeten Algorithmen in die Kategorien Goldsteins. k-NN ist demnach ein Algorithmus zur Erkennung globaler Outlier, während LOF ein Algorithmus zur Erkennung lokaler Outlier ist. HBOS wird als statistisches Verfahren kategorisiert.

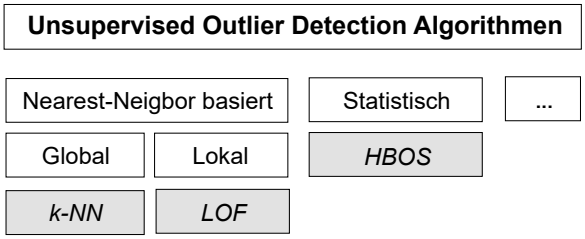


Abbildung 2.3: Kategorisierung der in dieser Arbeit verwendeten Outlier Detection Algorithmen, nach Goldstein et al. [13]

2.5 METRIKEN ZUR MESSUNG DER GENAUIGKEIT VON ALGORITHMEN

Zur Bewertung der Genauigkeit von Outlier Detection Algorithmen existieren verschiede Metriken, die in zwei Gruppen unterteilt werden können. Die erste Gruppe fasst Metriken zusammen, die keinen festen Schwellenwert zur Klasseneinteilung benötigen. Hierzu zählen unter anderem ROC Kurven (Receiver Operating Characteristic) und Precision-Recall-Kurven. Die zweite Gruppe kann dagegen lediglich eine binäre Klassifizierung in die Klassen Outlier und Normalverhalten bewerten. Hierunter fallen beispielsweise die True Positive Rate und die False Positive Rate.

Die meisten Metriken zur Bewertung der Genauigkeit basieren auf den in Abbildung 2.4 dargestellten Kennzahlen. Die in grün hervorgehobenen True Positives (TP) und True Negatives (TN) geben die Anzahl der korrekt klassifizierten Datenpunkte an, während die in rot hervorgehobenen False Positives (FP) und False Negatives (FN) die falsch vorhergesagten Datenpunkte darstellen.

False Positives sind in der Outlier Detection Datenpunkte, die vom Algorithmus als Outlier klassifiziert wurden, aber eigentlich keine Outlier darstellen. False Negatives sind hingegen Outlier, die vom Algorithmus nicht als solche identifiziert werden konnten. Je geringer diese beiden Kennzahlen sind, desto höher ist die Genauigkeit des Outlier Detection Algorithmus.

Aus den absoluten Kennzahlen kann man die oben erwähnten Raten berechnen. Die True Positive Rate (TPR) ist gegeben durch $\frac{TP}{TP+FN}$, und die False Positive Rate (FPR) durch $\frac{FP}{FP+TN}$.

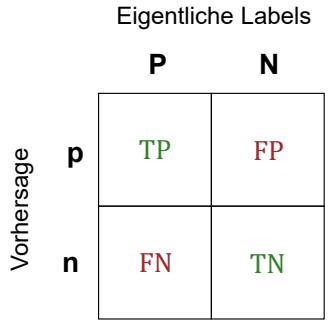


Abbildung 2.4: Confusion Matrix zur Darstellung von True Positives (TP), False Positives (FP), True Negatives (TN) und False Negatives (FN), nach Fawcett [10]

Die nachfolgenden Unterabschnitte stellen Metriken zur Bewertung von Outlier Rankings vor, die für die spätere Evaluation in Betracht gezogen werden.

2.5.1 *Precision at n*

Precision at n (P@n) ist eine einfache Metrik zur Bewertung von Rankings. Sie gibt die True Positives unter den n höchsten Scores eines Rankings an. Die Position der Punkte innerhalb dieser n Einträge, sowie der Abstand einzelner Scores, hat keine Auswirkung auf die P@n. Da zur Berechnung der P@n die Anzahl der True Positives unter den ersten n Rängen durch n dividiert wird, ist der bestmögliche P@n-Wert $1,0$ und der schlechteste Wert $0,0$.

2.5.2 *ROC Kurven und AUC Score*

ROC Kurven bilden die True Positive Rate in Abhängigkeit von der False Positive Rate ab. Eine ROC Kurve stellt somit den relativen Tradeoff zwischen Nutzen (True Positives) und Kosten (False Positives) dar [10]. Die optimale ROC Kurve führt durch die Punkte $(0 \mid 0)$, $(0 \mid 1)$ und $(1 \mid 1)$, also eine True Positive Rate von $1,0$ für alle Werte der False Positive Rate größer als 0.

ROC Kurven können zu einem einzelnen Wert, dem AUC Score, zusammengefasst werden. Dieser stellt die Fläche unter der ROC Kurve dar, beträgt also minimal $0,0$ und maximal $1,0$. Allerdings führt zufälliges Raten der Klassen zu einem AUC Score von $0,5$, weshalb kein Outlier Detection Algorithmus einen AUC Score von unter $0,5$ produzieren sollte [10].

2.5.3 *Precision-Recall Kurven und Average Precision*

Precision-Recall Kurven bilden die Precision in Abhängigkeit vom Recall ab. Die Precision $\left(\frac{TP}{TP+FP}\right)$ gibt an, wie viele der positiven Klassifizierungen korrekt sind, während der Recall $\left(\frac{TP}{TP+FN}\right)$ äquivalent zur True Positive Rate ist und das Verhältnis der korrekt erkannten Outlier angibt [22].

Die sogenannte Average Precision (AP) stellt die Fläche unter der Precision-Recall Kurve dar und hat äquivalent zum AUC Score einen Wertebereich von $0,0$ bis $1,0$.

Saito et al. merken an, dass zufälliges Raten bei Precision-Recall Kurven nur dann zu einer Average Precision von $0,5$ führt, wenn die positive und negative Klasse dieselbe Größe haben. Generell führt zufälliges Raten zu einer Average Precision, die durch das Verhältnis zwischen Outliern und normalen Datenpunkten $\left(\frac{P}{P+N}\right)$ gegeben ist [20].

3 VERWANDTE ARBEITEN

3.1 LITERATUR ZUR EVALUATION VON OUTLIER DETECTION ALGORITHMEN

In verschiedenen Arbeiten wurde sich mit der Frage beschäftigt, wie man die Ergebnisse von Outlier Detection Algorithmen bewerten und vergleichen kann.

Goldstein et al. evaluieren in einem 2016 veröffentlichten Artikel [13] 19 verschiedene Unsupervised Outlier Detection Algorithmen, darunter LOF, k-NN und HBOS. Ziel war es, mit einer universellen Evaluation auf öffentlich zugänglichen Datensätzen eine Basis für zukünftige Forschung auf diesem Gebiet zu bilden. Dazu wurden die Stärken und Schwächen der einzelnen Verfahren herausgearbeitet.

Zur Messung der Genauigkeit wurden AUC Scores verwendet. Der primäre Eingabeparameter jedes Algorithmus wurde über einen Wertebereich von 10 bis 50 variiert, und der Durchschnitt sowie die Standardabweichung der Ergebnisse festgehalten. Insgesamt schnitten lokale Algorithmen wie LOF schlecht auf Datensätzen mit globalen Outliern ab, während globale Algorithmen wie k-NN auf Datensätze mit lokalen Outliern mindestens durchschnittliche Ergebnisse erzielten. Goldstein et al. empfehlen den Einsatz von globalen Algorithmen, wenn die Art der Outlier eines Datensatzes nicht bekannt ist. Für solche Probleme wird k-NN im Durchschnitt als gute Wahl hervorgehoben, während für lokale Outlier Detection Probleme LOF als guter Kandidat genannt wird. HBOS zeigte die beste Performance für 4 der 10 Datensätze und wird zur Erkennung von globalen Outliern auf großen Datensätzen empfohlen.

In einer ebenfalls 2016 veröffentlichten Studie von Campos et al. [7] werden verschiedene Metriken und Datensätze zur Evaluation von Unsupervised Outlier Detection Algorithmen verglichen. 12 Algorithmen wurden auf mehreren Varianten der 23 gesammelten Datensätze ausgeführt. Die Genauigkeit wurde in erster Linie anhand des AUC Scores gemessen, während die Precision at n und Average Precision für zusätzliche Einblicke herangezogen wurden. Unter den Algorithmen, die mit ihrer besten Parameter-Konfiguration durchschnittlich über alle Datensätze die höchste Performance zeigten, waren k-NN und LOF. Die Autoren kommen zu dem Schluss, dass es nicht sinnvoll ist, die Überlegenheit eines Algorithmus für allgemeine Anwendungsfälle zu definieren. Algorithmen, die auf willkürlich gewählten Datensätzen nicht mit der Performance generischer Algorithmen mithalten können, können diese trotzdem in speziellen Domänen übertreffen.

3.2 LITERATUR ZUR GENERIERUNG VON ZEITREIHEN

Ein Großteil der Literatur zur Generierung von Zeitreihen befasst sich mit der Simulation von Windstärken und anderen natürlichen Vorgängen.

Shamshad et al. beschreiben den Einsatz von Markov-Ketten der ersten und zweiten Ordnung, um synthetische Zeitreihen von Windgeschwindigkeiten zu erzeugen [21]. Für zwei Datensätze wurden jeweils eine Übergangsmatrix der ersten und eine Übergangsmatrix der zweiten Ordnung erstellt. Zur Validierung der Ergebnisse wurden die statistischen Eigenschaften der generierten Zeitreihen mit denen der Originaldaten verglichen.

Während die meisten Eigenschaften erhalten blieben, konnten die Autoren keine signifikante Verbesserung des Modells zweiter Ordnung gegenüber dem Modell erster Ordnung beobachten. Beide Methoden scheiterten daran, das periodische Verhalten der beobachteten Windgeschwindigkeit beizubehalten. Die Autoren empfehlen daher, die saisonale Komponente der Eingabe-Zeitreihe vor dem Generierungsprozess zu entfernen.

Auch zur Modellierung des zeitlichen Verlaufs von Sonnenstrahlung wurden Markov-Ketten verwendet [18]. Ngoko et al. generierten Daten über einen Zeitraum von drei Jahren bei einminütigen Zeitintervallen. Als Eingabedaten dienten zwei Datensätze derselben Länge und Granularität. Der Wertebereich wurde in 100 Zustände unterteilt, und die Daten als Sequenzen dieser 100 Zustände betrachtet. Das spezielle Anwendungsgebiet erforderte es, jeweils eine Übergangsmatrix für verschiedene Wetterlagen zu erstellen. Aufgrund der statistischen Eigenschaften der Eingabedaten wurde ein Markov-Modell der zweiten Ordnung verwendet. Die synthetischen Daten wurden anschließend validiert, indem ihre statistischen Eigenschaften mit denen der Eingabedaten verglichen wurde. Besonders der Mittelwert, die Standardabweichung und die Verteilung der Werte konnten gut reproduziert werden.

4 AUSWAHL UND VORVERARBEITUNG DER DATENSÄTZE FÜR KONZEPTION UND EVALUATION

Bei der Konzeption werden empirische Experimente ausgeführt, um eine Reihe von Designentscheidungen begründet zu treffen. Die dazu benötigten Web Analytics Datensätze sollten sich von jenen unterscheiden, die zur Evaluation eingesetzt werden, um bei der Evaluation aussagekräftige Ergebnisse zu erhalten. Dieses Kapitel beschreibt die Auswahl und Vorverarbeitung der Datensätze und führt einen Generator für synthetische Daten ein.

Als produktive Daten stehen Statistiken von drei Unternehmens-Websites der OdiSys GmbH zur Verfügung. Der von der Hauptseite entnommene Datensatz wird im Folgenden als OWA-Datensatz (für OdiSys Web Analytics Datensatz) bezeichnet. Die anderen Datensätze stammen von zwei Produktseiten des Unternehmens, die mit der Hauptseite verlinkt sind. Sie werden nachfolgend als OWA-2-Datensatz bzw. OWA-3-Datensatz bezeichnet. Die im nächsten Abschnitt beschriebenen Schritte zur Vorverarbeitung wurden auf allen produktiven Datensätzen ausgeführt.

4.1 PRODUKTIVE DATENSÄTZE

4.1.1 *Rohdaten*

Die Websites der OdiSys GmbH werden über das Content Management System WordPress verwaltet. Das Plugin *WP Statistics* speichert die Besucher-Statistiken als einzelne Einträge ab. Aufrufe verschiedener URLs werden durch einen Zähler abgebildet, es entstehen also keine neuen Einträge. Die exakte Identität der Besucher kann vom Plugin allerdings nicht bestimmt werden. Stattdessen wird die Kombination aus IP-Adresse, User Agent und Plattform verwendet, um Besucher zu identifizieren. Diese Kombination ist für jeden Tag eindeutig. Zusätzlich enthalten die Rohdaten das mittels Geolocation aus der IP-Adresse abgeleitete Land, aus dem der Zugriff erfolgte.

Aus den Rohdaten wurden zunächst diejenigen Attribute entfernt, die keinen Mehrwert für die Outlier Detection bieten. Dazu zählen auch Attribute, die aufgrund von Einstellungen des Plugins nicht oder nur unvollständig erfasst wurden.

Zur Anonymisierung der Daten wurden die IP-Adressen der Besucher durch eine Klassifizierung als interne oder externe Besuche ersetzt [1]. Diese Information reicht für die Vorverarbeitung der Rohdaten aus.

1 Bekannte IP-Präfixe des Unternehmens wurden als intern klassifiziert, alle anderen Präfixe als extern.

4.1.2 *Reduktion von Noise*

Um die anonymisierten Rohdaten für die Anwendung der Outlier Detection Algorithmen vorzubereiten, wurden zwei Schritte zur Reduktion von Noise angewandt.

Zunächst wurden alle Einträge entfernt, die als intern klassifiziert wurden. Besucher, die dem Netzwerk des Unternehmens zuzuordnen sind, bieten keinen Mehrwert für die Datenanalyse und konnten daher ignoriert werden.

Weiterhin wurde eine Liste von User Agents angelegt, die eindeutig auf Robots oder Systemtests schließen lassen. Anhand dieser Liste konnte eindeutig identifizierbarer Noise herausgefiltert werden.

4.1.3 *Aggregierter Datensatz*

Die Rohdaten bestehen aus mehreren kategorischen Features und einem numerischen Feature. Da LOF und k-NN numerische Eingaben benötigen, müssten die kategorischen Features in numerische umgewandelt werden. Allerdings bieten die kategorischen Features nur dann einen tatsächlichen Mehrwert für die Outlier Detection, wenn ihre semantische Bedeutung erhalten bleibt. Für die gegebenen Websites mit deutschen Inhalten sollte beispielsweise die Distanz zwischen deutschsprachigen Ländern geringer sein, als die Distanz zu fremdsprachigen Ländern. Diese und ähnliche Abhängigkeiten müssten manuell abgebildet werden und können sich von Website zu Website unterscheiden. Ein weiteres Problem entsteht dadurch, dass ein Anteil der User Agents und Plattformen nicht identifiziert werden kann. In diesem Fall wird den Einträgen der Wert *Unknown* zugewiesen.

Aus diesen Gründen wurde sich dazu entschieden, die Rohdaten nach ihrem Datum zu aggregieren, und die Einzeldaten lediglich als unterstützende Hintergrundinformationen zu verwenden. Bei der Aggregation wurden für jeden Tag verschiedene numerische Features berechnet.

Bei Outliern in Web Analytics Daten ist zu erwarten, dass sie sich auf die täglichen Aufrufe und Besucherzahlen auswirken, weshalb beides als Feature aufgenommen wurde. Als zusätzlicher Anhaltspunkt dienen Veränderungen im Verhältnis zwischen Aufrufen und Besuchern. Hierzu wurde der Mittelwert der täglichen Aufrufe berechnet, da in diesen im Gegensatz zum Median extreme Werte einfließen. Der aggregierte Datensatz enthält also für jeden Tag die Summe der Aufrufe, die Anzahl der Besucher, sowie die durchschnittlichen Aufrufe pro Besucher.

4.1.4 *Interpretation der aggregierten Daten*

Abbildung 4.1a visualisiert die im OWA-Datensatz vorliegenden täglichen Aufrufe. Auffallend sind Extremstellen im Februar und April 2019, die von zwei Denial of Service (DoS) Angriffen erzeugt wurden. Der Mittelwert der Aufrufe für den Gesamtdatensatz beträgt 272, der Median dagegen nur 139. Das lässt darauf schließen, dass der Gesamtdurchschnitt durch die Extrem-

werte angehoben und somit verfälscht wird. Über den Zeitraum bis zum 01. Januar 2019 verteilt sind einzelne Tage erkennbar, deren Aufrufe signifikant vom Mittelwert abweichen.

Der in Abbildung 4.1b dargestellte zeitliche Verlauf der im OWA-2-Datensatz erfassten Aufrufe lässt Auswirkungen des DoS-Angriffs erkennen, die allerdings schwächer als auf dem OWA-Datensatz ausfallen. Zwischen Juli und September 2018 wurden über einen Zeitraum von mehreren Wochen hinweg keine Aufrufe erfasst, was als Ausfall zu interpretieren ist.

Auch für den in Abbildung 4.1c dargestellten OWA-3-Datensatz wurden zwischen Juli und September 2018 keine Aufrufe erfasst. Aus Gründen der Visualisierung zeigt die Zeitreihe nicht den April 2019, wo analog zum OWA-Datensatz an einem Tag über 14000 Aufrufe erreicht wurden.

In keinem der produktiven Datensätze ist ein signifikanter Trend oder eine Saisonalität erkennbar.

4.1.5 *Manuelle Klassifizierung*

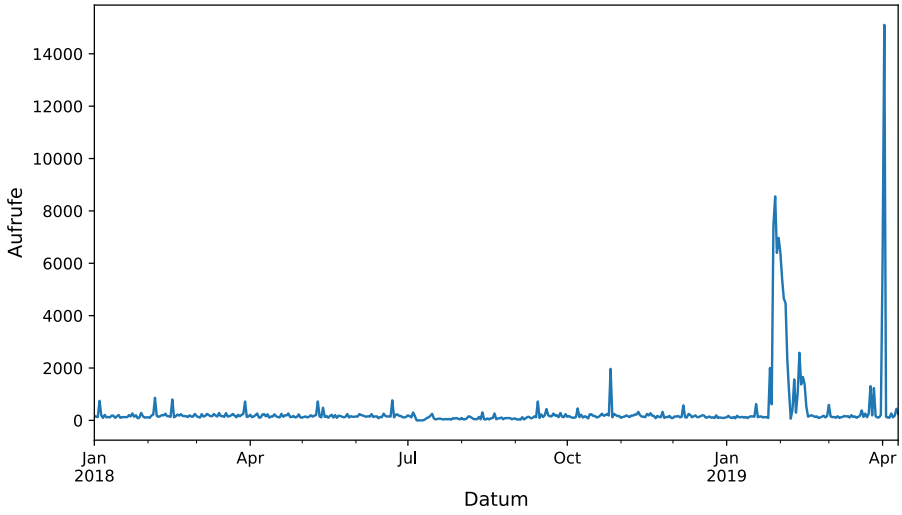
Da zur Evaluation der Outlier Detection Algorithmen gelabelte Daten benötigt werden, mussten die vorhandenen Daten unter Zuhilfenahme von Hintergrundinformationen interpretiert werden, sodass jeder Datenpunkt entweder der Klasse Outlier, oder der Klasse Normalverhalten zugeordnet werden kann.

Eine erste Einteilung wurde anhand der statistischen Eigenschaften der Zeitreihen vorgenommen. Einer häufig verwendeten Faustregel nach können Datenpunkte als Outlier gekennzeichnet werden, wenn sie drei oder mehr Standardabweichungen vom Mittelwert abweichen [23]. Diese Faustregel wurde auf den täglichen Aufrufen angewandt und durch detailliertere Betrachtung überprüft. Bei Grenzfällen wurden auch die Rohdaten hinzugezogen. Hier unterstützten die Herkunftsländer und Aufrufzahlen einzelner Besucher die korrekte Zuteilung. In einigen solcher Grenzfälle war es eine Frage der Interpretation, ob ein Datenpunkt einen Outlier darstellt, oder nicht. Deshalb kann nicht garantiert werden, dass alle Datenpunkte korrekt klassifiziert wurden.

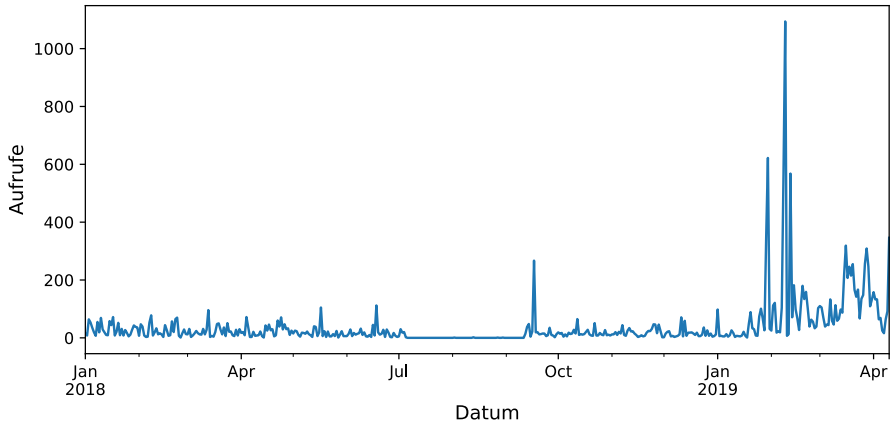
Ein Ausschnitt des gelabelten OWA-Datensatzes nach der manuellen Klassifizierung ist in Abbildung 4.2 zu sehen.

4.1.6 *Einschränkungen der produktiven Datensätze*

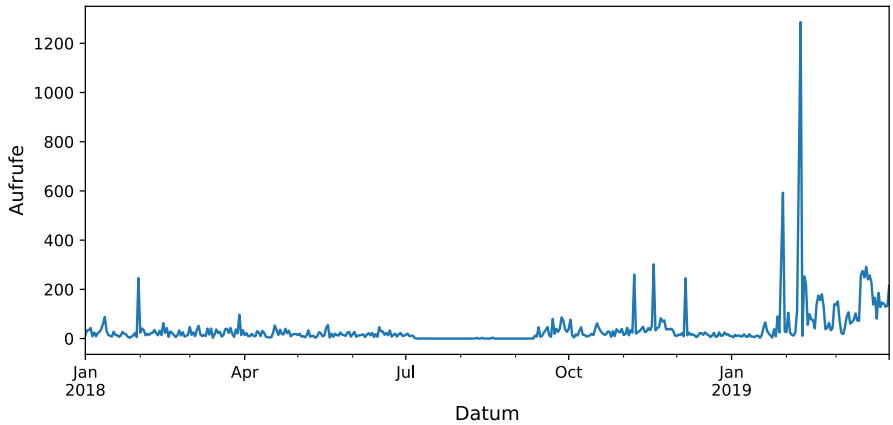
Die produktiven Datensätze bilden zwar die wichtigste Grundlage für die Konzeption und Evaluation, sind aber zur Tätigung von allgemeineren Aussagen über Web Analytics Daten unzulänglich, da sie einen speziellen Fall abbilden. Ähnlichkeit zwischen den Datensätzen besteht insofern, dass sie keinen signifikanten Trend beinhalten und sich die Outlier deutlich vom Rest der Daten abheben. Zur besseren Schätzung der Robustheit von Algorithmen ist es wünschenswert, auf Web Analytics Daten aus verschiedenen Quellen zurückgreifen zu können. In den folgenden Abschnitten werden



(a) OWA-Datensatz seit Januar 2018



(b) OWA-2-Datensatz seit Januar 2018



(c) OWA-3-Datensatz von Januar 2018 bis März 2019

Abbildung 4.1: Als Zeitreihen repräsentierte Aufrufe der produktiven Datensätze

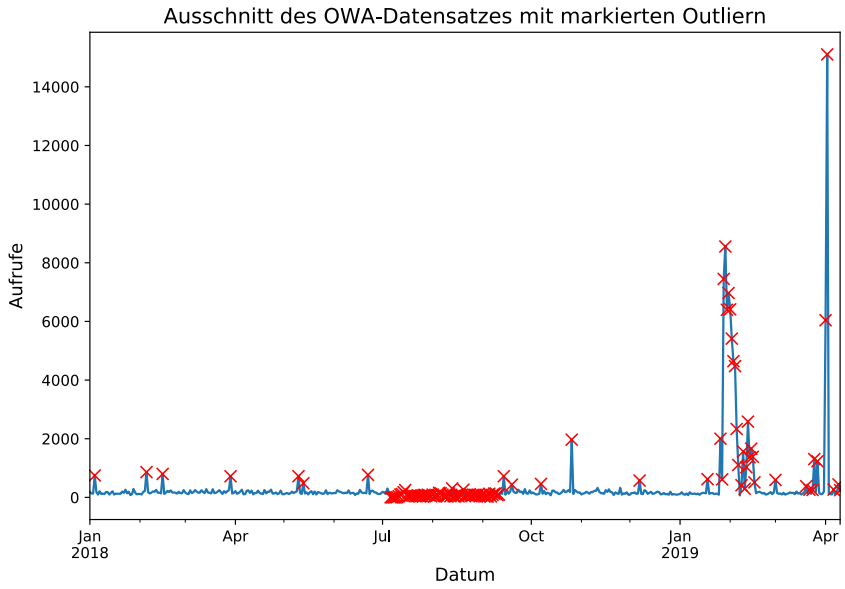


Abbildung 4.2: Aufrufe der Website seit Januar 2018 mit markierten Outliern

daher Alternativen betrachtet, mit denen die Evaluation der Algorithmen ergänzt werden kann.

4.2 OFFENE DATENSÄTZE

Es existieren verschiedene Quellen für öffentlich zugängliche Datensätze, die für den Einsatz im Bereich der Outlier Detection geeignet sind. Wang et al. nennen vier Quellen für offene Datensätze zur Outlier Detection [23]. Auch die von Goldstein et al. zur Evaluation von Unsupervised Outlier Detection Algorithmen verwendeten Datensätze wurden veröffentlicht [13].

Unter den Datensätzen der betrachteten Quellen wurden jedoch keine Web Analytics Datensätze gefunden, deren Struktur den produktiven Daten ähnelt. Somit sind die Datensätze zwar zur allgemeinen Bewertung von Outlier Detection Algorithmen über mehrere Anwendungsgebiete hinweg geeignet, nicht aber für die Evaluation des in dieser Arbeit entwickelten Konzepts.

Eine weitere Quelle für öffentlich zugängliche Datensätze ist das Open Data Network [2]. Über das Suchportal wurden verschiedene Web Analytics Datensätze gefunden, darunter der *LACity Website Traffic* Datensatz und der *Winnipeg Web Analytics* Datensatz. Allerdings sind die über das Portal gefundenen Datensätze bereits aggregiert und nicht gelabelt. Ohne detaillierte Informationen zu einzelnen Besuchern und Hintergrundinformationen zu den entsprechenden Websites ist eine manuelle Klassifizierung nicht möglich, weshalb auch diese Datensätze nicht zur Evaluation eingesetzt werden konnten.

4.3 GENERIERUNG VON SYNTHETISCHEN DATENSÄTZEN

Da keine passenden öffentlichen Datensätze gefunden wurden, bietet sich die Generierung von synthetischen Datensätzen an, um die Einschränkungen der produktiven Datensätze zu beheben. Ein Vorteil von generierten Daten besteht darin, dass sich der Outlier-Anteil eines Datensatzes, sowie die Stärke der Abweichung der Outlier von normalen Datenpunkten über Parameter steuern lässt.

Daher wurde im Rahmen der Arbeit ein Generator implementiert, der gelabelte Datensätze produzieren kann und über verschiedene Eingabeparameter konfigurierbar ist. Der Quellcode des Generators ist über GitHub [3] einsehbar.

4.3.1 *Ziele des Generators*

Die synthetischen Datensätze sollen eine Schätzung der Genauigkeit von Outlier Detection Algorithmen auf den produktiven Daten ermöglichen und daher in ihren statistischen Eigenschaften den Eingabedaten ähneln. Da die behandelten Outlier Detection Algorithmen auf Distanzen, Dichte oder Auf-

2 https://www.opendatanetwork.com (zuletzt besucht am 01.03.2020)
3 https://www.github.com/AndreBruecke/WebAnalyticsOutlierDetection (zuletzt besucht am 01.03.2020)

trittshäufigkeiten basieren, soll in erster Linie der Wertebereich und die Verteilung der Werte erhalten bleiben.

Diese Ziele sind identisch zu den Zielen bei der Nachbildung von natürlichen Prozessen. Die Ergebnisse der in Abschnitt 3.2 beschriebenen Literatur können daher zum Teil auf den Anwendungsfall dieser Arbeit übertragen werden. Allerdings ist die exakte Nachbildung bei Web Analytics Daten nicht so entscheidend, wie es bei Simulationen der Fall ist, die als Basis für Prognosen dienen sollen.

4.3.2 *Wahl des Modells*

Markov-Prozesse und autoregressive Modelle gehören zu den am weitesten akzeptierten Modellen für die Generierung von synthetischen Daten [4]. In einem empirischen Vergleich eines autoregressiven Modells mit Markov-Ketten erster und zweiter Ordnung ergaben sich keine wichtigen Unterschiede zwischen den Methoden [11]. Es wurde sich dazu entschieden, Markov-Ketten zur Nachbildung der Eingabedaten zu verwenden, da diese nicht auf spezielle Zeitreihen angepasst werden müssen und somit universeller einsetzbar sind. Da auch von Shamshad et al. keine signifikanten Verbesserungen des Markov-Modells zweiter Ordnung gegenüber dem Modell erster Ordnung beobachtet werden konnten [21], wurde ein Modell erster Ordnung gewählt. So wächst die Größe der Übergangsmatrix nur quadratisch zur Anzahl der Zustände.

4.3.3 *Funktionsweise im Detail*

Der `Preprocessor` erstellt für jede Spalte eines übergebenen Datensatzes eine Übergangsmatrix. Vereinfachend wird angenommen, dass der Wertebereich einer Zeitreihe alle Ganzzahlen zwischen dem Minimum und dem Maximum umfasst. Dieser Ansatz wurde gewählt, da im Wertebereich von Web Analytics Daten keine Lücken erwartet werden. Es ist also gewünscht, dass die generierten Zeitreihen auch Werte zwischen dem Minimum und dem Maximum annehmen können, die in den Eingabedaten nicht auftreten.

Dieser Wertebereich wird in eine bestimmte Anzahl von Zuständen unterteilt, sodass jedem Zustand die vom Parameter `approximateStateSize` gegebene Anzahl von Werten zugeordnet sind. Je kleiner die Zustandsgröße gewählt wird, desto genauer können die Eingabedaten nachgebildet werden, desto größer sind allerdings die Übergangsmatrizen.

Zur Berechnung der Übergangswahrscheinlichkeiten werden zunächst die Zustandsübergänge gezählt, indem paarweise über die Eingabe iteriert wird. Es entsteht eine Matrix M vom Typ $n \times n$, wobei n durch die Anzahl von Zuständen gegeben ist. Das Element M_{ij} gibt an, wie oft in der Eingabe vom Zustand i in den Zustand j übergegangen wurde. Diese totalen Werte werden schließlich in relative Wahrscheinlichkeiten umgewandelt, indem jedes

Element einer Zeile durch die Summe dieser Zeile geteilt wird.

Abbildung 4.3 zeigt eine beispielhafte Zeitreihe zur Demonstration der Funktionalität des Generators. Der Wertebereich dieser Zeitreihe umfasst die Werte 1 bis 7. Zur Veranschaulichung der Einschränkungen des Generators wurde eine Zeitreihe mit starker Saisonalität als Eingabe gewählt. Bei einer Zustandsgröße von `approximateStateSize = 2` wird der Wertebereich in drei Zustände unterteilt, wobei dem ersten Zustand die Werte 1 bis 3, dem zweiten Zustand die Werte 4 und 5, und dem dritten Zustand die Werte 6 und 7 zugewiesen werden.

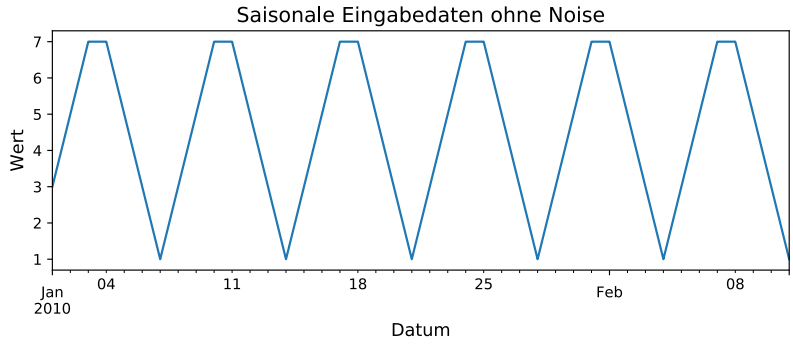


Abbildung 4.3: Einfache Zeitreihe mit Saisonalität

Die Übergangsmatrix M enthält die relativen Übergangswahrscheinlichkeiten für jeden Zustand.

$$M = \begin{bmatrix} 0.65 & 0.35 & 0.0 \\ 0.5 & 0.0 & 0.5 \\ 0.0 & 0.5 & 0.5 \end{bmatrix}$$

Der `DataGenerator` nutzt die vom `Preprocessor` erstellten Matrizen, um synthetische Zeitreihen zu generieren. Ausgehend von einem zufälligen Startzustand werden die möglichen Zustandsübergänge aus der entsprechenden Matrix ermittelt. Unter Berücksichtigung der relativen Wahrscheinlichkeiten als Gewichtung wird ein zufälliger Folgezustand gewählt. Dies wird wiederholt, bis die gegebene Anzahl von Werten generiert wurde. Dem Generator kann ein Seed übergeben werden, der zur Erzeugung aller pseudozufälligen Zahlen genutzt wird und das Ergebnis somit reproduzierbar macht.

Aus Abbildung 4.4 wird ersichtlich, dass ein regelmäßiges saisonales Pattern mit Markov-Ketten erster Ordnung nicht nachgebildet werden kann.

Zusätzlich zur Nachbildung von Eingabedaten bietet der Generator die Möglichkeit, einen einfachen linearen Trend zu ausgewählten Zeitreihen der Ausgabe hinzuzufügen. In diesem Fall bleiben die statistischen Eigenschaften der Eingabedaten allerdings nicht erhalten. Die Steigung des Trends ist eine positive Dezimalzahl und kann vom Benutzer bestimmt werden.

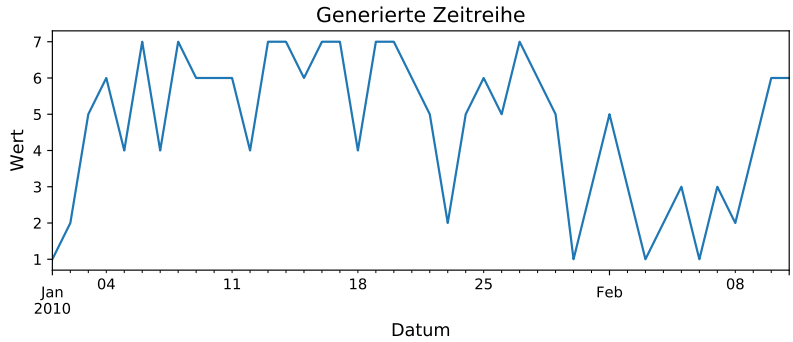


Abbildung 4.4: Generierte Zeitreihe, bei `approximateStateSize = 2` und `seed = 1`

In die generierten Daten können anschließend verschiedene Arten von Outliern eingefügt werden, die an die Outlier in den produktiven Daten angelehnt sind. So können Outlier entweder einzelne Datenpunkte ersetzen, einen ganzen Bereich um einen festen Wert senken oder heben, sowie Werte in einem Bereich exponentiell ansteigen lassen. Der Nutzer kann bestimmen, in welche Features die Outlier eingefügt werden, und wie stark die Outlier vom Rest der Daten abweichen sollen. Die Abweichung wird relativ zu einem lokalen Mittelwert des Features berechnet, um zu verhindern, dass Outlier auf verschiedenen Features unterschiedlich stark abweichen.

4.3.4 *Bewertung und Verbesserungsmöglichkeiten*

Um zu prüfen, wie gut die produktiven Eingabedaten vom Generator nachgebildet werden, wurden analog zu Hocaoglu [17] die grundlegenden statistischen Eigenschaften der Eingabedaten mit denen der generierten Daten vor dem Einfügen der Outlier verglichen.

Die in [17] zur Evaluation genutzten Eigenschaften Minimum, Maximum, Mittelwert, Median und Standardabweichung wurden für die einzelnen Features der Eingabedaten ermittelt. Für die Eigenschaften der generierten Features wurde der Mittelwert aus zehn Ergebnissen bei einer Zustandsgröße von 1 und Seeds von 1 bis 10 gebildet. Die Länge der generierten Daten betrug 356 Datenpunkte.

Tabelle 4.1 zeigt die Vergleichswerte für das Feature Aufrufe und Tabelle 4.2 die Vergleichswerte für das Feature Besucher. Die Aufrufe pro Besucher wurden nicht betrachtet, da sie bei der Erstellung von synthetischen Datensätzen nicht generiert, sondern aus den beiden anderen Features berechnet werden sollen. Zur Berechnung der Abweichung wurde der Mittelwert aus den absoluten Differenzen gebildet, damit sich positive und negative Differenzen nicht gegenseitig aufheben.

Die statistischen Eigenschaften beider Features konnten mit ausreichender Genauigkeit nachgebildet werden. Es sind keine Abweichungen zu erkennen, die so stark sind, dass sie Auswirkungen auf die Erkennung der Outlier haben könnten. Durchschnittlich ist der Wertebereich der generierten Daten kleiner als der Wertebereich der Eingabedaten. Es besteht also

die Möglichkeit, dass je nach gewähltem Seed die Erkennung der Outlier auf den synthetischen Datensätzen im Vergleich zu den Eingabedaten etwas vereinfacht wird.

Eigenschaft	Original	Generiert	Absolute Abweichung
Maximum	261	258,3	2,7
Minimum	65	71,4	6,4
Mittelwert	134,67	134,96	1,64
Median	128	128,6	1,4
Standardabweichung	33,39	34,04	1,05

Tabelle 4.1: Vergleich der statistischen Eigenschaften für das Feature Aufrufe. Die Eigenschaften der generierten Daten stellen einen Mittelwert über zehn verschiedene Seeds dar.

Eigenschaft	Original	Generiert	Absolute Abweichung
Maximum	58	57,6	0,4
Minimum	10	10,7	0,7
Mittelwert	28	28,21	0,43
Median	27	27	0,6
Standardabweichung	8,53	8,8	0,39

Tabelle 4.2: Vergleich der statistischen Eigenschaften für das Feature Besucher. Die Eigenschaften der generierten Daten stellen einen Mittelwert über zehn verschiedene Seeds dar.

Eine wesentliche Einschränkung des Generators ist, dass keine Korrelationen zwischen Features multivariater Eingabedaten berücksichtigt werden. Im Falle der Web Analytics Daten kann somit zum Beispiel die Abhängigkeit zwischen Besuchern und Aufrufen verloren gehen. Wenn eine feste Abhängigkeit vorliegt, wie bei den Aufrufen pro Besuchern, kann das entsprechende Feature nach Einfügen der Outlier berechnet werden.

Bisher werden als Eingabedaten nur Zeitreihen mit ganzzahligen Werten unterstützt. Trend und Saison können nicht nachgebildet werden, sollten also vorher aus den Zeitreihen entfernt werden.

4.3.5 *Synthetische Datensätze zur Unterstützung der Evaluation*

Die in dieser Arbeit betrachteten Outlier Detection Algorithmen erkennen Outlier entweder aufgrund ihrer Auftrittshäufigkeit, oder ihrer Distanz zu anderen Datenpunkten. Es ist daher zu erwarten, dass die Algorithmen auf Datensätzen mit geringem Outlier-Anteil, in denen die Outlier signifikant von anderen Datenpunkten abweichen, die höchste Genauigkeit erzielen.

Die synthetischen Datensätze sollen verschiedene Grenzfälle abdecken, in denen Outlier Detection erschwert wird. Der SYN-1-Datensatz umfasst lediglich 90 Datenpunkte, bei einem Outlier-Anteil von 20%. Durch die geringe Länge haben die Cluster des Datensatzes eine geringere Dichte, als bei den produktiven Daten.

Der SYN-2-Datensatz enthält einen einfachen linearen Trend und hat eine Länge von 2 Jahren. Neben einzelnen Outliern wurde ein 20-tägiger Ausfall simuliert. Insgesamt beträgt der Outlier-Anteil 8,36%.

5 KONZEPT ZUR ERKENNUNG VON OUTLIERN

5.1 AUSGANGSSITUATION

5.1.1 *Ziele und fachlicher Kontext*

Das zu konzipierende System soll zur Datenanalyse in einem Data Warehouse eingesetzt werden und in den dort zwischengespeicherten Web Analytics Daten Outlier identifizieren. Die Genauigkeit, mit der das System Outlier erkennt, soll maximiert werden. Weiterhin soll das System vollautomatisch funktionieren und möglichst wenig Konfigurationsaufwand nach der Inbetriebnahme erfordern. Deswegen wird bei der Konzeption zusätzlich auf eine möglichst hohe Robustheit Wert gelegt.

Die Optimierung der Performance wird nicht betrachtet, da die Verarbeitung der Daten nicht in einem performancekritischen Kontext stattfindet, und der Datensatz täglich nur um einen Eintrag wächst.

5.1.2 *Das PyOD Framework*

Zur Konzeption und Evaluation des Systems wurde Python als Programmiersprache verwendet. Grund hierfür ist die große Auswahl an Frameworks zur Verarbeitung, Analyse und Visualisierung von Daten.

Das eingesetzte Outlier Detection Framework, PyOD (Python Outlier Detection), ist das Python Framework mit der größten Auswahl an Algorithmen und bietet zudem für alle Algorithmen eine einheitliche API. Es wird bereits in vielen Forschungsprojekten und kommerziellen Projekten eingesetzt [23]. Jeder Algorithmus liefert sowohl ein Outlier Ranking, als auch eine Einteilung in die Klassen Outlier und Normalverhalten für die geprüften Datenpunkte zurück. Der Schwellenwert für die Umwandlung von Rankings in binäre Label kann durch den Nutzer über den Eingabeparameter `contamination` bestimmt werden. Jeder Algorithmus besitzt eine Methode zum Trainieren des Algorithmus und eine Methode zur Ausführung des Algorithmus.

5.1.3 *Vorauswahl der Algorithmen*

Da nicht alle im Framework implementierten Algorithmen in der Arbeit behandelt werden konnten, wurde eine Vorauswahl getroffen. Algorithmen, die auf Supervised oder Semi-Supervised Learning basieren, wurden aufgrund ihres Konfigurationsaufwands ausgeschlossen. Es ist im Web Analytics Bereich nicht unwahrscheinlich, dass sich die Eigenschaften der Daten stark ändern, beispielsweise durch natürliches Wachstum der Website. In

diesem Fall kann es notwendig werden, neue gelabelte Daten herzustellen, was der Anforderung des niedrigen Konfigurationsaufwands widerspricht. Auf Unsupervised Learning basierende Methoden erfordern in einem solchen Fall hingegen lediglich eine Optimierung der Eingabeparameter.

Zur besseren Vergleichbarkeit der Evaluationsergebnisse wurde sich auf die Verfahren beschränkt, die auch von Goldstein et al. [13] evaluiert wurden. Aus der Schnittmenge wurden schließlich drei Algorithmen für die Evaluation ausgewählt. LOF und kNN wurden als Vertreter der klassischen Verfahren gewählt, da sie in der Literatur häufig zum Vergleich mit neu entwickelten und spezialisierteren Algorithmen verwendet werden. HBOS wurde als Vertreter der statistischen Verfahren aufgenommen. Außerdem stellt sich bei Betrachtung dieses Verfahrens die interessante Frage, ob der Kompromiss zwischen Performance und Genauigkeit sich für das System lohnt.

5.2 VORSTELLUNG DER DESIGNENTSCHEIDUNGEN

Neben der Vorverarbeitung der Eingabedaten umfasst der Entwurf eines Outlier Detection Systems die Wahl der Feature-Kombinationen, die Konfiguration der Eingabeparameter der verglichenen Algorithmen und schließlich die Wahl des am besten geeigneten Algorithmus.

Aus den drei Features Aufrufe, Besucher und durchschnittliche Aufrufe pro Besucher soll eine Kombination gefunden werden, auf der die Algorithmen die zuverlässigsten Ergebnisse erzielen. Zur Auswahl stehen vier Kombinationen mit unterschiedlicher Anzahl von Features [1]:

(a) Aufrufe

(b) Aufrufe, Besucher

(c) Aufrufe, Aufrufe pro Besucher

(d) Aufrufe, Besucher, Aufrufe pro Besucher

Bevor die Algorithmen untereinander verglichen werden, wird für jeden Algorithmus eine optimale Konfiguration der Eingabeparameter ermittelt. Optimal ist in diesem Fall diejenige Kombination, die die höchste Genauigkeit auf dem OWA-Datensatz erzielt. Aus den betrachteten Algorithmen mit ihren optimierten Eingabeparametern wird schließlich der am besten geeignete Algorithmus ausgewählt. Die Bewertung erfolgt zusätzlich zur Genauigkeit anhand der Robustheit der Algorithmen.

Somit ergeben sich folgende Designentscheidungen, die im Rahmen der empirischen Experimente zu treffen sind:

1 Kombinationen ohne Aufrufe werden nicht betrachtet, da sich alle im OWA-Datensatz gefundenen Outlier deutlich auf dieses Feature auswirken

1. Wahl der Feature-Kombinationen
2. Konfiguration der relevanten Eingabeparameter
3. Wahl des zu verwendenden Algorithmus

5.3 KRITERIEN ZUR BEWERTUNG DER DESIGNENTSCHEIDUNGEN

Zur Konzeption des Gesamtsystems müssen die aufgestellten Designentscheidungen begründet getroffen werden. Der Fokus dieser Arbeit liegt auf der Maximierung der Genauigkeit, da Outlier zuverlässig identifiziert werden und keine manuelle Überprüfung erfordern sollen. Das zweite Kriterium ist die Robustheit. Sie gibt an, wie das System auf Veränderungen der Rahmenbedingungen reagiert.

5.3.1 *Metriken zur Bewertung der Genauigkeit*

Zur Messung der Genauigkeit von Outlier Detection Algorithmen wurde eine der in Abschnitt 2.5 beschriebenen Metriken gewählt.

Wenn die tatsächliche Anzahl von Outliern eines Datensatzes im Voraus bekannt ist, ist das einfachste Performance-Maß die Precision at n [7]. Der Parameter n kann in diesem Fall auf die Anzahl der Outlier gesetzt werden, so dass ein perfekter Score genau dann erreicht wird, wenn alle Outlier unter den ersten n Rängen platziert sind. Ein entscheidender Nachteil dieser Metrik ist, dass lediglich n Ränge in den Score einfließen. Die Reihenfolge der Outlier in den Rängen unter oder über n hat keinerlei Einfluss auf das Ergebnis. Campos et al. führen zur Veranschaulichung des Problems einen Datensatz mit 10 Outliern und einer Millionen normalen Datenpunkten an. Werden den Outliern in diesem Fall die Ränge 11 bis 20 zugewiesen, ergibt sich eine P@10 von $0,0$ [7]. Den gleichen Wert erhält man, wenn die Outlier den letzten 10 Rängen zugewiesen werden.

Daher ist die Precision at n nicht ausreichend, um darauf aufbauend die Designentscheidungen des zu konzipierenden Systems zu treffen.

ROC-Kurven sind eine im Bereich der Outlier Detection als de facto Standard angesehene Metrik zur Messung der Genauigkeit von Algorithmen [23], obwohl auch diese Metrik bekannte Einschränkungen hat. Saito et al. zeigen, dass ROC Kurven irreführend sein können, wenn sie auf Probleme mit unausgewogenen Klassen angewandt werden. Outlier Detection gehört zu dieser Art von Problemen, da die Klasse der Outlier per Definition deutlich kleiner als die Klasse des Normalverhaltens ist. Es wird gezeigt, dass Precision-Recall-Kurven in diesen Fällen eine bessere Alternative zur Messung der Genauigkeit sind [20].

Daher wurde sich dazu entschieden, im Rahmen der Konzeption Precision-Recall-Kurven statt ROC-Kurven zu verwenden. Um die Kurven zu einem einzelnen Wert zusammenzufassen, wurde die Average Precision (AP) verwendet.

5.3.2 *Bewertung der Robustheit*

Die Robustheit von Algorithmen ist im Gegensatz zur Genauigkeit nicht eindeutig messbar. In dieser Arbeit wird die Robustheit aus der Sensitivität der Algorithmen bei Änderungen der Parameter oder der Eingabedaten abgeleitet.

Zur Visualisierung werden Boxplots verwendet, die nicht nur Anhaltspunkte für die Robustheit liefern, sondern auch die maximale Genauigkeit abbilden.

5.4 DURCHFÜHRUNG DER EXPERIMENTE

Die vorgestellten Designentscheidungen sollen im Rahmen von Experimenten begründet getroffen werden. Primärer Datensatz ist der OWA-Datensatz, der das Verhalten des produktiven Systems abbildet. Unterstützend wird ein Teil der Experimente auch auf den generierten SYN-1 und SYN-2-Datensätzen ausgeführt.

5.4.1 *Reihenfolge der Experimente*

Eine Herausforderung bei der Evaluation der Designentscheidungen ist die Abhängigkeit der Eingabeparameter von der Beschaffenheit der Eingabedaten. Die optimalen Werte für die Eingabeparameter hängen von der Wahl der Features ab, während eine begründete Wahl der Feature-Kombinationen nicht unter der standardmäßigen Konfiguration der Parameter vorgenommen werden kann.

Um die Zahl der Versuche zu beschränken, mussten die betrachteten Eingabeparameter nach der Vorauswahl weiter eingegrenzt werden. Dazu wurde zunächst eine Sensitivitätsanalyse durchgeführt, um diejenigen Parameter zu bestimmen, die den größten Einfluss auf die Genauigkeit der Algorithmen zeigen. Anschließend wurden die Feature-Kombinationen über den gesamten Wertebereichen der Eingabeparameter ausgewertet und ausgewählt. Aufbauend darauf konnte die optimale Konfiguration der betrachteten Parameter bestimmt und schließlich ein geeigneter Algorithmus gewählt werden.

5.4.2 *Vorauswahl der Eingabeparameter*

Aufgrund der hohen Anzahl an möglichen Kombinationen ist es nicht praktikabel, alle Eingabeparameter auf allen Features zu evaluieren. Außerdem bietet das PyOD Framework neben den Parametern der ursprünglichen Outlier Detection Algorithmen zusätzliche Eingabeparameter, die sich lediglich auf die Zeit- und Speicherkomplexität der Algorithmen auswirken.

Tabelle 5.1 zeigt die betrachteten Eingabeparameter der Algorithmen mit den Standardwerten des Frameworks. Für die nicht aufgeführten Parameter wurden bei den Experimenten die Standardwerte übernommen.

Parameter	Standardwert	Beschreibung
		k-NN
n_neighbors	5	Anzahl der nächsten Nachbarn, entspricht dem Parameter k
method	*largest*	Methode zur Berechnung des Scores
metric	*minkowski*	Metrik zur Berechnung von Distanzen
		LOF
n_neighbors	20	Anzahl der nächsten Nachbarn, entspricht dem Parameter *MinPts*
metric	*minkowski*	Metrik zur Berechnung von Distanzen
		HBOS
n_bins	10	Anzahl der Bins pro Histogramm

Tabelle 5.1: Betrachtete Eingabeparameter und ihre Standardwerte in PyOD

5.4.3 *Bestimmung des Wertebereichs für n_neighbors und n_bins*

Für die kategorischen Parameter ist der Wertebereich durch die Kategorien klar definiert. Bei den Parametern `n_bins` und `n_neighbors` hingegen hängt der Wertebereich von den Eingabedaten ab. Die obere Grenze des Wertebereichs ist gegeben durch die Länge des Datensatzes, da dies die maximale Anzahl von Bins pro Histogramm bzw. die maximale Größe der Nachbarschaft darstellt. Die untere Grenze ist die minimale Anzahl von Bins oder Nachbarn, also 1. Allerdings sind Werte nahe dieser absoluten Grenzen nicht realistisch. Beispielsweise liefert HBOS bei `n_bins = 1` für jeden Datenpunkt identische Scores, und auch die Bestimmung der Dichte einer Nachbarschaft bei LOF ist für `n_neighbors = 1` nur sehr ungenau möglich.

Breunig et al. untersuchen in ihrer Arbeit zu LOF die untere und obere Grenze des Parameters *MinPts*. Demnach sollte *MinPts* unabhängig vom verwendeten Datensatz nicht kleiner als 10 gewählt werden, um statistische Schwankungen zu vermeiden. Eine Richtlinie für die Wahl der oberen Grenze ist die maximale Anzahl an Datenpunkten, die potenziell lokale Outlier darstellen können [5].

Leider fehlen derartige Richtlinien zur Bestimmung des Wertebereichs in der Primärliteratur zu HBOS und k-NN. Goldstein et al. erwähnen eine Faustregel, nach der die Anzahl der Bins auf die Wurzel der Instanzen eines Datensatzes gesetzt wird [12]. Untere und obere Grenzen für den Parameter werden allerdings nicht genannt. In [13] wird für den Parameter k des k-NN Algorithmus die Faustregel $10 < k < 50$ angegeben.

Campos et al. wählen den Wertebereich von k bei der Evaluation verschiedener auf k-NN und LOF basierender Algorithmen zwischen 1 und 100 [7], wobei nur die Genauigkeit der Algorithmen evaluiert wird, und nicht ihre

Robustheit.

Bei einer Bewertung der Robustheit verschiedener Algorithmen ist es wichtig, unrealistische Parameter-Werte zu vermeiden, da diese die Robustheit verfälschen. Gleichzeitig sollten die analysierten Wertebereiche verschiedener Algorithmen möglichst eine identische Breite haben, um eine Vergleichbarkeit herzustellen.

Da eine ausführliche Betrachtung des Wertebereichs nur für LOF vorliegt, wurde sich an diesen Grenzen orientiert. Der Anteil lokaler Outlier zur Bestimmung der oberen Grenze ist in jedem Fall kleiner oder gleich des absoluten Outlier-Anteils eines Datensatzes. Dieser Anteil ist im Zuge der Konzeption einfach zu bestimmen, da gelabelte Datensätze vorliegen. Um den Wertebereich nicht zu stark einzuschränken, wurde der ermittelte Outlier-Anteil jedes Datensatzes auf das nächsthöhere Zehntel aufgerundet. Die absolute Zahl an Datenpunkten, die diesen Anteil widerspiegelt, wird als obere Grenze gewählt. Die untere Grenze wird auf den festen Wert 10 gesetzt. Für k-NN wurde der Wertebereich vom Startwert 10 auf 5 verschoben, damit der vom Framework gegebene Standardwert des `n_neighbor` Parameters enthalten ist.

5.4.4 *Sensitivitätsanalyse*

Sensitivitätsanalysen werden unter anderem durchgeführt, um insignifikante Parameter zu finden, und zu bestimmen, welche Eingaben am meisten zur Variabilität der Ausgabe beitragen [14]. Somit ist die Sensitivität von Eingabeparametern im Bereich der Outlier Detection ein sinnvolles Indiz für die Robustheit der Algorithmen und hilft dabei, die wichtigsten Parameter für die Konfiguration zu bestimmen.

Bei der Sensitivitätsanalyse gibt es die Möglichkeiten, jeden Parameter unabhängig voneinander, oder aber Kombinationen von Parametern zu analysieren [14]. Es wurde sich dafür entschieden, die Eingabeparameter nicht in Kombination zu betrachten. Grund hierfür ist, dass sonst eine exponentielle Anzahl von Ausführungen erforderlich wäre, und die Anzahl der betrachteten Werte für die numerischen Parameter Eingabewerte stark verringert werden müsste.

Hamby nennt verschiedene Sensivitäts-Maße einzelner Parameter, die für bestimmte Punkte im Wertebereich der Parameter die Änderungen der Ausgabe messen. So werden Parameter beispielsweise von ihrem Mittelwert ausgehend um ± 4 Standardabweichungen variiert, oder es wird der gesamte Wertebereich der Parameter betrachtet [14].
In einer empirischen Evaluation stellte sich der Sensitivity Index (SI) unter den einfachen Methoden als das verlässlichste Maß bei einer moderaten Anzahl von Parametern heraus [15].

Da die Sensitivität der Eingabeparameter über ihren gesamten Wertebereich hinweg bestimmt werden soll, bietet sich der Sensitivity Index auch für die Evaluation der Algorithmen an.

Zur Berechnung des Sensitivity Index wird die Differenz zwischen dem maximalen und dem minimalen Wert der Ausgabe bestimmt und durch den maximalen Wert dividiert [14]. Da zur Messung der Genauigkeit von Algorithmen die Average Precision (AP) verwendet wird, ist der Sensitivity Index in diesem speziellen Fall gegeben durch:

$$SI = \frac{max(AP) - min(AP)}{max(AP)}$$

Eine maximale Sensitivität liegt bei $SI = 1$ vor, während ein Parameter mit $SI = 0$ keinen messbaren Einfluss auf die Genauigkeit der Algorithmen aufweist.

Bei der Durchführung der Sensitivitätsanalyse wurden die betrachteten Eingabeparameter jeweils über ihren gesamten Wertebereich variiert, während die übrigen Parameter auf ihren Standardwerten belassen wurden. Die Sensitivitätsanalyse wurde auf dem OWA-Datensatz für jede der vier Feature-Kombinationen durchgeführt. Die Ergebnisse sind in Tabelle 5.2 dargestellt. Die betrachteten Distanz-Metriken zeigten keinen signifikanten Einfluss auf die Genauigkeit der Algorithmen, unabhängig davon, welche Kombination von Features als Eingabe gewählt wurde. Allerdings nimmt der Sensitivity Index mit steigender Anzahl von Features leicht zu. Währenddessen hatte der Eingabeparameter `method` des k-NN Algorithmus durchschnittlich einen geringeren Einfluss auf die Genauigkeit des Algorithmus als der Parameter `n_neighbors`. LOF erzielte bei Variation von `n_neighbors` über den gesamten Wertebereich sehr unterschiedliche Genauigkeiten, was sich in einem hohen durchschnittlichen Sensitivity Index widerspiegelte. Nur der Parameter `n_bins` zeigte mit einem Wert von $0,4956$ im Durchschnitt eine noch höhere Sensitivität.

Die Sensitivitätsanalyse zeigte, dass die ursprünglichen Eingabeparameter der Algorithmen deren Genauigkeit am meisten beeinflussen. Für den k-NN Algorithmus sollte zudem der Parameter `method` beachtet werden.

Die gewählten Distanz-Metriken zeigten nur geringen Einfluss auf die Genauigkeit der Algorithmen. Diese Parameter wurden daher für die nachfolgenden Experimente auf den Wert gesetzt, der die höchste Genauigkeit auf dem OWA-Datensatz erzielte, und müssen bei möglicher Nachjustierung des Systems nicht mehr angepasst werden.

	A	A, B	A, D	A, B, D	Durchschnitt
k-NN					
n_neighbors	0,1048	0,2135	0,2235	0,108	0,1625
method	0,1306	0,0705	0,0518	0,0725	0,0814
metric	0,0	0,0006	0,0049	0,0068	0,0031
LOF					
n_neighbors	0,5945	0,4261	0,4341	0,4517	0,4766
metric	0,0	0,0073	0,0161	0,0236	0,0118
HBOS					
n_bins	0,6402	0,5347	0,4549	0,3525	0,4956

Tabelle 5.2: Sensitivity Index der Parameter bei Anwendung auf den OWA-Datensatz, für verschiedenen Kombinationen der Features Aufrufe (A), Besucher (B) und durchschnittliche Aufrufe pro Besucher (D).

5.5 ERGEBNISSE

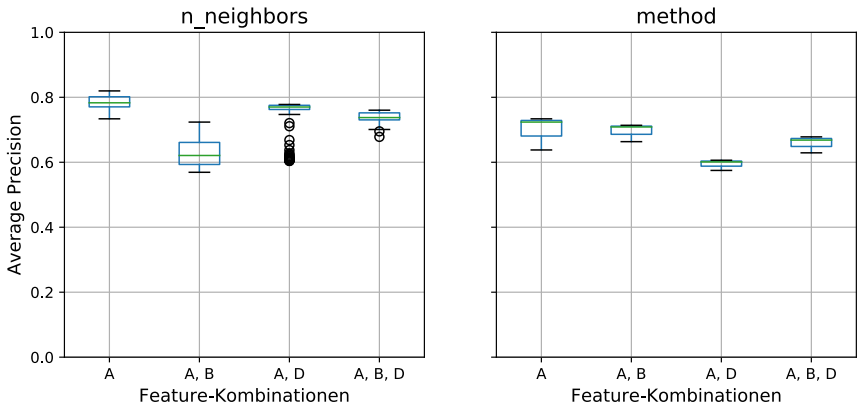
5.5.1 *Wahl der Feature-Kombinationen*

Zur Wahl der Feature-Kombinationen können die Ergebnisse der im Rahmen der Sensitivitätsanalyse durchgeführten Experimente herangezogen werden. Ausschlaggebend sind die Ergebnisse auf dem OWA-Datensatz, der das produktive System am besten beschreibt. Die synthetischen Datensätze sind zur Wahl der Features unzureichend, da Aufrufe und Besucher unabhängig voneinander generiert wurden, und die durchschnittlichen Aufrufe pro Besucher somit nicht notwendigerweise das Verhalten des realen Systems widerspiegeln.

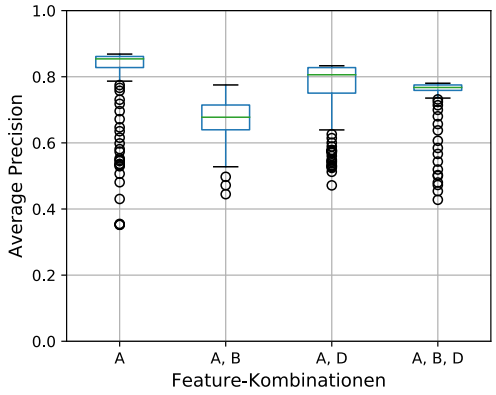
Von Bedeutung für die Auswahl der Features ist zum einen die maximale und minimale erreichte Genauigkeit pro Kombination, und zum anderen die Robustheit auf den definierten Wertebereichen der Eingabeparameter. Abbildung 5.1c visualisiert eine solche Verteilung für HBOS unter Variation von `n_bins`. Aus dem Boxplot ist ersichtlich, dass der höchste Maximalwert und der höchste Median auf der Kombination aus Aufrufen und durchschnittlichen Aufrufen pro Besucher erreicht wurden. Gleiches gilt für die 25% und 75% Perzentile (untere und obere Kante der Box). Der Minimalwert war dagegen für die Kombination aller Features etwas höher. Da das Hauptziel die Maximierung der Genauigkeit ist, wurde die erste der beiden Kombinationen als Eingabe gewählt.

Abbildung 5.1b zeigt, dass LOF den höchsten Maximalwert und Median erzielte, wenn als Feature nur die Aufrufe betrachtet wurden.

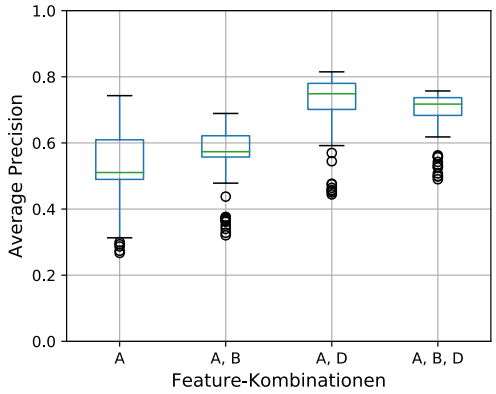
Für k-NN wurden die Parameter `n_neighbors` und `method` getrennt ausgewertet. Die Ergebnisse sind in Abbildung 5.1a visualisiert. Für beide Parame-



(a) k-NN



(b) LOF



(c) HBOS

Abbildung 5.1: Verteilung der von den Algorithmen erreichten Average Precision für verschiedene Feature-Kombinationen, unter Variation der einflussreichsten Parameter.

ter wurde der höchste Median und Maximalwert ebenfalls auf den Aufrufen erreicht.

5.5.2 *Konfiguration der Eingabeparameter*

Da die Menge der zu betrachteten Eingabeparameter während der Sensitivitätsanalyse auf die essenziellen Parameter reduziert wurde, konnten zur Konfiguration alle Kombinationen von Parametern betrachtet werden.

Da für LOF und HBOS nur jeweils ein Parameter betrachtet wird, können die Ergebnisse der durchgeführten Experimente wiederverwendet werden. Für k-NN wurden zwei signifikante Parameter ermittelt, deren Kombinationsmöglichkeiten noch untersucht werden müssen.

Dazu wurde die Average Precision bei Variation von `n_neighbors` in Abhängigkeit von der gewählten Methode ausgewertet. Die im letzten Unterabschnitt gewählten Features Aufrufe und durchschnittliche Aufrufe pro Besucher dienten als Eingabe. Abbildung 5.2 stellt die Ergebnisse der Auswertung dar. Während alle Methoden ähnliche Maximalwerte erzielten, war die Verteilung der Average Precision Werte bei den Methoden `mean` und `median` etwas besser als bei der Methode `largest`. Die Methode `median` wurde ausgewählt, da sie einen höheren Maximalwert erreicht. Allerdings wurden die Unterschiede zur Methode `mean` nicht als signifikant bewertet, man kann daher beide Methoden als geeignet betrachten.

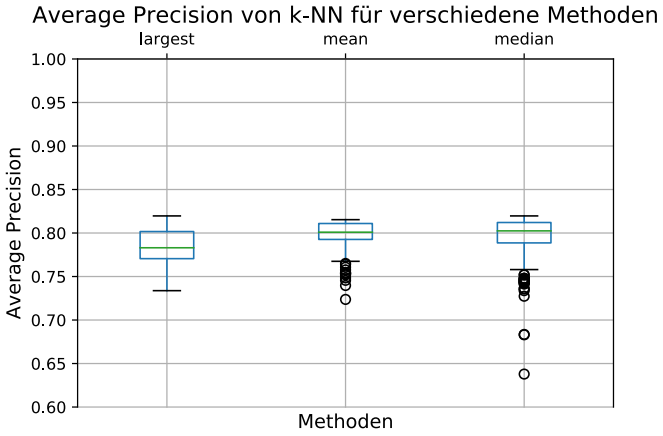


Abbildung 5.2: Verteilung der Average Precision von k-NN für verschiedene Werte des Eingabeparameters `method`, unter Variation von `n_neighbors`.

Die maximale Average Precision wurde bei `n_neigbors = 75` erreicht und betrug 0,8196. Somit wurde die optimale Konfiguration für k-NN ermittelt.

Für LOF wurde die höchste Average Precision ebenfalls bei `n_neigbors = 75` erzielt und betrug 0,8685, während für HBOS bei `n_bins = 142` die maximale Average Precision von 0,815 erreicht wurde.

5.5.3 *Wahl des Algorithmus*

Ziel der Experimente war es, die Genauigkeit der Algorithmen zu maximieren und dabei eine möglichst gute Robustheit beizubehalten. Diese beiden Kriterien sind daher auch bei der Wahl des Algorithmus ausschlaggebend.

Ein Indiz für die Robustheit der Algorithmen ist durch den Sensitivity Index der relevanten Eingabeparameter auf den gewählten Kombinationen der Feature gegeben (siehe Tabelle 5.2). Allerdings ist der Sensitivity Index eher dazu geeignet, Parameter mit geringem Einfluss auf die Ausgabe zu finden, als ein zuverlässiges Maß der Robustheit zu liefern. Dies liegt daran, dass die minimalen und maximalen Werte der Ausgabe betrachtet werden, ohne dass eine Aussage über die Verteilung der häufigsten Werte getroffen werden kann. Außerdem wurde der Sensitivity Index über einem breiten Wertebereich berechnet und spiegelt daher lediglich die allgemeine Sensitivität der Parameter wider.

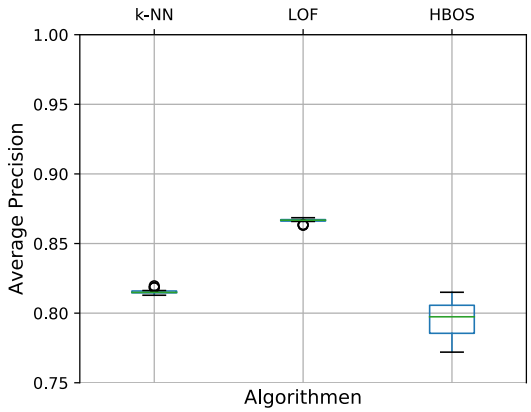
Um eine präzisere Aussage treffen zu können, wurde ein lokaler Wertebereich pro Algorithmus definiert. Dieser Wertebereich umfasst 10 Werte vor und 10 Werte nach demjenigen Parameterwert, der auf dem produktiven Datensatz die besten Ergebnisse erzielte. Diese Vorgehensweise wurde auch von Campos angewendet, um die Stabilität der Performance von Algorithmen zu bewerten [7]. Neben dem OWA-Datensatz wurden zur Bewertung der Robustheit die zwei synthetischen Datensätze herangezogen.

Die Ergebnisse des OWA-Datensatzes sind in Abbildung 5.3a zu sehen. Während der Sensitivity Index die Sensitivität zu einer einzelnen Zahl zusammenfasst, was bei einer hohen Anzahl von betrachteten Kombinationen vorteilhaft ist, bieten Boxplots zusätzliche Anhaltspunkte und liefern zudem eine grafische Repräsentation der Robustheit. Die Differenz zwischen Maximum und Minimum (auch oberer und unterer Whisker genannt) ähnelt dem Sensitivity Index, bezieht Ausreißer jedoch nicht mit ein. Die Höhe der Boxen gibt die sogenannte Interquantile Range (IQR) an, einen Bereich, in den 50% der mittleren Werte einer Verteilung fallen. Je kleiner die Interquantile Range und die Differenz zwischen Maximum und Minimum sind, desto stabiler sind die Ergebnisse über den lokalen Wertebereich hinweg. Da dieser lokale Wertebereich für die eventuelle Nachjustierung der Eingabeparameter mehr Bedeutung hat, als der gesamte Wertebereich, sind die Boxplots das wichtigere Indiz für die Robustheit der Algorithmen.

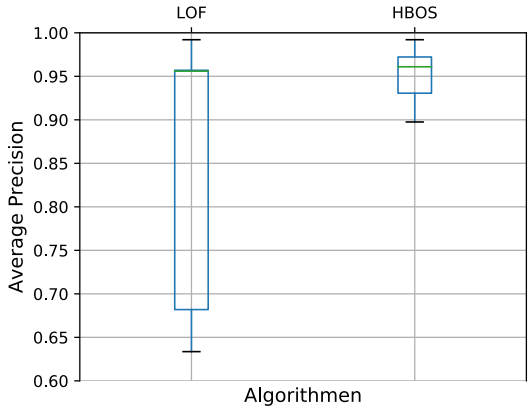
Die Ergebnisse von k-NN und LOF waren beinahe konstant, während HBOS eine höhere Distanz zwischen der minimalen und maximalen erreichten Average Precision, sowie eine breitere Interquantile Range zeigte.

Die Ergebnisse des SYN-2-Datensatzes in Abbildung 5.3c unterstützten diese Beobachtungen. Auch hier war die erreichte Genauigkeit von k-NN und LOF stabiler, als bei HBOS.

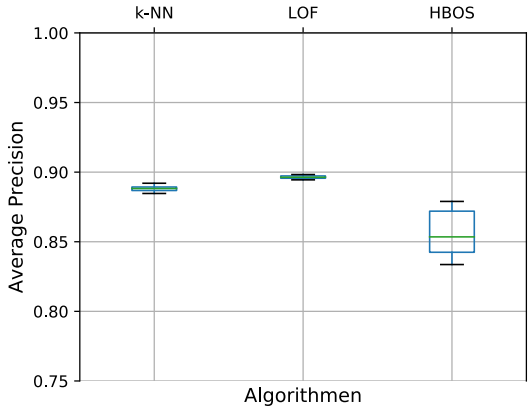
Für den SYN-1-Datensatz liegen nur die Ergebnisse von LOF und HBOS vor, da die Ausführung von k-NN auf dem Wertebereich zu einem Fehler führte. Abbildung 5.3b zeigt die verfügbaren Ergebnisse. LOF erreichte auf diesem Datensatz keine hohe Stabilität. Zwar lagen 50% der höheren Er-



(a) OWA-Datensatz



(b) SYN-1-Datensatz



(c) SYN-2-Datensatz

Abbildung 5.3: Genauigkeit der Algorithmen bei Betrachtung eines lokalen Wertebereichs von 10 Werten vor und nach dem optimalen Wert des primären Eingabeparameters. Für k-NN liegen auf dem SYN-1-Datensatz keine Daten vor, da die Ausführung bei den gegebenen Werten zu einer Exception führte.

gebnisse zwischen den AP-Werten 0,95 und 0,99, allerdings umfassten die niedrigeren 50% der Ergebnisse AP-Werte von unter 0,65 bis 0,95. Im Vergleich dazu fielen alle Ergebnisse von HBOS zwischen 0,9 und 0,99. Abbildung 5.4 zeigt, dass die niedrigen Genauigkeiten für die höheren Werte von `n_neighbors` ab dem Wert 79 auftraten. Dies lässt sich damit erklären, dass die für den OWA-Datensatz optimale Konfiguration des Parameters bei dem deutlich kleineren SYN-1-Datensatz in die Nähe der oberen Grenze des Wertebereichs fällt.

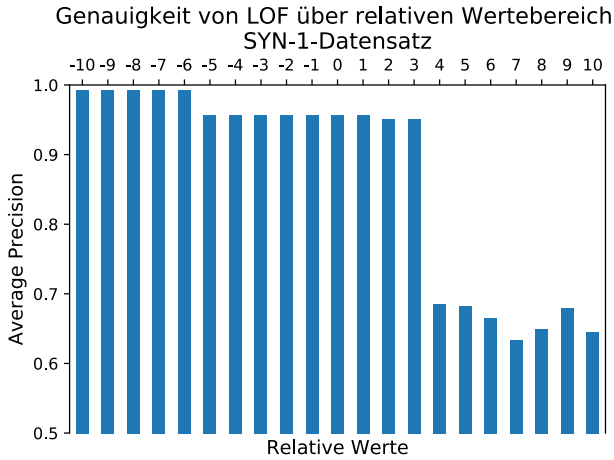


Abbildung 5.4: Genauigkeit von LOF über relativem Wertebereich des Parameters `n_neighbors`, um den Wert 75.

Hinsichtlich der Genauigkeit erzielte LOF auf den drei betrachteten Datensätzen die besten Ergebnisse, wobei auf dem SYN-1-Datensatz keine signifikanten Unterschiede zwischen LOF und HBOS bestanden. Weiterhin zeigte LOF eine sehr gute Robustheit, solange ein lokaler Wertebereich betrachtet wurde, der auf den jeweiligen Datensatz abgestimmt war. Daher wurde LOF als Algorithmus für das Outlier Detection System gewählt.

6 EVALUATION DES GESAMTSYSTEMS

In Kapitel 5 wurde das Konzept für ein System zur Erkennung von Outliern in Web Analytics Daten entwickelt. Vorgesehen ist der Einsatz des Local Outlier Factors auf dem Feature Aufrufe. Der Eingabeparameter `n_neighbors` wurde auf den Wert 75 gesetzt. Die Experimente zur Unterstützung dieser Entscheidungen wurden auf dem OWA-Datensatz und zwei synthetischen Datensätzen ausgeführt.

Im Rahmen der Evaluation soll die Leistungsfähigkeit des Systems anhand der OWA-2 und OWA-3-Datensätze verifiziert werden. Während die Genauigkeit der Algorithmen bei der Konzeption anhand der Average Precision gemessen wurde, wird bei der Evaluation als zusätzliche Metrik der AUC Score verwendet, um eine bessere Vergleichbarkeit mit den Ergebnissen verwandter Arbeiten zu ermöglichen.

Tabelle 6.1 fasst die Ergebnisse des Algorithmus unter gegebener Konfiguration für die produktiven Datensätze zusammen. Auf dem OWA-2 und OWA-3-Datensatz wurde eine deutlich geringere Genauigkeit erzielt, als auf dem OWA-Datensatz. Wie in Unterabschnitt 2.5.2 erwähnt, führt zufälliges Raten der Klassen zu einem AUC Score von 0,5. Diese Grenze wurde auf dem OWA-2-Datensatz mit einem AUC Score von 0,4985 knapp unterschritten, auf dem OWA-3-Datensatz mit 0,3667 sogar deutlich unterschritten. Eine derart geringe Genauigkeit ist unzureichend für ein Outlier Detection System.

Datensatz	**Average Precision**	**AUC Score**
OWA	0,8685	0,9222
OWA-2	0,3936	0,4985
OWA-3	0,3003	0,3667

Tabelle 6.1: Genauigkeit des Gesamtsystems auf den produktiven Datensätzen.

6.1 ANALYSE DER FALSCH KLASSIFIZIERTEN DATENPUNKTE

Um den Grund für die schlechte Performance auf den OWA-2 und OWA-3-Datensätzen zu finden, wurden die Outlier Rankings näher betrachtet. Zur Visualisierung wurden die Scores der Datenpunkte in Labels umgewandelt. Der Schwellenwert wurde so gewählt, dass in der Theorie ein optimales Ergebnis erzielt werden kann, also genau die tatsächlichen Outlier als solche klassifiziert werden. Dazu musste der Schwellenwert auf den Outlier-Anteil der betrachteten Datensätze festgelegt werden. Von Interesse für die Fehlersuche sind diejenigen Datenpunkte, die vom Algorithmus in eine falsche

Klasse eingeteilt wurden, also False Negatives und False Positives. Wie in Abbildung 6.1 zu sehen ist, treten alle False Negatives in einem Zeitraum zwischen Juli und Oktober 2018 auf. Die nicht erkannten Outlier in diesem Zeitraum sind auf einen Ausfall zurückzuführen, bei dem über 68 Tage hinweg keine oder nur sehr geringe Aufrufzahlen registriert wurden. Den betroffenen Datenpunkten wurde vom Algorithmus ein LOF von 1.0 zugewiesen. Breunig et al. zeigen, dass ein LOF von 1.0 in der Regel jenen Datenpunkten zugewiesen wird, die sich innerhalb eines Clusters befinden [5]. Ein wesentlicher Teil der Outlier wird also vom Algorithmus als Cluster interpretiert und daher falsch klassifiziert.

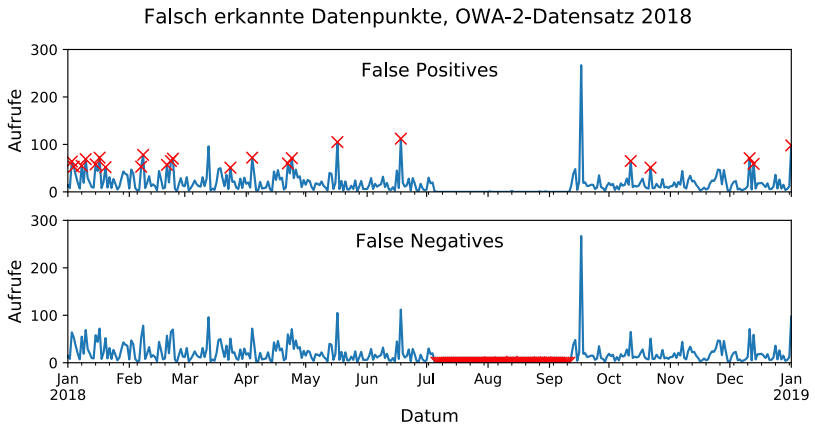


Abbildung 6.1: Datenpunkte des OWA-2-Datensatzes, die einer falschen Klasse zugeordnet wurden.

Anhand eines weiteren Experiments sollte geprüft werden, ab welcher Länge Ausfälle einen negativen Einfluss auf die Genauigkeit von LOF zeigen. Dazu wurden neue synthetische Datensätze generiert, die sich nur im Hinblick auf die Länge des simulierten Ausfalls unterscheiden. Im Folgenden werden die entstandenen Datensätze als SYN-3-Datensätze bezeichnet. Die SYN-3-Datensätze haben dieselbe Länge wie die produktiven Daten und enthalten neben den Ausfällen einige Punkt-Outlier. Abbildung 6.2 visualisiert die Average Precision und den AUC Score bei Anwendung von LOF auf die verschiedenen SYN-3-Datensätze. Zu erkennen ist, dass der AUC Score stark abfällt, wenn die Länge der Ausfälle den Wert von `n_neighbors` überschreitet, während die Average Precision bereits bei Ausfällen geringerer Länge sinkt.

Es kann angenommen werden, dass Ausfälle mit der kritischen Länge von über einem Monat in Web Analytics Daten selten auftreten. Dennoch sollte dieser Ausnahmefall behandelt werden. Dies kann in Form eines zusätzlichen Schrittes bei der Vorverarbeitung der Daten erfolgen, indem mehrere zusammenhängende Tage ohne Besucher direkt als Outlier klassifiziert werden.

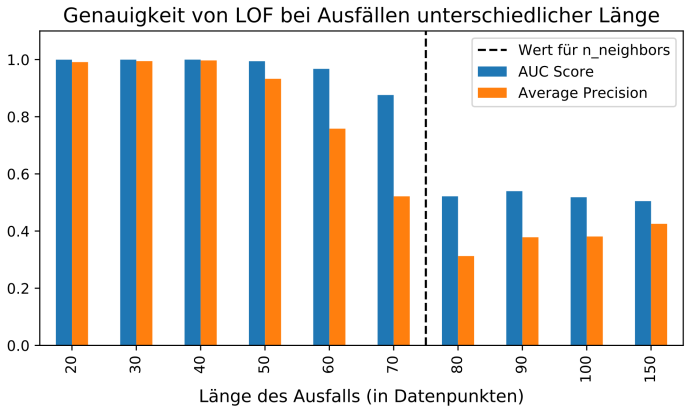


Abbildung 6.2: AUC Score und Average Precision von LOF auf den SYN-3-Datensätzen, bei steigender Länge der simulierten Ausfälle

6.2 GENAUIGKEIT NACH BEHANDLUNG VON AUSFÄLLEN

Wenn ein zusätzlicher Schritt zu Erkennung von längeren Ausfällen in die Vorverarbeitung der Daten eingefügt wird, müssen die entsprechenden Datenpunkte nicht mehr vom Algorithmus als Outlier erkannt werden. Indem die zu den Ausfällen gehörenden Datenpunkte vorübergehend als Normalverhalten klassifiziert werden, können die Auswirkungen auf die Genauigkeit des Gesamtsystems geschätzt werden.

Den in Tabelle 6.2 gezeigten Ergebnissen zufolge haben sich Average Precision und AUC Score deutlich verbessert. Die verbliebenen Outlier der Datensätze wurden zuverlässig erkannt.

Datensatz	**Average Precision**	**AUC Score**
OWA	0,8685	0,9222
OWA-2	0,9426	0,9976
OWA-3	0,9169	0,9974

Tabelle 6.2: Genauigkeit des Gesamtsystems auf den produktiven Datensätzen, ohne Betrachtung von Ausfällen

6.2.1 *Genauigkeit im Vergleich zu Standardparametern*

Während der Konzeption wurde die Average Precision von LOF anhand des OWA-Datensatzes optimiert. Tabelle 6.3 zeigt neben der prozentualen Verbesserung dieses Datensatzes auch die Verbesserungen für die OWA-2 und OWA-3-Datensätze. Als Basis für den Vergleich wurde LOF mit denen vom Framework gegebenen Standardwerten auf allen Feature Kombinationen ausgeführt, wobei nur die besten absoluten Ergebnisse pro Datensatz und Metrik betrachtet wurden. Somit gibt die Tabelle die minimal erzielten Verbesserungen an.

Auf allen Datensätzen konnte ein positiver Effekt festgestellt werden, der jedoch unterschiedlich stark ausfiel. Mit einer Verbesserung um 127,38% in der Average Precision konnte auf dem OWA-2-Datensatz die höchste Steigerung der Genauigkeit erreicht werden. Die Average Precision auf dem OWA und OWA-3-Datensatz verbesserte sich jeweils um ungefähr 20%.

Es konnte beobachtet werden, dass sich Veränderungen in der Konfiguration deutlich stärker auf die Average Precision auswirkten, als auf den AUC Score. Die prozentualen Steigerungen waren in Average Precision gemessen mindestens um den Faktor 8 höher, als die entsprechenden Verbesserungen im AUC Score.

Datensatz	**Optimale Features**	**Verbesserung (AP)**	**Verbesserung (AUC)**
OWA	A, B	19,46%	2,14%
OWA-2	A	127,38%	8,47%
OWA-3	A, B, D (für AP)	22,07%	
	A (für AUC)		2,72%

Tabelle 6.3: Prozentuale Erhöhung der Genauigkeit des Gesamtsystems im Vergleich zu den vom Framework gegebenen Standardwerten der Eingabeparameter. Für die Ergebnisse unter den Standardwerten wurde jeweils die Feature-Kombination mit der höchsten erzielten Genauigkeit gewählt, die Ausfälle wurden nicht betrachtet.

6.3 BEWERTUNG

Der AUC Score hat die wichtige Eigenschaft, dass er äquivalent zur Wahrscheinlichkeit ist, mit der ein zufällig gewählter Outlier im Ranking über einem zufälligen normalen Datenpunkt steht [10]. Für die OWA-2 und OWA-3-Datensätze beträgt diese Wahrscheinlichkeit über 99%, für den OWA-Datensatz ca. 92%. Dies ist ein gutes Ergebnis, wie ein Vergleich mit anderen Evaluationen zeigt.

In der von Goldstein et al. durchgeführten Evaluation [13] erreichte LOF auf zwei Datensätzen einen durchschnittlichen AUC Score von über 0,98, auf den anderen acht Datensätzen dagegen unter 0,9. Dabei liegt der maximale Outlier-Anteil der Datensätze bei 11%, also unter dem Outlier-Anteil des OWA-Datensatzes. Ein AUC Score von über 0,99 zählt unter den 19 verschiedenen Algorithmen zu den besten Ergebnissen, und auch 0,92 stellt eine überdurchschnittliche Genauigkeit dar. Auf vier Datensätzen wurde von keinem Algorithmus eine höhere Genauigkeit erzielt.

7 ZUSAMMENFASSUNG UND AUSBLICK

Im Rahmen der Arbeit wurde die Konzeption eines Outlier Detection Systems für Web Analytics Daten beschrieben und das entstandene Gesamtsystem evaluiert. Als Datengrundlage dienten Besucher-Statistiken der Unternehmenswebsites von OdiSys, die von eindeutig identifizierbarem Noise bereinigt und anschließend aggregiert wurden. Die entstandenen Datensätze enthalten für jeden Tag die Aufrufe, Besucher und durchschnittlichen Aufrufe pro Besucher. Zur Unterstützung der Konzeption wurde ein Generator entwickelt, der numerische Eingabedaten nachbilden kann, wobei grundlegende statistische Eigenschaften der Features erhalten bleiben. Während derzeit nur ganzzahlige Zeitreihen als Eingabe unterstützt werden, kann der Generator erweitert werden, um Dezimalzahlen zu unterstützen. Solange die Eingabe die Eigenschaften von Markov-Prozessen erfüllt, kann der Generator auch in anderen Anwendungsgebieten eingesetzt werden. In die generierten Daten können gezielt Outlier eingefügt werden, die automatisch mit Labeln versehen werden. Auf diese Weise entstanden zwei synthetische Datensätze, die zur Unterstützung der Konzeption dienten.

Als Hauptbeitrag der Arbeit wurde das Konzept eines Systems entwickelt, das Outlier in den vorliegenden Datensätzen zuverlässig erkennen kann. Ziel war die Maximierung der in Average Precision gemessenen Genauigkeit, unter Beibehaltung einer möglichst hohen Robustheit.

Für die Unsupervised Outlier Detection Algorithmen k-NN, LOF und HBOS wurden experimentell die besten Feature-Kombinationen und Werte ihrer Eingabeparameter bestimmt. k-NN erzielte die höchste Genauigkeit auf dem OWA-Datensatz, wenn als Feature nur die Aufrufe betrachtet wurden, bei `n_neighbors = 75` und der Methode `median`. HBOS zeigte auf der Kombination von Aufrufen und durchschnittlichen Aufrufen pro Besucher bei einem `n_bins` Wert von 142 die höchste Genauigkeit.

Insgesamt erreichte LOF mit einer Average Precision von $0,8685$ die höchste Genauigkeit. Als Feature dienten die Aufrufe, während der Parameter `n_neighbors` auf 75 gesetzt wurde. Im Hinblick auf die Robustheit konnte LOF auf dem OWA-Datensatz und einem der synthetischen Datensätze überzeugen, wenn ein lokaler Wertebereich ausgewertet wurde. Daher wurde LOF als Outlier Detection Algorithmus für das Gesamtsystem ausgewählt.

Durch die anschließende Evaluation des Gesamtsystems anhand von zwei weiteren produktiven Datensätzen konnte ein Problem der bestehenden Konfiguration gefunden werden. Zwei Ausfälle, bei denen die Aufrufe über einen Zeitraum von 68 Tagen sehr ähnliche Werte annahmen, bekamen einen sehr niedrigen LOF zugewiesen, da sie als Cluster erkannt wurden, was zu einer hohen Zahl von False Negatives führte. Derartige Muster müssen

bereits während der Vorverarbeitung der Datensätze erkannt und markiert werden, um eine Funktionsfähigkeit des Systems sicherzustellen.

Wenn dies bei der Auswertung berücksichtigt wurde, konnte auf allen Datensätzen eine prozentuale Verbesserung der Genauigkeit im Vergleich zur besten standardmäßigen Konfiguration verzeichnet werden. Die Optimierung anhand von einem produktiven Datensatz führte also auch zu einer Verbesserung der Genauigkeit auf den anderen produktiven Datensätzen.

Die Vorgehensweise bei der Konzeption kann in Zukunft auf andere Unsupervised Outlier Detection Algorithmen übertragen werden, um deren bestmögliche Konfiguration zu bestimmen. Die Ergebnisse können mit denen der Evaluation verglichen werden, um potenziell einen Algorithmus mit noch höherer Genauigkeit oder Robustheit zu finden. Zu diesem Zweck können auch Kombinationsmöglichkeiten von mehreren Algorithmen betrachtet werden.

Im Anschluss kann die eigentliche Implementierung des Systems erfolgen. Geplant ist auf der einen Seite die Visualisierung der Outlier, auf der anderen Seite die automatische Bereinigung der Daten. Dafür muss ein Schwellenwert gefunden werden, der die Outlier Scores in Label umwandelt.

Ausgehend von den aggregierten Datenpunkten, die als Outlier markiert wurden, können Muster in den Rohdaten erkannt werden. Beispielsweise könnten IP-Adressen oder Regionen als verdächtig markiert werden, wenn sie oft in Verbindung mit Outliern auftreten. Dies könnte eine zuverlässigere Bereinigung der Daten ermöglichen, oder sogar als zusätzliches Feature für die Outlier Detection in Betracht gezogen werden.

LITERATUR

[1] C. C. Aggarwal. *Outlier Analysis*. Springer International Publishing, 2017. DOI: 10.1007/978-3-319-47578-3.

[2] C. C. Aggarwal und P. S. Yu. "Outlier Detection for High Dimensional Data". In: *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*. SIGMOD '01. New York, NY: Association for Computing Machinery, 2001, S. 37–46. DOI: 10.1145/375663.375668.

[3] F. Angiulli und C. Pizzuti. "Fast outlier detection in high dimensional spaces". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Bd. 2431. 2002, S. 15–27. DOI: 10.1007/3-540-45681-3_2.

[4] N. D. Bokde, A. Feijóo, N. Al-Ansari und Z. M. Yaseen. "A Comparison Between Reconstruction Methods for Generation of Synthetic Time Series Applied to Wind Speed Simulation". In: *IEEE Access* 7 (2019), S. 135386–135398. DOI: 10.1109/ACCESS.2019.2941826.

[5] M. M. Breunig, H. P. Kriegel, R. T. Ng und J. Sander. "LOF: identifying density-based local outliers". In: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data - SIGMOD '00*. Bd. 29. 2. New York, New York, USA: ACM Press, 2000, S. 93–104. DOI: 10.1145/342009.335388.

[6] K. Brokish und J. Kirtley. "Pitfalls of modeling wind power using Markov chains". In: *2009 IEEE/PES Power Systems Conference and Exposition*. März 2009, S. 1–6. DOI: 10.1109/PSCE.2009.4840265.

[7] G. O. Campos, A. Zimek, J. Sander, R. J.G.B. Campello, B. Micenková, E. Schubert, I. Assent und M. E. Houle. "On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study". In: *Data Mining and Knowledge Discovery* 30.4 (2016), S. 891–927. DOI: 10.1007/s10618-015-0444-8.

[8] V. Chandola, A. Banerjee und V. Kumar. "Anomaly detection: A survey". In: *ACM Computing Surveys* 41.3 (Juli 2009), S. 1–58. DOI: 10.1145/1541880.1541882.

[9] L. Fahrmeir, C. Heumann, R. Künstler, I. Pigeot und G. Tutz. "Zeitreihen". In: *Statistik: Der Weg zur Datenanalyse*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, S. 503–527. DOI: 10.1007/978-3-662-50372-0_14.

[10] T. Fawcett. "An introduction to ROC analysis". In: *Pattern Recognition Letters* 27.8 (Mai 2006), S. 861–874. DOI: 10.1016/J.PATREC.2005.10.010.

[11] A. Feijóo und D. Villanueva. "Assessing wind speed simulation methods". In: *Renewable and Sustainable Energy Reviews* 56 (2016), S. 473 –483. DOI: https://doi.org/10.1016/j.rser.2015.11.094.

[12] M. Goldstein und A. Dengel. "Histogram-based Outlier Score (HBOS): A fast Unsupervised Anomaly Detection Algorithm". In: *Poster and Demo Track of the 35th German Conference on Artificial Intelligence (KI-2012)*. 2012, S. 59–63.

[13] M. Goldstein und S. Uchida. "A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data". In: *PLOS ONE* 11.4 (Apr. 2016). DOI: 10.1371/journal.pone.0152173.

[14] D. M. Hamby. "A review of techniques for parameter sensitivity analysis of environmental models". In: *Environmental Monitoring and Assessment* 32.2 (Sep. 1994), S. 135–154. DOI: 10.1007/BF00547132.

[15] D. M. Hamby. "A comparison of sensitivity analysis techniques". In: *Health Physics* 68.2 (Feb. 1995), S. 195–204. DOI: 10.1097/00004032-199502000-00005.

[16] D. M. Hawkins. "Introduction". In: *Identification of Outliers*. Dordrecht: Springer Netherlands, 1980, S. 1–12. DOI: 10.1007/978-94-015-3994-4_1.

[17] F. O. Hocaoglu, O. N. Gerek und M. Kurban. "The Effect of Markov Chain State Size for Synthetic Wind Speed Generation". In: *Proceedings of the 10th International Conference on Probablistic Methods Applied to Power Systems*. Mai 2008, S. 1–4. URL: https://ieeexplore.ieee.org/document/4912620.

[18] B.O. Ngoko, H. Sugihara und T. Funaki. "Synthetic generation of high temporal resolution solar radiation data using Markov models". In: *Solar Energy* 103 (2014), S. 160 –170. DOI: https://doi.org/10.1016/j.solener.2014.02.026.

[19] S. Ramaswamy, R. Rastogi und K. Shim. "Efficient algorithms for mining outliers from large data sets". In: *SIGMOD Record (ACM Special Interest Group on Management of Data)* 29.2 (2000), S. 427–438. DOI: 10.1145/335191.335437.

[20] T. Saito und M. Rehmsmeier. "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets". In: *PLoS ONE* 10.3 (März 2015). DOI: 10.1371/journal.pone.0118432.

[21] A. Shamshad, M.A. Bawadi, W.M.A. Hussin, T.A. Majid und S.A.M. Sanusi. "First and second order Markov chain models for synthetic generation of wind speed time series". In: *Energy* 30.5 (2005), S. 693 –708. DOI: https://doi.org/10.1016/j.energy.2004.05.026.

[22] S. N. Shirazi, A. Gouglidis, K. N. Syeda, S. Simpson, A. Mauthe, I. M. Stephanakis und D. Hutchison. "Evaluation of Anomaly Detection techniques for SCADA communication resilience". In: *2016 Resilience Week (RWS)*. IEEE, Aug. 2016, S. 140–145. DOI: 10.1109/RWEEK.2016.7573322.

[23] H. Wang, M. J. Bah und M. Hammad. "Progress in Outlier Detection Techniques: A Survey". In: *IEEE Access* 7 (2019), S. 107964–108000. DOI: 10.1109/ACCESS.2019.2932769.

[24] R. Xu und D. Wunsch. "Survey of clustering algorithms". In: *IEEE Transactions on Neural Networks* 16.3 (Mai 2005), S. 645–678. DOI: 10.1109/TNN.2005.845141.

[25] Y. Zhao, Z. Nasrullah und Z. Li. "PyOD: A Python Toolbox for Scalable Outlier Detection". In: *Journal of Machine Learning Research* 20.96 (2019), S. 1–7. URL: http://jmlr.org/papers/v20/19-011.html.