LAPPEENRANTA-LAHTI UNIVERSITY OF TECHNOLOGY LUT
School of Engineering Science
Degree Program in Industrial Engineering and Management
Business Analytics

Noona Jantunen

DETECTING DATA QUALITY ISSUES IN CATEGORICAL DATA THROUGH ANOMALY DETECTION

Master's Thesis

Examiners: Professor Pasi Luukka
Junior Researcher Mahinda Mailagaha Kumbure

ABSTRACT

Lappeenranta-Lahti University of Technology LUT
School of Engineering Science
Degree Program in Industrial Engineering and Management

Noona Jantunen

Detecting data quality issues in categorical data through anomaly detection

Master's Thesis
2022
86 pages, 16 figures, 10 tables and 2 algorithms
Examiners: Professor Pasi Luukka and Junior Researcher Mahinda Mailagaha Kumbure

Keywords: data quality, anomaly detection, categorical data

Organizations have increasingly started to understand that data are one of their most important business assets. Nevertheless, for the data to be valuable, it has to be of good quality. Anomaly detection is one approach for detecting possible data quality issues without using pre-defined rules or examining the data manually. The research on anomaly detection is heavily focused on numerical data, although categorical data are ubiquitous in practical applications. Several scholars have identified the issue and proposed anomaly detection methods specifically designed for categorical data.

The objective of this study was to compare and assess anomaly detection methods for detecting potential data quality issues in categorical data. The study discusses the concepts of data quality and anomaly detection, and further defines important considerations in selecting an anomaly detection method for categorical data. A literature review was conducted to survey potential methods. Selected anomaly detection methods were then applied to case data obtained from a case company.

The findings of the study suggests that many anomaly detection methods for categorical data are complex, and some methods define an anomaly differently compared to other methods. In this study, the evaluated algorithms detected rather different records as anomalies, and therefore it is assumed to be important to select an appropriate algorithm for the intended use. At least one of the evaluated algorithms showed potential for detecting data quality issues in categorical data. However, further analysis is required to determine the feasibility of the methods in the specific context by investigating whether the detected anomalies are actual data quality issues or abnormal but legitimate data records. If the methods prove feasible, the case company can use the methods for detecting data quality issues and can eventually improve data quality. Nonetheless, this study provides understanding of anomaly detection in categorical data. In addition, the findings of the study can be utilized in evaluating possible anomaly detection solutions provided by vendors regardless of company or industry.

TIIVISTELMÄ

Lappeenrannan-Lahden teknillinen yliopisto LUT
School of Engineering Science
Tuotantotalouden koulutusohjelma

Noona Jantunen

Datan laatuongelmien tunnistaminen kategorisesta datasta poikkeamantunnistusta käyttäen

Diplomityö
2022
86 sivua, 16 kuvaa, 10 taulukkoa ja 2 algoritmia
Tarkastajat: Professori Pasi Luukka ja Nuorempi tutkija Mahinda Mailagaha Kumbure

Hakusanat: datan laatu, poikkeamien tunnistaminen, kategorinen data

Yritykset ovat enenevässä määrin alkaneet ymmärtää, että data on yksi niiden tärkeimmistä liiketoiminnan voimavaroista. Jotta data olisi arvokasta, on sen kuitenkin oltava hyvälaatuista. Poikkeamien tunnistaminen on yksi tapa havaita mahdolliset datan laatuongelmat ilman ennalta määritettyjä sääntöjä tai datan manuaalista tutkimista. Poikkeamien tunnistamisen tutkimus keskittyy vahvasti numeeriseen dataan, vaikka kategorinen data on hyvin yleistä käytännön sovelluksissa. Useat tutkijat ovat tunnistaneet ongelman ja ehdottaneet poikkeamien tunnistamismenetelmiä, jotka on suunniteltu erityisesti kategoriselle datalle.

Tämän tutkimuksen tavoitteena oli verrata ja arvioida poikkeamien tunnistamismenetelmiä mahdollisten datan laatuongelmien havaitsemiseksi kategorisessa datassa. Tutkimus käsittelee datan laadun ja poikkeamien tunnistamisen käsitteitä ja määrittelee, mitä tulee ottaa huomioon valittaessa poikkeamien tunnistamismenetelmää kategoriselle datalle. Potentiaalisia menetelmiä kartoitettiin kirjallisuuskatsauksen avulla. Valittuja poikkeamien tunnistamismenetelmiä sovellettiin sen jälkeen case-yritykseltä saatuun case-dataan.

Tutkimustulokset viittaavat siihen, että monet kategorisen datan poikkeamien tunnistamismenetelmät ovat monimutkaisia, ja eri menetelmien määrittelyt poikkeamalle eroavat toisistaan. Tässä tutkimuksessa arvioidut algoritmit havaitsivat poikkeamiksi melko erilaisia tietueita. Täten voidaan olettaa, että on tärkeää valita algoritmi, joka sopii käyttötarkoitukseensa. Ainakin toinen arvioiduista algoritmeista osoitti potentiaalia havaita datan laatuongelmia kategorisessa datassa. Analyysien syventäminen on kuitenkin tarpeen menetelmien käyttökelpoisuuden määrittämiseksi kyseisessä kontekstissa; tulee tutkia, ovatko havaitut poikkeamat todellisia datan laatuongelmia vai poikkeavia, mutta kelvollisia, tietueita. Jos menetelmät osoittautuvat käyttökelpoisiksi, case-yritys voi käyttää menetelmiä datan laatuongelmien havaitsemiseen ja lopulta datan laadun parantamiseen. Joka tapauksessa tämä tutkimus auttaa ymmärtämään poikkeamien tunnistamista kategorisessa datassa. Tutkimuksen tuloksia voidaan myös hyödyntää toimittajien mahdollisesti tarjoamien poikkeamien tunnistamisratkaisujen arvioimiseen yrityksestä ja toimialasta riippumatta.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my thesis supervisor from LUT University, Professor Pasi Luukka, for guidance and feedback during this project. I would also like to thank my supervisor, and everyone else involved, from the case company for the support and interesting discussions around the thesis topic.

Finally, I want to thank my family and friends for always supporting me throughout my studies.

Espoo, 21.1.2022
Noona Jantunen

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

ABBREVIATIONS

AD	Alternating Decision
APD	Anomaly Pattern Detection
AUC-ROC	Area Under the Curve of Receiver Operating Characteristic
AVF	Attribute Value Frequency
BSVC	Bidirectional Selective Value Coupling
CA	Conditional Algorithm
CBRW	Coupled Biased Random Walks
CNB	Common-Neighbor-Based
COD	Contextual Outlier Detection
COSH	Coupled Outlier Scoring of High-dimensional data
CUOT	Coupled Unsupervised OuTlier detection
ELM	Extreme Learning Machines
ETL	Extract, Transform, Load
FI	Frequent Itemset
FNADI-OD	Frequent Non-Almost Derivable Itemsets
FNDI-OD	Frequent Non-Derivable Itemsets
FPOF	Frequent Pattern Outlier Factor
GA	Greedy Algorithm
HOT	Hypergraph-based Outlier Test
HR	Human Resources
IG	Information Gain
IID	Independent and Identically Distributed
IT	Information Technology
ITB-SP	Information-Theory-Based Single-Pass

ITB-SS	Information-Theory-Based Step-by-Step
k-LOF	k-Local Anomalies Factor
KPI	Key Performance Indicator
LOADED	Link-based Outlier and Anomaly Detection in Evolving Data sets
LOF	Local Anomalies Factor
LSA	Local-Search heuristic-based Algorithm
MDL	Minimum Description Length
NBNDI-OD	Negative Border of frequent Non-Derivable Itemsets
NDI	Non-Derivable Itemset
ODMAD	Outlier Detection for Mixed Attribute Datasets
POP	Partial Outlierness Propagation
RBF	Radial Basis Function
ROAD	Ranking-based Outlier Analysis and Detection
SCAN	Skip-gram architecture on a biased value Coupling-based vAlue Network
SDLE	Sequentially Discounting Laplace Estimation
SDRW	Subgraph Densities-augmented Random Walks
SRA	Spectral Ranking method for Anomaly detection
UA	Unsupervised Approach
WDOD	Weighted Density-based Outlier Detection
WSARE	What’s Strange About Recent Events

1 INTRODUCTION

This chapter defines the research objective and gives an overview to the topic of this thesis. First, the background for the study is presented, followed by describing the objective and research questions. In addition, the chapter defines the scope of the study, describes how the research was conducted, and finally presents the structure of this report.

1.1 Background

The value of data increases all the time, as new ways to utilize data are introduced to help organizations succeed (Olson 2003). Today, data are an important business asset for any organization (Mohr and Hürtgen 2018). However, for the data to be valuable, it has to be of good quality; data quality is the foundation of the value of organization's data assets. According to a recent Gartner survey, poor data quality costs organizations on average $12.8 million per year. Furthermore, as business environments digitize, the costs are estimated to rise. (Jain et al. 2020)

Data quality is a broad concept and there exist variety of different data quality issues and methods to detect them. Detecting data quality issues in numerical data through anomaly detection has been studied by several scholars, such as Dai et al. (2017), Liu et al. (2019), Vilenski et al. (2019) and Jesus et al. (2021). However, there is very limited research on using anomaly detection methods to detect data quality issues in categorical data.

Overall, the research on anomaly detection is heavily focused on the methods designed for numerical data. That is likely to be because there exists no inherent similarity measure for categorical data, which makes the problem of detecting anomalies in categorical data challenging. (Suri et al. 2012; Wu and Wang 2013) Therefore, in practice, categorical attributes often get ignored or are encoded into numerical attributes to make use of the well-known methods for numerical data. In many applications, encoding categorical attributes is not feasible or does not produce reasonable results. (Aggarwal 2013, p. 200; Nian et al. 2016)

In real-life applications, categorical data are ubiquitous, and thus, have to be handled appropriately. Several scholars have recognized this issue and have proposed anomaly detection methods for categorical data. This thesis studies those methods and applies selected methods to real-life datasets with the aim to detect potential data quality issues.

1.2 Objective and scope

The objective of the study is:

To compare and assess anomaly detection methods for detecting potential data quality issues in categorical data

The study gives reader an overview of data quality, the reasons for poor data quality and the actions needed to ensure that data are of high quality. Then, the concept of anomaly detection is introduced and anomaly detection in categorical data is discussed. To reach the objective of the study, previous literature is examined to find different methods for detecting anomalies, i.e., potential data quality issues, in categorical data. Two anomaly detection methods are then introduced, tested and evaluated.

The research objective can be divided into following research questions:

1. *What has to be taken into consideration when selecting an anomaly detection method for categorical data?*
2. *What are the state-of-the-art methods for anomaly detection in categorical data?*
3. *What kind of anomalies can be found from the case data?*

This study is delimited to the algorithms of anomaly detection methods, while the technical architecture of the methods is excluded from the study. Due to the lack of labels in the case data and for a wider applicability of this study, anomaly detection algorithms are delimited to unsupervised algorithms. In addition, the empirical part of the study is delimited to existing publicly available algorithms. Therefore, the scope of the study does not include implementing any algorithms not available for public, nor making any major changes to existing

implementations of the algorithms. Although the study focuses on anomaly detection from the data quality perspective, the evaluation of whether the detected anomalies are data quality issues or abnormal but legitimate data records is out of the scope of this study.

As a limitation of this study, since the case data do not contain data labels, it is not possible to measure the performance of the anomaly detection algorithms in terms of whether they detect true anomalies or not. Furthermore, a limitation, that has to be considered when interpreting the results of this study, is that the algorithms are only applied to the limited set of case company data, and their feasibility to any other data is not validated. An assumption made in this study, is that the case data are mostly of good quality.

1.3 Research methods

The research methods of this study are literature review and empirical case study. To build a theoretical framework for the study, scientific publications and industry literature were studied. Literature review was then conducted to study existing methods for detecting anomalies in categorical data. Since the literature search for detecting anomalies in categorical data in the context of data quality did not yield many relevant results, the literature review was not limited to the context of data quality, but anomaly detection in categorical data in general. The case study was conducted by applying selected anomaly detection algorithms in practice to case data.

1.4 Thesis structure

This thesis is divided into seven chapters. The first chapter introduces the background for the thesis, defines the objective and scope of the study, and presents the research methods. The second chapter gives an overview of data quality by first defining what data quality means, followed by discussing the reasons for poor data quality and how to ensure that data are of high quality. The third chapter studies anomaly detection. It begins by defining an anomaly, proceeds to presenting different approaches to anomaly detection, and finally discusses the problem of detecting anomalies in categorical data.

The fourth chapter is a literature review on previous studies of anomaly detection in categorical data. The fifth chapter then introduces selected algorithms from the conducted literature review to be tested in the case study. The sixth chapter applies the selected algorithms to case data and discusses the results. Finally, the seventh chapter concludes the findings of the study.

2 DATA QUALITY

Data quality has gained research interest of numerous scholars and data quality practitioners for many years. Yet, recently, it has drawn more attention as organizations are starting to understand its importance – and especially the costs associated with poor data quality. However, many organizations still lack adequate data quality and proper data quality practices.

Data quality is often perceived to be part of information quality. The concepts of data and information are many times further presented as parts of a pyramid – the knowledge pyramid or Data Information Knowledge Wisdom pyramid. Data can be seen to provide the raw material (simple facts) for the information product. (Sebastian-Coleman 2013, p. 14)

This chapter first defines data quality and introduces some of the most common dimensions of data quality to make the concept of data quality more concrete. Next, the chapter discusses the reasons for poor data quality – how data quality issues are created. Finally, the chapter discusses how to improve data quality.

2.1 Definition of data quality

Data quality scholars are rather unanimous about the definition of data quality. Wang and Strong (1996 p. 6) define data quality as "data that are fit for use by data consumers". Olson (2003 p. 24) states that data quality is the extent to which data satisfies the requirements of its intended use. Sebastian-Coleman (2013 p. 40) agrees that data quality is the "degree to which data meets the expectations of data consumers, based on their intended uses of the data". Both Olson (2003 p. 24) and Sebastian-Coleman (2013 p. 40) emphasize that data quality is directly related to the intended use of the data, meaning that the same dataset might be considered to be of high quality for one use case but of low quality for another use case.

Data quality is often represented by a set of characteristics – data quality dimensions (Fu and Easton 2017). Wang and Strong (1996, p. 6) define data quality dimension as a "set of data quality attributes that represent a single aspect or construct of data quality". Sebastian-Coleman (2013, p. 40) states that the measurable and quantifiable aspects of data quality are widely

accepted to be referred to as data quality dimensions. To define expectations for or to measure the quality of a dataset, a set of data quality dimensions can be introduced. (Sebastian-Coleman 2013 p. 40)

Many different sets of data quality dimensions have been proposed in literature. There is no universally accepted set of most important dimensions. Rather, there are several different proposals, of which definitions of the dimensions often differ. Moreover, some dimensions may be introduced as subdimensions in some data quality dimension proposals. However, most of the data quality dimension proposals include some version of accuracy and validity, completeness, consistency, and currency or timeliness (Sebastian-Coleman 2013, p. 40). The dimensions most often occurring in literature are next introduced shortly.

Accuracy

Wang and Strong (1996, p. 31) define accuracy as "the extent to which data are correct, reliable, and certified free of error". Moreover, Batini and Scannapieco (2016, p. 23) affirm accuracy as the closeness between a presented data value and a data value that is considered to correctly represent the real-world phenomenon that also the presented data value aims to represent.

Completeness

Wang and Strong (1996, p. 32) affirm completeness as "the extent to which data are of sufficient breadth, depth, and scope for the task at hand". Lee et al. (2006) state that completeness can be viewed from three different perspectives: schema completeness, column completeness, and population completeness. Schema completeness refers to the degree to which entities and attributes are not missing from the schema. Column completeness refers to the degree to which there are missing values in a table column. On the other hand, population completeness refers to the degree to which members of the population occur in the data if they are supposed to occur. (Lee et al. 2006)

Consistency

Lee et al. (2006) state that consistency is another data quality dimension that can be looked at from different perspectives. First, consistency can be viewed as consistency of redundant data in one or more tables. Second, consistency can be viewed as consistency between two related data elements, for example postal code and city should be consistent. Third, consistency can be viewed as consistency of format for the same data element across multiple tables. (Lee et al. 2006) Batini and Scannapieco (2016) define consistency simply as a dimension capturing the violation of semantic rules.

Timeliness

Timeliness is sometimes also referred to as currency. Wang and Strong (1996, p. 31) define timeliness as "the extent to which the age of the data is appropriate for the task at hand". Lee et al. (2006) and many other scholars agree with that definition.

2.2 Reasons for poor data quality

Before addressing the problem of detecting data quality issues or ultimately ensuring high data quality, it is necessary to understand the sources that generate data quality issues. There are several reasons for the occurrence of data quality issues. In general, they can be categorized into four categories based on the area where the issue occurs: initial data entry, data decay, moving and restructuring data, as well as using data, as shown in Figure 1. The first three are the sources of data quality issues within databases, while the fourth one considers a broader scope of data quality issues and represents the area of issues in the information products produced using the data. (Olson 2003, p. 43)

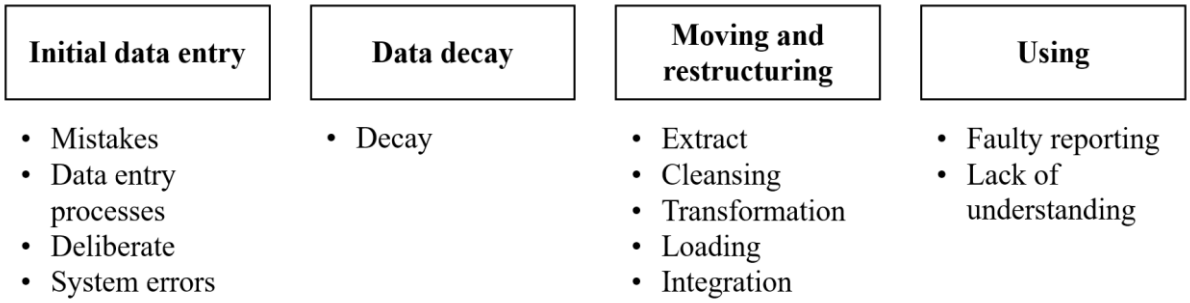

Figure 1. Areas in which data quality issues occur (adapted from Olson 2003, p. 43)

Initial data entry

Singh and Singh (2010) state that a typical cause for data quality issues is having poor quality data in the data source systems. Furthermore, Olson (2003, p. 44) argues that many people believe data quality issues are always generated by initially entering wrong data. Although it is a typical source of data quality issues, it is not the only one. Data quality issues occurring in the creation of the data can be caused by mistakenly entering wrong data, flawed data entry processes, system errors, or even deliberately entering wrong data. (Olson 2003, p. 44)

Singh and Singh (2010) suggest that in addition to initially entering wrong data, the data can also be updated erroneously, and the errors can be created by a human or a computer system. Olson (2003, p. 44) remarks that one can judge their systems in terms of the aforementioned reasons for data quality issues; some systems are designed to allow entry of erroneous data, while others are designed to promote data quality.

Data accuracy decay

Another reason for the existence of data quality issues in the source systems is failing to update the data (Singh and Singh 2010). Even though correct data was initially entered successfully, the data can become incorrect over time. This means that the data value itself does not change, but the actual value, that the data represents, changes – thereby leading to inaccurate data if no updates are made. (Olson 2003, p. 50)

Not all data objects are subject to value accuracy decay. In general, data that defines an object is not subject to decay, while data that provides additional information about the object may decay. As an example, consider personal information of an employee in a Human Resources (HR) system. People move, change phone numbers and their marital statuses change. Thereby, addresses, phone numbers and marital statuses are subject to decay. In contrast, employee ID is rarely subject to decay. (Olson 2003, p. 50)

Moving and restructuring data

Another common area to create data quality issues into data, that is initially of good quality, is in the processes of moving and restructuring data. The processes typically consist of extracting data from operational databases and taking it for example into data warehouses. The processes might also include restaging the data for the use through corporate portals. Generating data quality issues in the process of moving and restructuring data is often overlooked, even though its contribution to decreasing data quality is significant. When the data in a data warehouse is wrong, it might be the result of flawed data movement processes, and not always wrong data in the source databases. (Olson 2003, p. 52)

Extract, Transform, Load (ETL) processes are a typical approach for moving and restructuring data. Moreover, data cleansing may be performed along with moving the data from a system to another. In the process of moving the data, data quality issues can occur in any of the aforementioned steps: extraction, data cleansing, transformation, or loading. (Olson 2003, p. 52–62) In fact, Singh and Singh (2010) argue that most data quality issues originate from ETL processes making them the most critical phase for data quality. Furthermore, Olson (2003, p. 52–62) annotates that data integration projects taking data from source systems to the use of new applications face all the same issues. Consequently, the processes of data integration have to be properly executed to have the data in a form that the target application understands – to avoid data quality issues. (Olson 2003, p. 52–62)

Using data

The final area in which data quality issues can occur is in using the data – in the process of putting it into business objects such as reports, as well as in the use of the business objects by business professionals. Accordingly, even if the data itself is of good quality, but users do not understand its meaning or its context, the users may interpret or use the data incorrectly – resulting into wrong information. The ultimate reason for that is often the lack of good, accessible metadata repository. Most organizations do not have that in place, and even the ones that have, often lack an appropriate mechanism for other than Information Technology (IT) professionals to access that information. (Olson 2003, pp. 62–63)

2.3 Ensuring high data quality

To improve data quality, the first action is to realize current status. Gartner presents data quality maturity scale and actions defining the maturity level and presenting the improvement actions to achieve higher maturity level. At the lowest maturity level, an organization understands the impact of data on Key Performance Indicators (KPIs) and data quality improvements on business outcome. The second maturity level adds frequent data profiling and data quality dashboards. Whereas, assigning business accountability with relevant follow-ups and procedures brings an organization to the third maturity level. Finally, when data quality improvement is embedded into organizational culture, the organization has reached the data quality maturity. (Sakpal 2021)

Olson (2003, p. 14) argues that data quality problems are often viewed as isolated instances, rather than symptoms. He adds that it is natural, since organizations often do not want to believe they have problems before they face them. Therefore, when it comes to data quality, they tend to be reactive, not proactive. However, ensuring significant improvements in data quality demand proactive activities. Data quality improvement has to be considered as a long-term and continuous activity. Furthermore, it is important that even when high data quality is achieved, it has to be maintained – data quality improvement is not a one-time project. (Olson 2003, p. 14–15)

Many of the data quality issues require a long time to be fixed. The systems initially gathering the data are the main place in which to fix the issues. For example, rebuilding the systems to ensure higher quality data might take several years to complete. Even though, in general, ensuring high data quality is a long-term topic, returns can also be achieved in the short term. (Olson 2003, p. 15) In parallel with the long-term improvement, Olson (2003, p. 15) argues that short-term improvement can be achieved for example by filtering of input data, cleansing of data in databases, as well as by educating data consumers on the quality of the data. On the other hand, Redman (2013) argues that cleansing of old data may not add any value, and rather, the root causes for the issues should be identified and fixed.

As discussed earlier, data quality issues emerge at a number of phases and for several different reasons. To achieve high-quality data, the entire spectrum of opportunities for data quality issues has to be understood and taken into account. (Olson 2003, p. 64) When it comes to data quality issues generated in the initial data entry, the key is paying attention to the system design, making sure it promotes high data quality, as well as testing it for typical errors (Olson 2003, p. 44–49).

To mitigate data quality decay, database designers have to define in metadata that a data element is subject to decay, and design processes to verify and update the information. For example, an HR system could request employees to verify and update their personal information every time they change anything in the system. In addition, if no changes are made for a year, the system could request a separate review of the information. (Olson 2003, pp. 50–51)

In terms of avoiding the generation of data quality issues in moving and restructuring data, there are several things to take into consideration. When data cleansing is performed before moving the data into the data warehouses, it is often made dirtier rather than cleaner. For example, automatically rejecting incorrect data values that are still recognizable misspellings leads to dirtier data – this should be avoided. (Olson 2003, pp. 52–62)

Furthermore, databases are often designed to meet the requirements of the initiating application, while the requirements of different systems vary. Therefore, it is important to have a complete understanding and up-to-date documentation of the data and database designs of different

systems. Moreover, one needs to have a complete understanding and up-to-date documentation on matching source databases to target databases in such way that the results are meaningful. Once the data are properly understood and matched, one can design appropriate processes of moving and restructuring data without decreasing the quality of data in the process. (Olson 2003, pp. 52–62)

To avoid data quality issues generated by the incorrect interpretation and use of the data, an organization should have a good metadata repository that is continuously maintained. The repository should describe what each data element represents, how the data are encoded, how to interpret special values, what is the source of the data, when was it updated, as well as the last known levels of data quality. The data quality levels are needed, so that users can assess if the data are of enough high quality for their needs. The users – also non-IT professionals – have to be able to access the metadata repository easily. (Olson 2003, pp. 62–63)

To ensure high data quality, organizations have to put strong focus on the design of systems, continuously monitor data collection, and take aggressive actions in order to correct issues generating or propagating inaccurate data (Olson 2003, p. 3). They have to promote understanding the concepts of high-quality data, as well as to educate their employees and make data quality a requirement of all new projects they work on (Olson 2003, p. 15). Furthermore, they have to build the concept of data quality assurance into all of their data management practices (Olson 2003, p. 65). The key to ensuring high data quality is having the knowledge about the data to successfully assess, move and use it (Olson 2003, p. 64).

Olson (2003, p. 34) argues that it is not realistic to reach data accuracy of 100%. He compares data accuracy to air quality remarking that it is not possible to get 100% pure air quality in an area where people live and work. However, many people are able to distinguish between high and low air quality. The same goes for data quality; even though it is not possible to reach 100% data accuracy, it is still possible to differentiate between data of high and low quality. (Olson 2003, p. 34) Moreover, since the concept of high-quality data is dependent on the intended use case, the tolerance level can be determined based on the requirement of the intended use (Olson 2003, p. 25), or one can aim for a degree that makes the data highly useful for all intended use cases (Olson 2003, p. 35).

The discussion so far focuses on general proactive actions in ensuring high data quality. Nevertheless, as Olson (2003) argues, it is not possible to reach perfect data quality. Furthermore, even if all the proactive activities to ensure high data quality were performed, data quality issues will occur, and the data may not be of sufficiently high quality. Many scholars discuss improving data quality through the measurement or assessment of data quality – one has to assess and monitor data quality, or otherwise find data quality issues, in order to know what to improve in particular.

There exists a number of different ways to measure and assess data quality through different data quality dimensions. For example, completeness can be measured by counting the number of missing values (Vaziri et al. 2016) and consistency can be measured by counting the number of outliers (or anomalies) (Nisingizwe et al. 2014). Rettig et al. (2015) remark that for assessing data quality, there exists a number of fundamental data services to be deployed, one of which is anomaly detection. Das and Schneider (2007) state that anomaly detection can also be used to detect data quality errors.

Batini and Scannapieco (2016, p. 174) emphasize that outliers are only abnormal data. Thereby, once the outliers have been detected, it is yet to be decided whether they represent data quality issues or abnormal but legitimate behavior. Accordingly, many scholars discuss anomaly detection as a specific method for detecting possible data quality issues – or anomalies in general – rather than a straightforward means to measure data quality of any data quality dimension. The numerous other techniques to assess data quality or detect data quality issues are not discussed in this study, but the rest of the study focuses on anomaly detection to detect data quality issues.

3 ANOMALY DETECTION

Anomaly detection (often also referred to as outlier detection) has been applied for a long time to detect anomalous data instances, and when applicable, to remove them from the data (Hodge and Austin 2004). In fact, Goldstein and Uchida (2016) argue that the main reason for anomaly detection was to remove them from the training data, as pattern recognition algorithms were rather sensitive to them. Therefore, the development of more robust algorithms caused the interest in anomaly detection to decrease. Nevertheless, around the year 2000, researchers developed more interest in anomalies themselves, as they often associate with interesting events or suspicious data records – possible data quality issues. (Goldstein and Uchida 2016)

In addition to applying anomaly detection for detecting unexpected entries in databases, anomaly detection is an important method in many application domains, such as fraud detection, network intrusion detection and medical condition monitoring (Hodge and Austin 2004). Numerous anomaly detection methods have been designed and introduced for specific application domains, whereas others are proposed as general-purpose methods suitable for different datasets (Chandola et al. 2009). Moreover, Taha and Hadi (2019) annotate that anomaly detection is an important problem that acquires research interest not only within diverse application domains but also within diverse research areas, such as statistics, data mining and machine learning.

This chapter first defines what is meant by anomaly and what different types of anomalies exist. Different types of anomalies are illustrated using examples from numerical data space to provide an intuitive understanding of the concepts. Next, different approaches to anomaly detection are discussed, followed by introducing the problem of detecting anomalies in categorical data.

3.1 Definition of anomaly

There exists no universally accepted definition for an anomaly, but rather there exist several different definitions which mainly follow the same idea. One of the most used definitions of an anomaly was proposed by Hawkins (1980): An outlier is “an observation which deviates so

much from other observations as to arouse suspicions that it was generated by a different mechanism". Later, Chandola et al. (2009) define anomalies as patterns in data that do not comply with the expected behavior. Goldstein and Uchida (2016) add that in addition to anomalies being "different from the norm with respect to their features", they are also infrequent in a dataset in comparison to normal instances. Recently, Taha and Hadi (2019) conclude that, in general, anomalies are a small number of objects that are not consistent with the pattern suggested by a major part of the same dataset's objects.

Taha and Hadi (2019) study anomalies in categorical data, and state that there exists no single widely accepted definition for an anomaly in categorical data. Therefore, anomaly detection methods adopt different definitions, leading to different sets of observations being detected as anomalies (Taha and Hadi 2019). Du et al. (2021) argue that this idea applies to anomaly detection in general; they present five most well-known definitions for an anomaly: widely adopted definition, abnormality degree-based definition, cluster-based definition, density-based definition, and distance-based definition.

An important aspect of anomalies and anomaly detection is the nature of anomaly. Anomalies are widely accepted in literature to be categorized into three categories by their nature: *point anomalies*, *contextual anomalies*, and *collective anomalies*. They can also be categorized on higher level into simple anomalies and complex anomalies – simple anomalies being the aforementioned point anomalies and complex anomalies consisting of contextual and collective anomalies. (Chandola et al. 2009)

Point anomaly

Point anomaly refers to an individual data instance that is considered anomalous compared to the rest of the data. This is the simplest type of anomaly, and majority of research on anomaly detection focuses on identifying point anomalies. As an example, in Figure 2, points O_1 and O_2 are located far from the rest of the data, and thereby can be considered as point anomalies. (Chandola et al. 2009) As a real-life example, consider the context of credit card fraud detection: if one purchase is of much higher cost compared to any other purchase of the same customer, it can be considered a point anomaly (Mehrotra et al. 2017, p. 156).

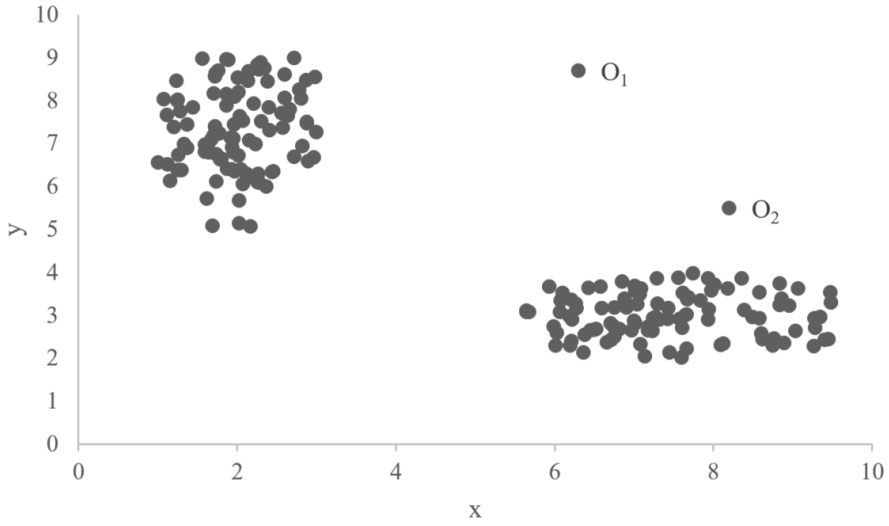

Figure 2. Example of point anomalies (adapted from Chandola et al. 2009)

Contextual anomaly

A data instance that is considered anomalous only in a specific context is termed contextual anomaly (Chandola et al. 2009) – also referred to as *conditional anomaly* (Song et al. 2007). The notion of context is induced by the structure of the dataset and has to be defined along with the problem formulation. Each data instance consists of two different sets of attributes: *contextual attributes* and *behavioral attributes*. Contextual attributes define the context for the data instance. For example, in time series data, time is a contextual attribute, whereas in spatial data, longitude and latitude are the contextual attributes. Behavioral attributes, on the other hand, are the attributes that determine the non-contextual characteristics of a data instance. One example of a behavioral attribute is the amount of rain at any single location in a spatial dataset reporting the amount of rain in the whole world. (Chandola et al. 2009)

Data instances are defined as contextual anomalies using the values of behavioral attributes within a particular context. Accordingly, a data instance may be a contextual anomaly in one context, but a data instance with identical behavioral attributes may be normal in a different context. (Chandola et al. 2009) Most commonly, contextual anomalies are studied in time-series data (Salvador and Chan 2005; Tripathi and Baruah 2020) and spatial data (Kou et al. 2006; Zheng et al. 2017).

As an example of a contextual anomaly, consider a normal temperature during a year to be between 0°C and 30°C. Hence, a temperature of 5°C is rather normal in general. However, when taking the context into account, a temperature t_2 of 5°C in June (Figure 3) would be considered as a contextual anomaly. (Goldstein and Uchida 2016) Note that the same temperature t_1 of 5°C occurring in December and January is not considered as anomaly.

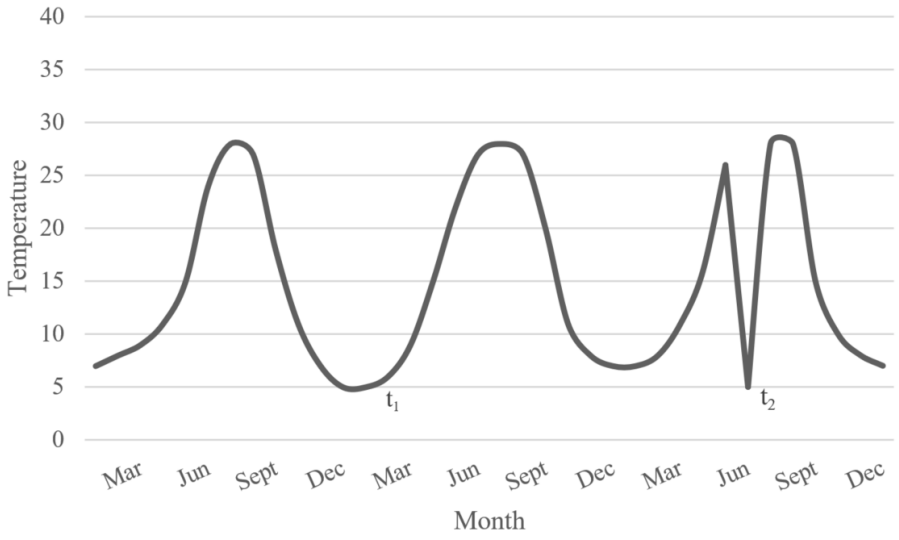

Figure 3. Example of contextual anomaly (adapted from Chandola et al. 2009)

Collective anomaly

A set of data instances significantly differing from the rest of the dataset, is referred to as collective anomaly (Aggarwal 2013, pp. 23–24). In a collective anomaly, the individual data instances might not be anomalous when analyzed individually, but their co-occurrence as a collection is considered anomalous (Chandola et al. 2009). Figure 4 illustrates an electrocardiogram output (Goldberger et al. 2000). The highlighted part indicates an anomaly because the same value is present for abnormally long time. Note that the value itself is within normal limits and is not an anomaly, but its existence as a collection for a rather long time is what makes it anomaly. (Chandola et al. 2009)

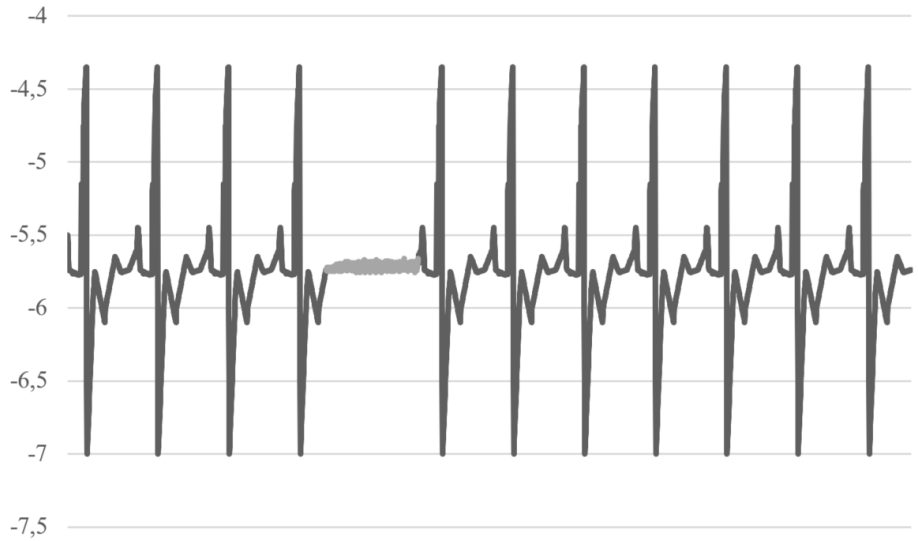

Figure 4. Example of collective anomaly (adapted from Chandola et al. 2009)

As discussed, point anomalies are the simplest type of anomalies with the heaviest research focus, but they are also more general in the sense that they can occur in any dataset. Whereas contextual anomalies can occur only in datasets with contextual attributes, and collective anomalies can occur in such datasets in which data instances are associated with each other. An anomaly can be of one type or two types; incorporating a context in analyzing a point anomaly or a collective anomaly may reveal a contextual anomaly. (Chandola et al. 2009)

3.2 Anomaly detection input data, setups and algorithm output

Nature of input data

An important aspect of anomaly detection is the nature of input data. Data can be of different types, such as continuous, categorical or binary. A dataset may comprise of only one attribute (univariate) or several attributes (multivariate). (Chandola et al. 2009) A multivariate dataset may consist of attributes that are all of the same type, or the attributes may be of different types (Chandola et al. 2009), in which case the data are called mixed data (Taha and Hadi 2019). The nature of the input data determines which anomaly detection methods are applicable for the dataset. In general, methods designed for one data type are not directly applicable for another data type. (Chandola et al. 2009)

Anomaly detection setups

Anomaly detection can be categorized into three different setups: supervised, semi-supervised and unsupervised. The setup of *supervised anomaly detection* requires the anomaly detection model to be trained before use, and the data used for training is required to have labels classifying each data object to either be an anomaly or a normal object. (Wu and Wang 2013) New unseen data can then be compared against the learned anomalous and normal classes to determine which class they belong to – whether they are anomaly or normal (Chandola et al. 2009). Goldstein and Uchida (2016) argue that supervised anomaly detection is rarely relevant for real-life applications, as anomalies are typically not known, and thereby the data cannot be – or is too expensive to be – labeled with anomalies and normal instances.

Semi-supervised anomaly detection is similar to supervised anomaly detection in the sense that it requires the model to be trained and the data to be labelled. However, typically, in this approach, the training data are assumed to be free from anomalies, only consisting of normal data objects. The idea is that anomalies are detected based on the deviation from the model of normal objects. (Goldstein and Uchida 2016) There exist also semi-supervised anomaly detection methods that assume the training data to consist of only anomalous objects (Dasgupta and Nino 2000; Dasgupta and Majumdar 2002), but they are less common in practice, mainly because acquiring a training dataset that includes every possible anomalous behavior is difficult (Chandola et al. 2009).

Unsupervised anomaly detection is the most flexible and most widely applicable approach in the context of anomaly detection, as it does not require data to be labeled. Unsupervised anomaly detection algorithms define anomalies merely based on intrinsic properties of the dataset (Goldstein and Uchida 2016), assuming that majority of the objects in the dataset are normal but there exist anomalies as well (Wu and Wang 2013). If the assumption, that normal data objects are significantly more frequent compared to anomalous objects, does not hold, the methods of unsupervised anomaly detection approach may lead to high false alarm rates (Chandola et al. 2009).

To implement a supervised anomaly detection method, the first task is to label the training data (Wu and Wang 2013). However, as Wu and Wang (2013) argue, classifying the data to anomalous and normal objects to achieve a good training set is labor-intensive and time-consuming – especially when it comes to large high-dimensional datasets with low anomalous data rates. Chandola et al. (2009) also emphasize that acquiring accurate labeled data that is representative of all kinds of behavior is exorbitantly expensive. Furthermore, anomalous behavior is often dynamic in nature – new kinds of anomalies, not present in training data, may emerge.

Semi-supervised anomaly detection approach is more widely applicable than the supervised one, because it does not require labels for the anomaly class. This applies especially to some specific domains, in which anomalies are difficult to model, such as spacecraft fault detection (Fujimaki et al. 2005), in which an anomaly would represent an accident. In comparison to supervised and semi-supervised anomaly detection methods, unsupervised methods are more widely applicable in real-life applications. (Chandola et al. 2009) This study focuses on unsupervised anomaly detection.

Anomaly detection algorithm output

Another aspect of anomaly detection is the way of reporting anomalies – the anomaly detection algorithm output. Typically, anomaly detection algorithms produce an output of either of the following types: labels or scores. (Chandola et al. 2009) Algorithms of which the output are labels, assign for each data object a label that indicates whether the object is an anomaly or not. In contrast, algorithms of which the output are scores, assign for each data object an anomaly score (or confidence value) which indicates the degree to which the data object is considered an anomaly. (Goldstein and Uchida 2016)

Goldstein and Uchida (2016) state that scores can be considered as a more informative output, and Chandola et al. (2009) point out that in case the output are scores, an analyst can choose to take into further analysis only a desired number of top anomalies or use a threshold in selecting the most relevant anomalies. Supervised anomaly detection algorithms often use labels as output due to the availability of classification-based algorithms. In contrast, semi-supervised

and unsupervised anomaly detection algorithms often use scores as output. The reason for that is mainly practical, since many applications rank anomalies and only report the top anomalies to the user. (Goldstein and Uchida 2016)

3.3 Anomaly detection in categorical data

Suri et al. (2012) state that although there exist numerous methods for anomaly detection in numerical data, the problem of detecting anomalies in categorical data is still evolving. They recognize the primary challenge to be the difficulty of defining an appropriate similarity measure (also referred to as distance measure) for the categorical values. Wu and Wang (2013) agree and argue that detecting anomalies in datasets consisting mostly of categorical attributes is a challenging problem, as there exist no inherent distance measure between the objects.

Aggarwal (2013, p. 200) remarks that categorical data can be transformed to binary data by creating a separate binary attribute for each distinct value of a categorical attribute (often referred to as one-hot encoding). Thereby, existing algorithms designed for numerical data can be applied to the transformed data. Nevertheless, he annotates, that in practical applications, such an approach is rather expensive for datasets with categorical attributes for which the number of possible values is large. (Aggarwal 2013, p. 200)

Nian et al. (2016) also recognize the issue of most existing anomaly detection methods focusing only on numerical data. They state that a typical preprocessing technique applied to categorical data is to expand each categorical feature into a set of binary indicators – as presented by also by Aggarwal (2013, p. 200). After which, Euclidean distance can be used as a similarity measure. However, in addition to the issue of expensiveness introduced by Aggarwal (2013, p. 200), Nian et al. (2016) state that this technique may fail in capturing relevant information of a dataset, such as nominal value frequency distribution. Moreover, they state that it can potentially create distortion. (Nian et al. 2016)

According to Nian et al. (2016) a more suitable treatment for categorical data is to select a similarity measure that captures the relationships between categorical variables. Similarity measures for categorical data have been studied for a long time and there exist hundreds of

them (Boriah et al. 2008). However, Nian et al. (2016) recognize that choosing a suitable similarity measure might depend on the application context. Jian et al. (2019) agree and argue that there exists no similarity measure that would be universally effective for all different datasets.

Taha and Hadi (2019) argue that anomaly detection in categorical data has received less attention and still lacks attention compared to that in quantitative data, because of the challenging nature of the problem of detecting anomalies in categorical data. However, fortunately, the research interest in detecting anomalies in categorical data has been increasing. (Taha and Hadi 2019) The next chapter discusses some relevant existing methods for the problem.

In addition to the inherent difficulty of the problem of detecting anomalies in categorical data, the problem faces another challenge – computational complexity. Many real-life applications use datasets with high number of observations and high number of categorical variables with numerous categories in each. Therefore, time complexity is a considerable issue that has to be taken into account in applying anomaly detection methods to categorical data. Another important aspect of anomaly detection methods for categorical data are the number of input parameters required for the methods. It is an important issue, since the choice of parameter values affects the performance of anomaly detection methods – some methods are more sensitive than others. (Taha and Hadi 2019)

4 PREVIOUS STUDIES OF ANOMALY DETECTION IN CATEGORICAL DATA

Literature review was conducted to get an encompassing view of existing anomaly detection methods for categorical data. It was not necessary to study each and every existing method, but rather, relevant and well-known methods. Literature search for anomaly detection in categorical data in the context of detecting data quality issues did not yield many relevant results. Therefore, the literature review is conducted about detecting anomalies in categorical data in general.

As the focus of this study is on unsupervised anomaly detection, literature was first searched including search word *unsupervised* in the search query, but since it did not yield many results, it was noted that many scholars do not specify whether the anomaly detection method is unsupervised, semi-supervised or supervised. Therefore, the search was not limited to unsupervised anomaly detection, but the search results were assessed and only the ones considering unsupervised anomaly detection were included in the literature review.

Literature search was performed in two different databases accessible with university's account: Scopus and Web of Science. Following search query was used to search for literature based on title, abstract and keywords: *"(anomaly OR outlier) AND detect* AND categorical"*. The search was limited to articles and conference papers published in English, and it yielded 308 results in Scopus and 248 results in Web of Science. The initial analysis of search results proved that sorted by relevance, approximately the last third of papers were already quite irrelevant. Thus, sorted by relevance, first 200 results from both databases were taken to further analysis.

The titles and abstracts of the papers were read through, and irrelevant ones were excluded. The duplicate results were excluded at this point as well. Based on title and abstract, there were 90 relevant results, of which full texts were assessed and most relevant ones, including 33 papers, were included in the literature review. The references of relevant papers were also scanned to find additional references. The scanning led to two additional references found, which were then included in the literature review. Finally, the literature review consists of 35 references.

The literature review revealed that there exist many different unsupervised anomaly detection methods for categorical data, and they are often categorized based on the technique on how they determine the anomalous data records. The categorization is, however, often not similar between different papers. In this literature review also, the anomaly detection methods are categorized (Table 1), and the categorization represents just one way of categorization, while it could be done also for example on higher level.

Table 1. Literature review sources

Category	Method	Source
Density	HOT	Wei et al. (2003)
	k-LOF	Yu et al. (2006)
	WATCH	Li et al. (2020)
Marginal Frequency	AVF	Koufakou et al. (2007)
	WDOD	Zhao et al. (2014)
Itemset Frequency	LOADED	Ghoting et al. (2004), Otey et al. (2006)
	FPOF	He et al. (2005a)
	ODMAD	Koufakou & Georgiopoulos (2010)
	FNDI-OD, NBNDI-OD, FNADI-OD	Koufakou et al. (2011)
	COD	Tang et al. (2015)
Bayesian Network / Conditional Frequency	CA	Das & Schneider (2007)
	APD	Das et al. (2008)
	<no name>	Rashidi et al. (2011)
Information-Theory	LSA	He et al. (2005b)
	GA	He et al. (2006)
	ITB-SS, ITB-SP	Wu & Wang (2013)
Compression	KRIMP	Smets & Vreeken (2011)
	CompreX	Akoglu et al. (2012)
Clustering	ROAD	Suri et al. (2012)
	<no name>	Ahmed & Mahmood (2015)
	Rough-ROAD	Suri et al. (2016)
Coupling	CBRW	Pang et al. (2016a)
	POP	Pang et al. (2017)
	COSH	Jian et al. (2019)
	SCAN	Xu et al. (2019)
	SDRW	Pang et al. (2021)
Other	SmartSifter	Yamanishi et al. (2004)
	CNB	Li et al. (2007)
	UA	Pai et al. (2014)
	<no name>	Rettig et al. (2015)
	<no name>	Rossi et al. (2016)
	SRA	Nian et al. (2016)
	ZERO++	Pang et al. (2016b)
	ELMAD	Janakiraman & Nielsen (2016)

Density-based methods

Wei et al. (2003) state that most of the outlier detection methods are designed for numerical data, and therefore they do not work well with real-life applications containing categorical data. They study the detection of local outliers in high-dimensional data and propose Hypergraph-based Outlier Test (HOT). Instead of using distance metrics like many (back then) existing outlier detection methods, HOT uses connectivity property, which makes the method more robust for handling data with missing values. (Wei et al. 2003)

Despite the HOT algorithm proposed by Wei et al. (2003), Yu et al. (2006) claim that existing local density-based outlier detection methods only focus on numerical data. Consequently, they propose a mutual-reinforcement-based local outlier detection method, namely k-Local Anomalies Factor (k-LOF), which can be applied to categorical and quantitative data. The method adds a categorical data handling capability to Local Anomalies Factor (LOF) – outlier detection method introduced by Breunig et al. (2000). The k-LOF method identifies a data instance as a local outlier if its relationship with its neighbors is weaker than the relationships among its neighbors' neighborhood. (Yu et al. 2006)

Recently, Li et al. (2020) proposed a weighted outlier detection method, WATCH, that aims to detect local outliers residing in a subset of correlated features in high-dimensional categorical datasets. The method consists of two distinctive phases: feature grouping and outlier detection. Feature grouping is performed by applying mutual information and entropy to find correlations among the features. The actual outlier detection is then performed by calculating outlier score for each object with respect to each feature group. (Li et al. 2020)

Marginal frequency-based methods

Koufakou et al. (2007) argue that many existing anomaly detection methods require quadratic time complexity and multiple dataset scans. To address the issue, they propose a frequency-based anomaly detection method Attribute Value Frequency (AVF). The method performs only one scan and scales linearly with the number of data points and attributes. AVF is a simple algorithm that computes an AVF score for each data record based on the frequencies of each of

its attribute values, and flags as outliers the data records with lowest AVF scores. (Koufakou et al. 2007)

Zhao et al. (2014) argue that most existing outlier detection methods are not suitable for practical applications, as they cannot handle large high-dimensional datasets, and they often require several user-defined parameters, which are difficult to estimate in practice. Consequently, they state that an effective outlier detection algorithm for categorical data is yet to be proposed. They attempt to take a step towards that direction and propose a Weighted Density-based Outlier Detection (WDOD) algorithm for categorical data. The algorithm estimates the density of each variable, based on which it then computes the weighted density for the entire data using complement entropy, to capture both the uncertainty and the fuzziness in the density of each variable. (Zhao et al. 2014)

Itemset frequency-based methods

Ghoting et al. (2004) claim that an outlier detection method must be "sensitive to response time constraints" set by the domain to which it is applied. Accordingly, they propose Link-based Outlier and Anomaly Detection in Evolving Data sets (LOADED), a tunable outlier detection algorithm for evolving datasets with categorical and continuous features. The algorithm can be tuned to trade off computation for accuracy based on domain needs. The categorical part of LOADED is based on frequent itemset mining. (Ghoting et al. 2004) Later, Otey et al. (2006) build on their earlier work with the LOADED algorithm (Ghoting et al. 2004) by enhancing it to better handle continuous variables.

He et al. (2005a) argue that most outlier detection methods focus only on identifying outliers, even though in real applications it is important to know why a data object is identified as outlier. They aim to provide a simple solution to the issue; they study transaction databases and state that frequent patterns found by an association rule algorithm reflect the "common patterns" in the dataset, and thus, it is intuitive to define as outliers those data objects that contain infrequent patterns. Consequently, they propose FP-Outlier (or FPOF as denoted by many scholars) – an algorithm that detects outliers by discovering frequent patterns. The algorithm first uses an existing association rule mining algorithm Apriori, introduced by Agrawal and Srikant (1994),

to discover the frequent patterns. Then it computes Frequent Pattern Outlier Factor (FPOF) for each data object to define their degree of outlierness. (He et al. 2005a)

Koufakou and Georgiopoulos (2010) extend their previous work on AVF (Koufakou et al. 2007) and present Outlier Detection for Mixed Attribute Datasets (ODMAD). The method is extended to cover not only categorical data, but also continuous data, as well as to define outliers not only based on irregular values, but also based on irregular sets of values – covering the scenario in which all the values of a data record are frequent, but their co-occurrence is infrequent. Therefore, the categorical part of ODMAD assigns anomaly scores based on the idea of frequent itemset mining, similarly to the method introduced by Ghoting et al. (2004) and Otey et al. (2006) but considering the frequency of infrequent values and assigning higher anomaly scores for more infrequent itemsets. (Koufakou and Georgiopoulos 2010)

Koufakou et al. (2011) argue that while outlier detection methods based on frequent itemset mining – like the ones proposed by Ghoting et al. (2004), He et al. (2005a) as well as Koufakou and Georgiopoulos (2010) – have proved to detect outliers well, they "face significant challenges" when applied to large high-dimensional data. Therefore, Koufakou et al. (2011) study outlier detection using Non-Derivable Itemsets (NDIs) – a condensed representation of Frequent Itemsets (FIs) introduced by Calders and Goethals (2007). Koufakou et al. (2011) propose three different methods based on Frequent NDIs (FNDI-OD), based on the Negative Border of frequent NDIs (NBNDI-OD) and based on Frequent Non-Almost Derivable Itemsets (FNADI-OD). (Koufakou et al. 2011)

Despite the attempt of He et al. (2005a) to bring interpretability to outlier detection, Tang et al. (2015) argue that it still remains a critical issue. To address the issue, they propose Contextual Outlier Detection (COD) algorithm, which aims to provide interpretation and context to identified anomalies. The algorithm is based on finding closure groups which are then used to assemble contextual outliers. (Tang et al. 2015)

Bayesian network / conditional frequency-based methods

Das and Schneider (2007) introduce the approach to detect anomalies by comparing against attribute subsets' distributions. They propose an anomaly detection method, Conditional Algorithm (CA) as denoted by many scholars, that employs conditional anomaly test to detect unusual combinations of attribute values. CA constructs a conditional Alternating Decision (AD) Tree (introduced by Moore and Lee (1998)), computes a mutual information matrix, and constructs a cache for the denominator counts, after which it measures the rareness of data records to define anomalies. (Das and Schneider 2007)

Das et al. (2008) address the problem of detecting anomalous patterns in multidimensional large datasets and present Anomaly Pattern Detection (APD) method. The method first uses CA algorithm proposed by Das and Schneider (2007) to give an anomaly score for all individual records. After detecting individual anomalies, the method uses a rule-based method "What's Strange About Recent Events" (WSARE) (introduced by Wong et al. (2002)) to detect anomalous clusters of counts. (Das et al. 2008)

Rashidi et al. (2011) propose an anomaly detection method that considers the number of occurrences of different attribute value combinations. The method utilizes a Bayesian network and stores the number of occurrences of different attribute value combinations using AD Tree. The proposed method then computes the anomaly scores similarly to the conditional anomaly test algorithm proposed by Das and Schneider (2007). (Rashidi et al. 2011)

Information-theoretic methods

He et al. (2005b) state that most existing methods are not based on a solid theoretical foundation or alternatively they assume underlying distributions that often do not exist in practical applications. To address this issue, they formulate outlier detection as an optimization problem: to find a small subset of data, of which removal minimizes the entropy of the resultant dataset. To solve the problem, they propose a Local-Search heuristic-based Algorithm (LSA), which iteratively improves the value of objective function until an optimal subset to be defined as outliers is found. (He et al. 2005b)

Later, He et al. (2006) acknowledge that their LSA algorithm (He et al. 2005b) is still time-consuming on very large datasets, like most of the iterative algorithms. They present a "very fast" Greedy Algorithm greedyAlg1 (or GA, as denoted by many scholars) for detecting outliers in very large datasets using the same optimization model as in LSA. The algorithm defines as outliers the data objects of which removal results in maximal decrease in entropy, performing only as many scans as is the number of desired outliers. (He et al. 2006)

Like He et al. (2005b, 2006), also Wu and Wang (2013) formulate outlier detection as an optimization problem and propose two 1-parameter algorithms for large-scale categorical datasets: Information-Theory-Based Step-by-Step (ITB-SS) and Information-Theory-Based Single-Pass (ITB-SP). The algorithms rely on a concept of weighted holoentropy integrating both entropy and total correlation. (Wu and Wang 2013)

Compression-based methods

Smets and Vreeken (2011) agree with He et al. (2005a) on that explanation for why a data object is identified as an outlier is very important. Consequently, they propose an approach for identifying outliers in transaction data – with characterization of why they were identified as outliers. Their method employs Minimum Description Length (MDL) principle and KRIMP itemset-based compressor to build a code table. (Smets and Vreeken 2011)

Akoglu et al. (2012) improve over Smets and Vreeken's (2011) approach to using a compression technique in anomaly detection. They introduce CompreX, a pattern-based compression method for anomaly detection. Instead of building only one code table like the method proposed by Smets and Vreeken (2011), CompreX builds multiple code tables to exploit correlations between groups of attributes. (Akoglu et al. 2012)

Clustering-based methods

Suri et al. (2012) state that the problem of outlier detection in categorical data remains a challenge, and they address it by proposing Ranking-based Outlier Analysis and Detection

(ROAD) algorithm. ROAD defines two different scenarios as outliers: the categorical values of the data object are relatively infrequent (frequency-based outlier) or the combination of the categorical values of the data object is relatively infrequent, even though each of the individual values are frequent (clustering-based outliers). ROAD algorithm consists of two phases; first, object density computation is performed and data clustering using k-modes (Huang 1997) is carried out, after which the most likely outliers are defined using both frequency-based ranking and clustering-based ranking. (Suri et al. 2012)

Ahmed and Mahmood (2015) argue that existing clustering-based anomaly detection methods have high false alarm rates and only consider the behavior of individual data instance for assessing anomalousness. Their study focuses on detecting denial of service attacks as collective anomalies in network traffic data. To address the problem, they introduce an extension to information theoretic co-clustering algorithm – the ability to handle categorical attributes. They recognize that there exist several different similarity measures for categorical data, but for simplicity, they use dissimilarity of one when data instances do not match, and zero otherwise. (Ahmed and Mahmood 2015)

Suri et al. (2016) argue that when it comes to clustering-based methods for outlier detection, the uncertainty regarding the cluster memberships of outliers must be treated properly. Therefore, they extend the ROAD algorithm (Suri et al. 2012) by proposing Rough-ROAD algorithm. Rough-ROAD uses rough set theory to modify the k-modes algorithm of ROAD (Suri et al. 2012) to rough k-modes algorithm to capture the uncertainty of cluster memberships. (Suri et al. 2016)

Coupling-based methods

Pang et al. (2016a) introduce Coupled Biased Random Walks (CBRW) method to address the problem of identifying outliers in categorical data with various frequency distributions and multiple noisy features. CBRW uses biased random walks to model feature value level couplings to estimate feature values' outlier scores. It considers both intra-feature coupling, meaning distribution within a category, and inter-feature coupling, meaning interactions between categories. (Pang et al. 2016a)

Pang et al. (2017) argue that existing outlier detection methods for categorical data face challenges when applied to high-dimensional datasets with many noisy features. To address the issue, they study whether outlying behaviors can be well separated from non-outlying behaviors by modeling only selective value couplings. Consequently, they propose Partial Outlierness Propagation-based method POP, for learning selective features – interactions between the full value set and a set of outlying values. POP captures the selective value couplings by modeling a partial outlierness propagation process to define the outlierness. (Pang et al. 2017)

To address the issue of capturing complex hierarchical value coupling relationships in categorical data, Jian et al. (2019) introduce an outlier detection model – Coupled Outlier Scoring of High-dimensional data (COSH). The model uses k-means clustering to learn multi-granularity value clusters, based on which it computes the most outlying aspect of values. (Jian et al. 2019)

Xu et al. (2019) argue that even though coupling-based outlier detection methods have proved to be reliable in detecting outliers in data that is not Independent and Identically Distributed (IID), existing methods often consider only pairwise primary value couplings, while failing to expose real relations hiding in complex couplings of non-IID data. Thereby, the methods may result in suboptimal and unstable performance. To address the issue, Xu et al. (2019) propose Skip-gram architecture on a biased value Coupling-based vAlue Network (SCAN) which is able to learn and utilize also high-order complex value couplings. (Xu et al. 2019)

Recently, Pang et al. (2021) build on their preliminary version of CBRW (Pang et al. 2016a) and introduce an outlier detection framework Coupled Unsupervised OuTlier detection (CUOT). The framework is separated into two different methods: CBRW and Subgraph Densities-augmented Random Walks (SDRW). SDRW is a parameter-free enhancement to CBRW, and models outlierness propagation on undirected value graphs instead of directed value graphs like CBRW. (Pang et al. 2021)

Other methods

Yamanishi et al. (2004) present SmartSifter – a model for online unsupervised outlier detection, experimentally evaluated with a network intrusion detection data consisting of both continuous and categorical features. The model is a probabilistic model, based on statistical learning theory. The categorical part of the model learns the histogram density through Sequentially Discounting Laplace Estimation (SDLE) algorithm. SmartSifter gradually "discounts" the effect of past examples – putting more weight to the recent examples. An outlier score for a datum is then determined based on statistical distance measuring how much the model changed after adding a new datum, the bigger the change, the higher the outlier score. (Yamanishi et al. 2004)

According to Li et al. (2007) when outlier detection is performed by applying a distance-based method to categorical data, a typical way to handle categorical data is simple matching; the distance between two identical categorical values is 0, whereas the distance between two non-identical values is 1. They argue that it is not an appropriate approach for high-dimensional categorical data because of the "curse of dimensionality". Consequently, they suggest a Common-Neighbor-Based (CNB) distance function to measure the proximity between two data points. CNB first generates the neighbor set for each data object, followed by computing the similarity-based distances. (Li et al. 2007)

Pai et al. (2014) claim that the frequent itemset mining-based approaches to outlier detection prune most of the data and utilize only limited information, leading to lower accuracy. To address this issue, they propose the concept of relative patterns discovery on association analysis. They introduce a hash-index-based intersecting approach to efficiently examine the relative patterns, based on which they then propose an Unsupervised Approach (UA) to assess which observations are anomalous – without limiting the information. (Pai et al. 2014)

Rettig et al. (2015) approach the problem of anomaly detection from the data quality point of view, and specifically study the detection of anomalies in data streams. They propose the detection of anomalies using two different measures: relative entropy and Pearson correlation. They state that their solution works for both numerical as well as categorical data and can be used for both data streams and for data at rest. (Rettig et al. 2015)

The study of Rossi et al. (2016) focuses on increasing the level of intelligence of an electricity network through anomaly detection. They present an unsupervised anomaly detection approach for detecting contextual and collective anomalies, and they apply it to data streams from a large energy distributor. Their approach uses frequent itemset mining and categorical clustering based on entropy minimization. After which the anomalous behavior is then detected by clustering silhouette thresholding. (Rossi et al. 2016)

Nian et al. (2016) propose a Spectral Ranking method for Anomaly detection (SRA) for auto insurance fraud detection. They consider two different methods for assessing anomalies – rare class ranking and anomaly ranking. The former assesses anomalies with respect to a one single majority pattern, while the latter assesses anomalies with respect to multiple different patterns in the data. Their method focuses on detecting anomaly in attribute dependence using similarity kernels, in which it is assumed that the similarity constructed to capture the input attributes' dependence relation is given. To deal with categorical variables in the data, Nian et al. (2016) use the simple overlapping similarity measure and its derived kernels. They emphasize that an appropriate selection of similarity measure remains crucial for a fraud detection problem. (Nian et al. 2016)

Pang et al. (2016b) approach the problem of anomaly detection from a different angle. They introduce ZERO++, a method that detects anomalies by employing the number of zero appearances in subspaces. The method is based on the assumption that anomalies are not likely to occur in small subsamples, and also, they are less likely, compared to normal instances, to occur in subsamples of any size. They argue that the existing frequency-based anomaly detection algorithms conduct a subspace pattern searching with at least quadratic time complexity compared to data size and dimensionality, while the time complexity of ZERO++ is linear, as it includes no searching. (Pang et al. 2016b)

Janakiraman and Nielsen (2016) examine Extreme Learning Machines (ELM) based anomaly detection to achieve fast training and good generalization performance with aviation safety data. An ELM is a feed-forward model with one hidden layer. The input layer parameters for the model are assigned randomly and fixed during training. Janakiraman and Nielsen (2016)

propose three anomaly detection algorithms, sparse autoencoder model (L_1-ELMAD), non-sparse autoencoder model (L_2-ELMAD) and embedding model (Em-ELMAD).

5 ALGORITHMS SELECTED FOR THE EMPIRICAL STUDY

This chapter first introduces the criteria based on which two anomaly detection algorithms are selected for the empirical study. Second, it discusses the selections, including limiting factors. Finally, the selected algorithms are described in detail.

The algorithms are selected for the empirical study based on following criteria:

1. The algorithm seems potential for the use case of detecting data quality issues
2. The algorithm is mentioned in several studies or otherwise considered state-of-the-art
3. The algorithm is publicly available for use and implemented in Python
4. The algorithm does not require an extensive number of input parameters

Different categories of algorithms were analyzed for the use case of detecting data quality issues, and irrelevant ones were excluded. To detect data quality issues in the context of this study, it is important that the algorithm considers values across attributes. Therefore, algorithms employing only marginal frequency are excluded, as they may not find all relevant anomalies in this study, since erroneous data records may often comprise of "normal" values that only together represent a data quality issue.

Previous literature on anomaly detection in categorical data consists almost explicitly of publications that introduce a new algorithm, method, or framework for the problem. The scholars often compare their proposed method to existing state-of-the-art methods, and it is evident that the proposed method is nearly without an exception considered superior to the existing state-of-the-art methods. It might be an indication of research bias, and it makes the comparison of the performance of different methods difficult. However, methods that are mentioned to be state-of-the-art methods by many scholars, can rather objectively be considered to be such.

To the best of the author's knowledge, at the time of conducting the study, there does not exist any easily available and commonly used Python libraries that would include unsupervised anomaly detection algorithms specifically for categorical data. Therefore, the algorithms are

searched from other public resources, such as GitHub. Still, the availability of algorithms proved to be the major limiting factor in algorithm selection. However, it was not necessary for this study to select the most potential algorithms, but rather to select some potential algorithms for testing and evaluation.

CBRW was selected for the empirical study because it was mentioned in several studies. Due to the algorithm availability limitations, SCAN was selected for the study to represent a more recently introduced algorithm. CBRW and SCAN are both coupling-based methods. Regarding the number of input parameters, CBRW requires one input parameter, while SCAN requires two.

CBRW

CBRW is an anomaly detection algorithm that considers both distribution within a category and interactions between categories. To describe how the CBRW algorithm works, following notations are used. Let $X = \{x_1, x_2, \ldots, x_N\}$ be a set of data records with size N. It is described by a set of D categorical features $F = \{f_1, f_2, \ldots, f_D\}$. Each feature consists of a finite set of possible feature values, domain $dom(f) = \{v_1, v_2, \ldots, v_N\}$. To refer to the value of a data object x in feature f, notation v_f^x is used. Each feature f is affiliated with a categorical distribution, in which f takes one of the possible feature values $v \in dom(f)$ with frequency of $p(v)$ (Pang et al. 2016a):

$$p(v) = \frac{\left|\{x \in X | v_f^x = v\}\right|}{N} \quad (1)$$

The domains are distinct between features: $dom(f_i) \cap dom(f_j) = \emptyset, \forall i \neq j$, and the full set of feature values V is the union of all feature domains: $V = \cup_{f \in F} dom(f)$ (Pang et al. 2016a).

As an input, the algorithm takes the set of data objects X to be analyzed and a damping factor α (referred to as α_1 in this study). First, CBRW algorithm builds a directed and weighted graph $G = < V, E >$. In graph G, each feature value is represented by a node $v \in V$, while the entries

of the out-degree adjacent matrix *A* of graph *G* represent the weights that are assigned to the edges. (Pang et al. 2016a)

In the graph *G*, for each node *v*, a node property is defined based on the frequency of the feature value and the frequency of the mode within the feature. Accordingly, CBRW computes the intra-feature outlierness of a feature value $v \in dom(f)$ using the deviation of the value frequency from the mode frequency, *dev*(*v*), and the outlierness of the feature mode *m*, *base*(*m*) (Pang et al. 2016a):

$$\gamma(v) = \frac{1}{2}\,(dev(v) + base(m)) \tag{2}$$

where $dev(v) = \frac{p(m)-p(v)}{p(m)}$ and $base(m) = 1 - p(m)$.

The intra-feature outlierness γ gets values from the range (0,1). However, any features having $p(m) = 1$, are ignored since they contain no information for anomaly detection. The intra-feature outlierness γ is intended for enabling semantical comparison between values from different frequency distributions. As a basis for this measure, the outlierness of the feature mode is used – the larger the deviation of the frequency of a feature value from the mode frequency, the higher the outlierness of that feature value. (Pang et al. 2016a)

Inter-feature couplings are examined to find out if feature values are coupled with the outlying behaviours of other features. The out-degree adjacent matrix *A* builds on conditional probabilities between every two nodes, and the entry *A*(*u, v)* of matrix *A* is a weight assigned to the edge from node *u* to node *v* – i.e. the strength of coupling between nodes *u* and *v*. It is determined as follows (Pang et al. 2016a):

$$A(u,v) = p(u|v) = \frac{p(u,v)}{p(v)} \tag{3}$$

where $p(u, v)$ denotes the co-occurrence frequency of values *u* and *v*, $\forall u, v \in V$.

Based on intra-feature outlierness $\gamma(v)$ and the strength of inter-feature coupling $A(u, v)$, CBRW then builds a biased random walks transition matrix W_b, of which entry $W_b(u, v)$ is defined as follows (Pang et al. 2016a):

$$W_b(u,v) = \frac{\gamma(v)A(u,v)}{\sum_{v \in V} \gamma(v)A(u,v)} \tag{4}$$

where $W_b(u, v)$ denotes the transition from node u to node v. It has a probability proportional to $\gamma(v)A(u, v)$, and thereby, each random move is biased by the value of $\gamma(v)$. (Pang et al. 2016a)

After generating the matrix W_b, CBRW initializes the column vector π_0 as a uniform distribution. Then, it calculates π_{t+1}, in which $\pi_t \in \mathbb{R}^{|V|}$ denotes the probability distribution of the biased random walk at time step t (Pang et al. 2016a):

$$\pi_{t+1} = (1 - \alpha_1)\frac{1}{|V|}1 + \alpha_1 W_b^T \pi_t \tag{5}$$

where α_1 is a damping factor guaranteeing convergence.

Finally, in estimating value outlierness, CBRW computes the outlier score of each node (feature value) v based on its stationary probability from converged probability distribution (Pang et al. 2016a):

$$value\_score(v) = \pi^*(v) \tag{6}$$

where $0 < \pi^*(v) < 1$ and $\sum_{v \in V} \pi^*(v) = 1$.

The outlier scores of the feature values can then be utilized to weight the features based on their relevance for outlier detection. The relevance of feature *f* is computed as follows (Pang et al. 2016a):

$$rel(f) = \sum_{v \in dom(f)} value\_score(v) \tag{7}$$

Eventually, outlier score of each object *x* can be obtained by taking a weighted sum of the outlier scores of the feature values of the object, using relevance weighting factor to give more weight to the outlier scores of feature values of relevant features (Pang et al. 2016a):

$$object\_score(x) = \sum_{f \in F} \omega_f * value\_score(v_f^x) \tag{8}$$

where $\omega_f = \frac{rel(f)}{\sum_{f \epsilon F} rel(f)}$ is a feature weighting component.

In summary, as an input, the algorithm takes the set of data objects *X* to be analyzed and a damping factor α_1. It first maps the categorical data into a value-value attribute graph. The topological graph structure is defined by inter-feature value couplings, whereas the node property builds on intra-feature couplings. Thereby, the problem of estimating the value outlier degree is transformed to a graph-based ranking problem – a problem of ranking the nodes. Eventually, the algorithm obtains the value outlier scores by building biased random walks on the graph. CBRW algorithm for estimating value outlierness is described in Algorithm 1. Thereafter, outlier score of each data object is computed based on outlier scores of its values. (Pang et al. 2016a)

Algorithm 1. CBRW (adapted from Pang et al. 2016a)

Input: X - data objects, α_1 - damping factor
Output: π^* - the stationary probability distribution

for i = 1 to D **do**
  Compute $p(v)$ for each $v \in dom(f_i)$
  Find the mode of f_i
  Compute $\gamma(v)$
  for $j = i + 1$ to D **do**
    Compute $p(u, v)$, $\forall u \in dom(f_j)$
  end for
end for
Generate the matrix W_b
Initialize π^* as a uniform distribution
repeat
  $\pi^* \leftarrow (1 - \alpha_1)\frac{1}{|V|}1 + \alpha_1 W_b^T \pi^*$
until Convergence, i.e., $\|\pi_t^* - \pi_{t-1}^*\|_\infty \leq 0{,}001$ or reach the maximum iteration $I_{max} = 100$
return π^*

SCAN

SCAN is a coupling-based algorithm like CBRW, but it also considers relationships that hide in high-order complex value couplings. The notations used in describing how the SCAN algorithm works are next presented. Let $X = \{x_1, x_2, \ldots, x_N\}$ be a set of data records with size N. It is described by a set of D categorical features $F = \{f_1, f_2, \ldots, f_D\}$. To denote the value of a data object x in feature f, notation v_f^x is used, and $dom(f)$ is the value domain of feature f. The value domains are distinct between features: $dom(f_i) \cap dom(f_j) = \emptyset, \forall i \neq j$, and the full set of feature values V is the union of all feature domains: $V = \cup_{f \in F}\, dom(f)$. Let $h : V \rightarrow \mathbb{R}^r$ be the mapping function from values to numerical feature representations that SCAN aims to learn to evaluate data object outlierness. (Xu et al. 2019)

As an input, the algorithm takes the set of data objects X to be analyzed, subset size factor α (referred to as α_2 in this study) and representation dimensionality r. First, SCAN learns primary value couplings – both direct and indirect couplings. Pairwise direct value couplings are learned

by building Ochiai coefficient-based matrix $M_1 \in \mathbb{R}^{|V|\times|V|}$ and conditional probability-based matrix $M_2 \in \mathbb{R}^{|V|\times|V|}$ as follows (Xu et al. 2019):

$$M_1(u,v) = \frac{P(u,v)}{\sqrt{P(u)\times P(v)}}, u, v \in V, \tag{9}$$

$$M_2(u,v) = \frac{P(u,v)}{P(u)}, u, v \in V, \tag{10}$$

where $P(v)$ is the marginal probability of the value v, i.e. $P(v) = \left|\left\{x \in X \left| v_f^x = v\right.\right\}\right|/N$, and $P(u, v)$ is the joint probability of value u and v, i.e. $P(u,v) = \left|\left\{x \in X \left| v_{f_u}^x = u \cap v_{f_v}^x = v\right.\right\}\right|/N$.

Indirect value coupling matrix $M' \in \mathbb{R}^{|V|\times|V|}$ is then computed by the cosine similarity between conditional probability vectors (Xu et al. 2019):

$$M'(u,v) = \frac{M_2(u,\cdot)\cdot M_2(v,\cdot)}{\|M_2(u,\cdot)\|\|M_2(v,\cdot)\|}, u, v \in V, \tag{11}$$

where $M_2(v,\cdot)$ is row vector of value v in matrix M_2 and $\|\cdot\|$ is ℓ_2-Norm.

To learn complex value couplings with different granularities, that can exhibit different semantics and reflect the characteristics of data, SCAN performs value clustering using spectral clustering on matrix M_2 with different cluster numbers k, i.e. $D_k = SC(M_2, k)$, where $D_k \in \mathbb{R}^{|V|\times|V|}$, $D_k(u, v) = 1$ if value u and v are in same cluster and $D_k(u, v) = 0$ if they are not. To acquire affinity matrix and apply discretization to assign labels, SCAN uses the default Radial Basis Function (RBF) kernel. The cluster number k is increased from initial value of two until a cluster with only one member appears. This dynamic setting enables capturing different granularities of value couplings. (Xu et al. 2019)

Matrix $D \in \mathbb{R}^{|V|\times|V|}$ is defined based on clustering results (Xu et al. 2019):

$$D = \frac{1}{k_{max} - 1} \sum_{i=2}^{k_{max}} D_i \,, \tag{12}$$

where k_{max} is the cluster number due to which a cluster with only one member appears.

Bidirectional Selective Value Coupling (BSVC)-based function is used to calculate initial value outlierness. BSVC-based value outlierness scoring function $\phi(C, S_o, S_n, \alpha_2)$ is defined to get value outlierness vector $\eta \in \mathbb{R}^{|V|}$:

$$\eta = \phi(C, S_o, S_n, \alpha_2) = \frac{1}{2\alpha_2|V|}\left(\sum_{v\in S_o} C(v,\cdot) + \left(e - \sum_{v\in S_n} C(v,\cdot)\right)\right), \tag{13}$$

where S_o and S_n are value subsets that contain outlying values and normal values respectively with size $\alpha_2|V|$, $C \in \mathbb{R}^{|V|\times|V|}$ is a value coupling matrix and $e = \sum_{i=1}^{|V|} e_i$ is vector of ones. (Xu et al. 2019)

To get the initial value outlierness vector η_0 through function ϕ, SCAN uses rough value scoring function δ to rank the values and obtain value subsets S_o^δ and S_n^δ, i.e. $\delta(v) = \frac{P(m)-P(v)}{P(m)}$, where m is the mode value of the same feature of v. As input matrix C, symmetric direct value coupling matrix M_1 is used, while α_2 is the subset size factor given as an input parameter to SCAN. Therefore, initial value outlierness vector is obtained as $\eta_0 = \phi\left(M_1, S_o^\delta, S_n^\delta, \alpha_2\right)$. (Xu et al. 2019)

After obtaining clustering statistics matrix and value outlierness vector, in order to direct the focus more on outliers, SCAN builds a non-zero value coupling bias matrix $B \in \mathbb{R}^{|V|\times|V|}$ as follows (Xu et al. 2019):

$$B(u,v) = \left(1 + \frac{\eta_0(u) + \eta_0(v)}{2}\right) \times \big(1 + D(u,v)\big), u, v \in V \tag{14}$$

To further learn complex value couplings, SCAN utilizes network embedding and constructs an undirected weighted value network $G = < V, E >$. Nodes of the network represent feature values and edge weights represent the biased value couplings. The adjacency matrix of the network $A \in \mathbb{R}^{|V|\times|V|}$ is as follows (Xu et al. 2019):

$$A = M_1 \circ M' \circ B, \tag{15}$$

where $\circ$ denotes entrywise product.

Algorithmic framework *node2vec* (Grover and Leskovec 2016) is employed on value network G to represent each value with an r-dimensional vector – value representation matrix $N_v \in \mathbb{R}^{|V|\times r}$ is generated. After obtaining the value representation matrix N_v, BSVC learning is used to evaluate value outlierness. Final value outlierness vector η^* is then used to get object outlier scores. BSVC learning performs BSVC-based value outlierness scoring and ranking-based value selection until it finds a stationary value rank, after which a stationary value outlierness vector is generated, as shown in Figure 5. (Xu et al. 2019)

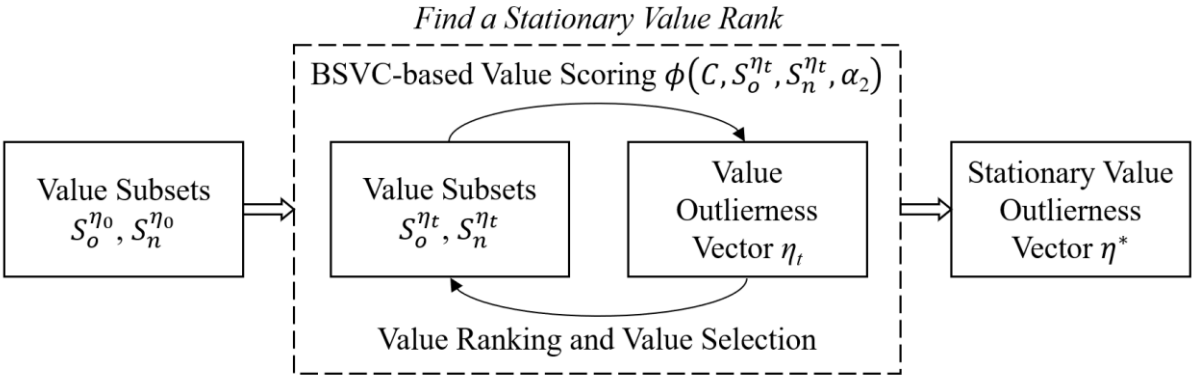

Figure 5. BSVC learning to evaluate value outlierness (adapted from Xu et al. 2019)

The input value subsets $S_o^{\eta_0}$ and $S_n^{\eta_0}$ are the top α_2 and bottom α_2 values when the value rank is sorted by initial value outlierness η_0. Value similarity matrix $M_c \in \mathbb{R}^{|V|\times|V|}$ computed from matrix N_v, i.e. $M_c(u,v) = \frac{N_v(u,\cdot)\cdot N_v(v,\cdot)}{\|N_v(u,\cdot)\|\|N_v(v,\cdot)\|}, u, v \in V$, is set to be the input coupling matrix for function $\eta^* = \phi(M_c, S_o^{\eta_0}, S_n^{\eta_0}, \alpha_2)$. (Xu et al. 2019)

Finally, SCAN calculates data object outlier scores through summation of value outlierness (Xu et al. 2019):

$$\tau(x) = \sum_{f\in F} \eta^*(v_f^x), x \in X \tag{16}$$

In summary, as an input, the algorithm takes the set of data objects *X* to be analyzed, subset size factor α_2 and representation dimensionality *r*. It first learns direct and indirect primary value couplings using Ochiai coefficient and conditional probability. To learn complex value couplings, it introduces value coupling bias and uses an extended Skip-gram architecture designed for network – *node2vec* (Grover and Leskovec 2016). Finally, the algorithm uses BSVC learning to assess value outlierness, followed by further measuring object outlierness. (Xu et al. 2019) The algorithm is described in Algorithm 2.

Algorithm 2. SCAN (adapted from Xu et al. 2019)

Input: X - data objects, α_2 - subset size factor, r – representation dimensionality

Output: R - outlier rank

1: $M_1(u,v) \leftarrow \frac{P(u,v)}{\sqrt{P(u)\times P(v)}}, u,v \in V$

2: $M_2(u,v) \leftarrow \frac{P(u,v)}{P(u)}, u,v \in V$

3: $M'(u,v) \leftarrow \frac{M_2(u,\cdot)\cdot M_2(v,\cdot)}{\|M_2(u,\cdot)\|\|M_2(v,\cdot)\|}, u,v \in V$

4: **repeat**

5: $D_k = SC(M_2, k)$

6: $k = k + 1$

7: **until** Appear a cluster with only one member

8: $D \leftarrow \frac{1}{n}\sum_{i=2}^{n} D_i$

9: $\eta_0 \leftarrow \phi\left(M_1, S_o^{\delta}, S_n^{\delta}, \alpha_2\right)$

10: $B(u,v) \leftarrow \left(1 + \frac{\eta_0(u)+\eta_0(v)}{2}\right) \times \left(1 + D(u,v)\right), u,v \in V$

11: Construct value network G as $A \leftarrow M_1 \circ M' \circ B$

12: Run *node2vec* on G to obtain matrix $N_v \in \mathbb{R}^{|V|\times r}$

13: $M_c(u,v) \leftarrow \frac{N_v(u,\cdot)\cdot N_v(v,\cdot)}{\|N_v(u,\cdot)\|\|N_v(v,\cdot)\|}, u,v \in V$

14: $\eta^* \leftarrow \phi\left(M_c, S_o^{\eta_0}, S_n^{\eta_0}, \alpha_2\right)$

15: $\tau(x) \leftarrow \sum_{f\in F} \eta^*(v_f^x), x \in X$

16: $R \leftarrow$ Sort X w.r.t. τ in descending order

17: **return** R

6 APPLYING ANOMALY DETECTION TO DETECT DATA QUALITY ISSUES IN CATEGORICAL DATA

This chapter constitutes the empirical part of this study. First, it describes the background for the empirical case study, followed by reporting how the case study was implemented in practice. Finally, the results of the case study are presented and discussed.

6.1 Case background

In the case study, selected anomaly detection algorithms are applied to data collected from a case company that operates globally in engineering and service business. Like many organizations today, case company has recognized the importance of data quality – including the importance of proactive activities. The case company has piloted some new proactive practices for ensuring high data quality. To promote, follow and eventually take corrective actions to data quality, they have a vision of how to make it visible for everyone in a way suitable for different viewpoints to the topic. Furthermore, they have defined data quality dimensions to be followed, and they are exploring concrete ways and tools to monitor data quality and detect issues in it using analytical algorithms and models. This study – detecting data quality issues in categorical data through anomaly detection – contributes to that activity.

The aim of this case study is to test two different publicly available anomaly detection algorithms, introduced in chapter 5. The results of the algorithms are compared and analyzed. The data used in the empirical study consists of selected attributes of the company's order book data collected from existing databases specifically for the purpose of this study. The data are assumed to be of rather high quality, but some data quality issues are assumed to exist. Thus, it matches the assumptions of the concept of unsupervised anomaly detection.

The case data are limited to new equipment orders which have been confirmed for manufacturing ("Point of no return" milestone) in year 2021 before December. Moreover, only orders of actual equipment are considered, and all additional work are excluded. The case data consists of three different datasets. Since they are hierarchically on different levels, joining them all together would have resulted in unwanted duplicating of data in most attributes.

Therefore, they are analyzed as separate datasets. The sizes of the datasets are summarized in Table 2.

Table 2. Dataset sizes

Dataset	Number of records	Number of categorical attributes
T1	34 479	11
T2	169 705	6
T3	16 867 344	3

The first dataset, T1, consists of 11 categorical attributes in addition to a unique identifier *SalesOrderNumber*. The dataset has 34 479 data records in total, and missing values in attributes *OrderReason* and *MajorProjectType*. A summary of the dataset, including attribute names, number of distinct values and share of missing values, is presented in Table 3.

Table 3. Summary of dataset T1

Attribute	Distinct values	Missing values
SalesOrderNumber	34 479	0 %
SalesOrganization	67	0 %
Creator	267	0 %
SalesDocumentType	3	0 %
OrderReason	7	99,72 %
SoldToParty	20 632	0 %
SalesOffice	492	0 %
Division	3	0 %
DistributionChannel	2	0 %
MajorProjectType	2	98,77 %
DocumentCurrency	38	0 %
ShippingConditions	9	0 %

The second dataset T2 consists of 6 categorical attributes in addition to two unique identifiers: *SalesOrderNumber-SalesOrderItem* and *NetworkNumber*. There are two unique identifiers because the attributes are combined from different tables to a single dataset. The unique identifiers have one-to-one relationship with each other, but both of them are included in this study because they are needed for the activity following this study – tracing and analyzing whether the detected anomalies are data quality issues or abnormal but legitimate data records.

The dataset T2 has 169 704 data records in total, and a high share of missing values in attribute *ReasonForRejection*. In addition, there is a minor share of values missing from attribute *RelevantForBilling*. A summary of the dataset is presented in Table 4.

Table 4. Summary of dataset T2

Attribute	Distinct values	Missing values
SalesOrderNumber-SalesOrderItem	169 704	0 %
NetworkNumber	169 704	0 %
Company	63	0 %
Plant	66	0 %
ItemCategory	11	0 %
Currency	38	0 %
RelevantForBilling	2	0,0024 %
ReasonForRejection	6	99,97 %

The third dataset T3 consists of three categorical attributes in addition to a unique identifier *RoutingNumber-ActivityCounter*. The dataset has 16 867 344 data records in total, and missing values in attribute *ActivityType*. A summary of the dataset is presented in Table 5.

Table 5. Summary of dataset T3

Attribute	Distinct values	Missing values
RoutingNumber-ActivityCounter	16 867 344	0 %
ActivityNumber	2 092	0 %
ActivityType	36	93,79 %
Plant	69	0 %

6.2 Implementation

The two selected algorithms – CBRW and SCAN – are obtained as Python codes from GitHub (Kaslovsky 2018; Xu 2019) and used mostly in their original form. Because the purpose of this empirical study is to test and assess the publicly available methods, no major changes to the codes are made. Since the case datasets do not contain data labels, it is not possible to measure the performance of the algorithms in terms of whether the detected anomalies are true anomalies or not. That is because it is not known beforehand which data records are anomalous or

erroneous data. Moreover, it is not feasible in this study to obtain the labels, as it would require multiple different experts to go through the data manually. That would be too time-consuming, expensive and difficult, as also identified in chapter 3.2 to be a common issue in many practical applications.

All of the datasets include some attributes with missing values as presented in Table 3, Table 4 and Table 5. Since attributes *OrderReason*, *MajorProjectType*, *ReasonForRejection* and *ActivityType* are such that there should be more values missing than present, and the records for which the values are missing exhibit similar properties, the missing values are treated as another category. For attribute *RelevantForBilling*, missing values are allowed, although rare, and can be interpreted to represent another category as well. Therefore, all of the missing values are treated as additional categories in this study. In addition, for the attribute *ActivityType*, some experiments are conducted also by excluding the missing values in order to further analyze this attribute of interest.

None of the datasets contain duplicates as such since there is always the unique identifier. Unique identifiers are excluded from the actual anomaly detection algorithms but are again included in presenting the results (anomalous records), since they are a necessary piece of information when tracking whether the found anomalies are data quality issues or actual legitimate but abnormal data records.

CBRW

CBRW algorithm was obtained as Python code from GitHub (Kaslovsky 2018). The original code outputs anomaly scores both per record and individual value for all the analyzed data records. However, because of the sizes of the case datasets, it is not feasible to examine the results manually. Therefore, and because the interest is in the anomalous records themselves, the code is set to output only the desired number of most anomalous records. The number of desired anomalies is, in this study, selected to be 0,5 % of the total number of records.

However, it should be noted that the actual anomaly detection algorithm does not flag records as anomalous or normal but gives an anomaly score that indicates which records are more or

less likely to be anomalies within the dataset. Outputting the desired number of most outlying records is an additional feature added to the code by the author of this study in order to analyze and compare the results, and this addition requires an additional input parameter – number of desired anomalies – to be given when running the code.

SCAN

SCAN algorithm was obtained as Python code from GitHub (Xu 2019). The original code is built with a test dataset including data labels. The labels are not used by the actual anomaly detection algorithm, but the labels are used in analyzing the performance of the algorithm using the Area Under the Curve of Receiver Operating Characteristic (AUC-ROC) measures. However, again, as the case dataset does not contain labels and the interest is in the anomalies themselves, the code is set to output a desired number of records with the highest anomaly scores, and the desired number of records is again selected to be 0,5 % of the total number of records.

By nature, SCAN is a stochastic algorithm, meaning that it relies on probabilistic operations and therefore the output cannot be predicted precisely. In practice, it means that the algorithm may give different results on different runs. To take the stochastic nature of the algorithm into account, the algorithm was set to run 50 times recording the anomalies from each run and then reporting the most often occurring anomalies from all runs. It was noticed that without significant modifications, the specific implementation of the algorithm cannot handle features with a high number of distinct categories. Because the scope of this study does not include major modifications to the existing algorithms, to deal with the issue, such feature – namely *SoldToParty* – was excluded from the dataset used in the algorithm comparison.

Because SCAN is a more complex algorithm in comparison to CBRW, as it is built to capture high-order complex value couplings, it can be assumed to be slower than CBRW. Since both algorithms are coupling-based algorithms, it could be assumed that they detect to some extent same records as anomalies. However, since SCAN also considers high-order complex value couplings, it is likely to detect as anomalies rather different records.

The experiments are conducted using the input parameter values recommended by the authors of the algorithms. Pang et al. (2016a) state that CBRW performs stably with damping factor $\alpha_1 \in [0{,}85; 0{,}99]$ and they employ $\alpha_1 = 0{,}95$. Therefore, also in this study, damping factor α_1 is set to 0,95. For SCAN, Xu et al. (2019) recommend using subset size factor $\alpha_2 = 0{,}15$ and representation dimensionality $r = 128$. Accordingly, parameter values $\alpha_2 = 0{,}15$ and $r = 128$ are used in this study. All experiments are performed on Intel® Core™ i9-9900X CPU @ 3.50GHz with 64GB of RAM running on 64-bit operating system.

6.3 Results

Run times

The run times of both algorithms for each dataset are presented in Table 6. The run times measure the time it takes to execute the actual anomaly detection algorithm once, while other parts of the codes, such as loading the data and printing the results, are not included in the reported run times. It can be noted that CBRW is clearly a faster algorithm for all of the three datasets.

Table 6. Run times of algorithms per dataset

	CBRW	SCAN
T1	1,6s	32,6s
T2	4,9s	13,9s
T3	240,9s	601,5s

As shown in Table 6, CBRW is faster for dataset T1 than T2, whereas SCAN is faster for dataset T2 than T1. SCAN's growing complexity to attributes is assumed to be the reason for this difference. Xu et al. (2019) report that the time complexity of SCAN is quadratic to number of attributes, while Pang et al. (2016a) claim that even though the time complexity of CBRW is theoretically quadratic to number of attributes, in practice, it is nearly linear.

However, based on some experiments and SCAN's inability to handle an attribute with high number of distinct categories, a bigger factor increasing the run time of SCAN is the complexity

when it comes to number of possible categories. Xu et al. (2019) do not report the overall complexity to number of possible attribute values, but they report the complexity of multiple steps of SCAN to be quadratic to number of possible attribute values.

Pang et al. (2016a) do not report the complexity to number of possible attribute values at all. However, the empirical experiments show that the run time of SCAN increases drastically when the number of possible attribute values increases. On the contrary, the same effect is not present for CBRW. In regard to this remark, it has to be noted that SCAN is a more complex algorithm in comparison to CBRW, as SCAN aims to capture high-order complex value couplings.

When building the theoretical framework and conducting the literature review of this study, it was noticed that in the field of anomaly detection in categorical data, algorithm complexity is considered a critical issue and scholars often report complexity of their proposed algorithm in terms of data size (number of records) and dimensionality (number of features). Only few scholars report the complexity in terms of the number of possible categories, and even in doing so, the complexity may be reported per step of algorithm and not in full, as reported for SCAN by Xu et al. (2019). However, this empirical study suggests that complexity in terms of number of distinct values is also a critical issue – at least when it comes to SCAN – and should thereby be considered and further investigated in regard to other algorithms.

Detected anomalies

The desired share of anomalies was selected to be 0,5 % of the total number of records for all datasets. Accordingly, the numbers of anomalies were set as follows: 172 for dataset T1, 849 for dataset T2 and 84 337 for dataset T3. This experiment was conducted for dataset T3 also by excluding the records for which attribute value of *ActivityType* is missing. The limited dataset then comprises of 1 047 034 records, and because the same share of anomalies was used, the desired number of anomalies was set to 5 235. To test whether the algorithms detect same or different records as anomalies, two different comparisons were made. In interpreting the results of both comparisons, it has to be taken into account that SCAN is a stochastic algorithm and may give different results if the comparisons were conducted multiple times. The stability of SCAN is analyzed later in this study.

In the first comparison, the anomalies detected by CBRW were compared to the anomalies detected by SCAN considering the most frequently occurring anomalies from 50 runs. The results are presented in Table 7. It can be observed that the algorithms detected very different records as anomalies. In this experiment the algorithms detected even completely different anomalies in dataset T2. Moreover, for other datasets the shares of same anomalies were also less than 1 %.

Table 7. Share of same records detected as anomalies, most frequent SCAN anomalies

Dataset	Same records as anomalies, most frequent SCAN anomalies from 50 runs
T1	0,58 %
T2	0,00 %
T3	0,18 %
T3, rows with missing ActivityType excluded	0,019 %

To further analyze how same or different anomalies the algorithms detect, the comparison was conducted also by comparing the CBRW anomalies to all anomalies detected by SCAN on 50 runs – not only the most frequent ones from 50 runs. In interpreting these results, it has to be noted that the number of anomalies detected by SCAN is higher than that of CBRW, as all anomalies detected by SCAN on 50 runs are recorded – even if they occurred only once. The results are reported in Table 8.

Table 8. Share of same records detected as anomalies, all SCAN anomalies

Dataset	CBRW anomalies detected as anomaly at least ones in 50 runs of SCAN
T1	15,12 %
T2	36,87 %
T3	69,57 %
T3, rows with missing ActivityType excluded	19,75 %

When considering all anomalies detected on 50 runs of SCAN in dataset T3, it can be observed that SCAN can detect a rather high share, almost 70 %, of anomalies detected by CBRW, even though the most often occurring anomalies are almost completely different. In the other

datasets, SCAN also detects some same anomalies as CBRW, but not a significant share. In conclusion, the algorithms still detect rather different records as anomalies. However, it has to be remarked that selecting a bigger share of records as anomalies is likely to increase the share of same anomalies, as now the share of anomalies was set to 0,5 %.

Whereas the actual investigation of whether the detected anomalies are data quality issues or abnormal but legitimate data records is out of the scope of this study, the detected anomalies were quickly scanned to get an overview of the results. It was noticed that SCAN detects from dataset T3 – both the original T3 and the limited T3 – some anomalies for which the values of attribute *ActivityNumber* seem to differ from the syntax of majority of the records. They may be indicative of syntactic data quality issues, but further analysis by experts with domain knowledge is needed to confirm the finding.

Since SCAN is a stochastic algorithm, it was also further analyzed whether same records often occur as anomalies or if the anomalies vary significantly from one run to another. SCAN was run 50 times on each dataset and the anomalies of each run were recorded. In T1, on each run, 172 records (0,5 % of the total number of records), were flagged as anomalies, and it led to detecting 4 701 distinct anomalies in total. Figure 6 presents the frequency distribution of the detected anomalies in dataset T1. Of all records in dataset T1, 13,63 % were detected as anomalies at least once on 50 runs. The maximum number of times any record was detected as an anomaly was 13 out of 50, which can be considered as rather low. Some records were detected as anomalies seven or more times, but the majority occurred as anomaly only once or twice.

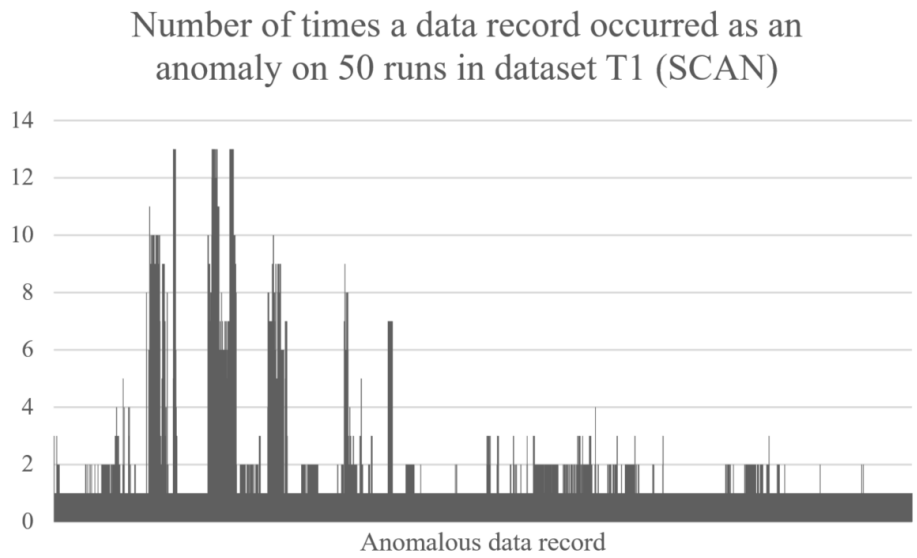

Figure 6. Frequency distribution of anomalies detected by SCAN in dataset T1 on 50 runs

The same experiment was performed also for dataset T2. On each of the 50 runs, 849 records (0,5 % of the total number of records), were flagged as anomalies, which then led to detecting 16 215 distinct anomalies in total. The frequency distribution of the anomalies in dataset T2 is presented in Figure 7. The records that were detected as an anomaly at least once on 50 runs, comprised 9,55 % of the total records in dataset T2. There were in total 381 records that were detected as an anomaly 24 times out of 50 times, which is a considerably high number – almost half of the times. Moreover, there were as many as 845 records that were detected as an anomaly more than 20 times. It is clearly visible that some records are likely to be detected as anomalies, whereas many only occur as anomalies once or twice.

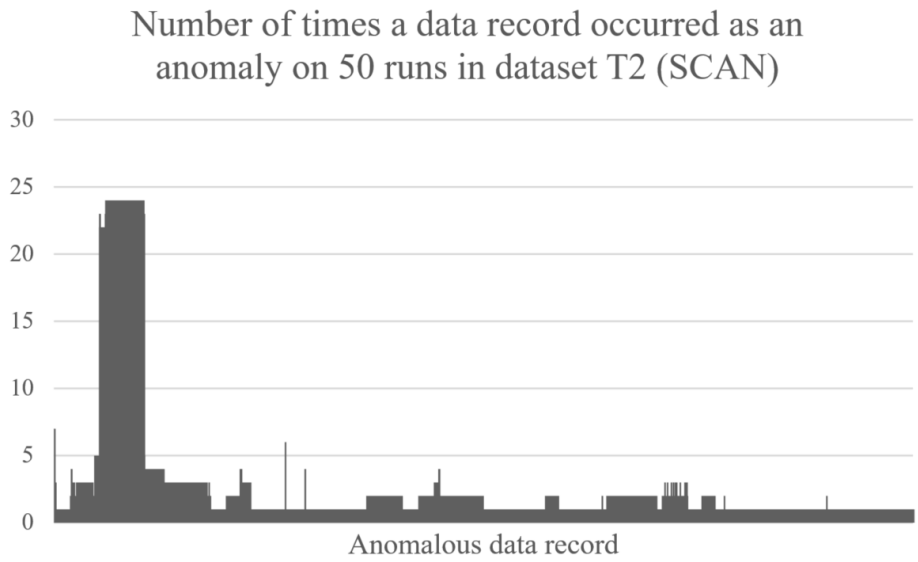

Figure 7. Frequency distribution of anomalies detected by SCAN in dataset T2 on 50 runs

The same analysis was conducted for T3 by running SCAN 50 times on the dataset and on each run flagging 84 337 records (0,5 % of the total number of records) as anomalies. It led to detecting 3 108 117 distinct anomalies in total. Majority (75,91 %) of the detected anomalies occurred as anomaly only once. Because it was not feasible to visualize more than 3 million records, Figure 8 presents a frequency distribution of the detected anomalies excluding the records which were detected as anomaly only once. Therefore, in interpreting the figure, it has to be taken into account that it presents only 24,09 % of the detected anomalies, and the rest of the detected anomalies occurred only once. Of all of the dataset T3 records, 18,43 % were detected as an anomaly at least once on 50 runs. The highest number of times any record was detected as an anomaly was 12 out of 50, and records that were detected as anomalies more than 5 times are very rare.

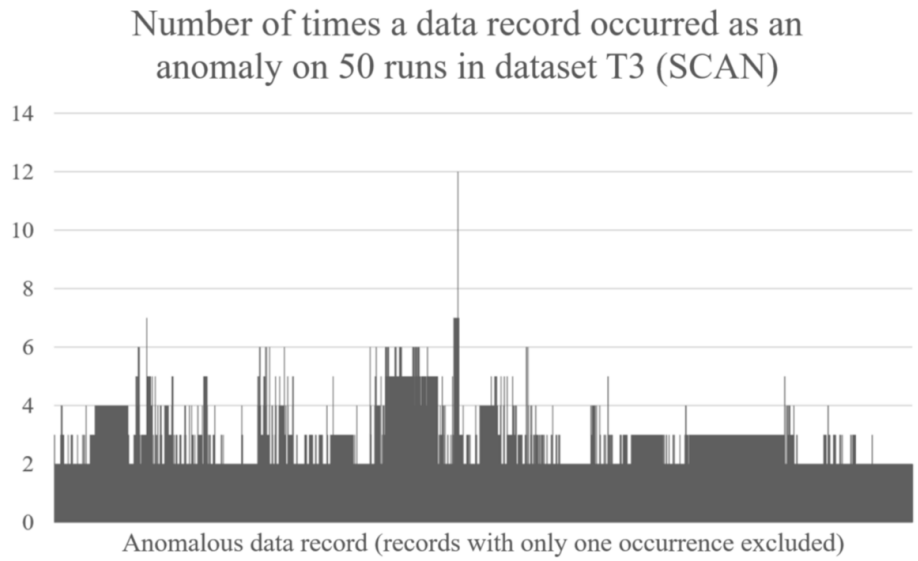

Figure 8. Frequency distribution of anomalies detected by SCAN in dataset T3 on 50 runs

As explained in chapter 6.2, attribute *SoldToParty* of dataset T1 had to be excluded from the comparison. However, because it is considered as an important attribute by case company, the CBRW algorithm is ran also including that attribute. It is analyzed how much the inclusion of that attribute affects the anomalies detected.

Running CBRW on dataset T1 with and without attribute *SoldToParty* gives completely different results when using 0,5 % as the share of anomalies. This derives from the logic of CBRW algorithm, as it computes feature weights based on how much information each feature (attribute) gives to anomaly detection. Since the feature weights are used in computing object anomaly scores, attributes with higher weights affect the anomaly scores more than the ones with lower weights.

As shown in Table 9, when *SoldToParty* is included in the dataset, it is the attribute with the highest feature weight. On the other hand, when it is excluded from the dataset, more weight is assigned to other attributes, because the weights always sum up to one. Thereby, excluding *SoldToParty* from the dataset is likely to lead to different results.

Table 9. Feature weights from CBRW for T1 with and without attribute SoldToParty

T1 Attribute	Weight with SoldToParty	Weight without SoldToParty
SalesOrganization	0,0449	0,1484
Creator	0,0726	0,3032
SalesDocumentType	0,0007	0,0029
OrderReason	0,0009	0,0092
SoldToParty	0,4615	-
SalesOffice	0,3953	0,4391
Division	0,0003	0,0038
DistributionChannel	0,0001	0,0010
MajorProjectType	0,0005	0,0031
DocumentCurrency	0,0195	0,0711
ShippingConditions	0,0036	0,0182

To further study the finding, similar analysis was performed for dataset T2. The analysis was not performed for dataset T3, because it consists of only three categorical attributes. Removing one attribute would have left only two, which may not give very reasonable results. Table 10 presents the results for dataset T2. As shown, the attribute with the highest weight in the original dataset is *Plant*, and when it is excluded, all the other attributes get higher weights. When using 0,5 % as the share of anomalies, CBRW does not detect any same record as an anomaly in dataset T2 when excluding the attribute *Plant*, compared to including it. This is in line with the results for dataset T1.

Table 10. Feature weights from CBRW for T2 with and without attribute Plant

T2 Attribute	Weight with Plant	Weight without Plant
Company	0,3562	0,4304
Plant	0,3594	-
ItemCategory	0,0489	0,1045
Currency	0,2002	0,3881
RelevantForBilling	0,0137	0,0364
ReasonForRejection	0,0215	0,0407

These experiments suggest that when performing anomaly detection, especially with CBRW algorithm, it is important to take into account that the selection of attributes affects the results and making slightly different selections may change the results drastically – or even completely.

Therefore, even after selection of attributes, it may be reasonable to analyze whether the attributes with high weights are in fact relevant in the context of detecting data quality issues, or if they should be considered noise.

Sensitivity analysis

As discussed in chapter 3, the number of input parameters is an important factor in anomaly detection. While CBRW requires only one parameter, SCAN requires two. However, the underlying reason for the importance of the number of input parameters is that the selection of parameter values may affect the anomaly detection results. Therefore, the sensitivity of these parameters is analyzed in this study.

The sensitivity analysis of CBRW's damping factor α_1 value was performed by running the algorithm with different values of $\alpha_1 \in [0{,}85; 0{,}99]$ in each dataset. As before, 0,5 % was used as the share of anomalies. The detected anomalies were then stored and compared with the anomalies detected when running the algorithm with $\alpha_1 = 0{,}95$, hereafter referred to as baseline.

The shares of same anomalies detected in dataset T1 compared to running with the baseline α_1 value are presented in Figure 9. It can be seen that the results are rather stable, around 80 % of same anomalies, except for $\alpha_1 = 0{,}99$ when the share of same anomalies is only 13,95 %. These results differ from the empirical results of Pang et al. (2016a); they state that CBRW performs stably with damping factor $\alpha_1 \in [0{,}85; 0{,}99]$. However, in this study for the dataset T1, CBRW does not perform stably anymore with $\alpha_1 = 0{,}99$, and on the other hand, it still performs stably with $\alpha_1 = 0{,}75$. Nevertheless, the results are not directly comparable, since Pang et al. (2016a) measure the AUC-ROC performance as they have labelled data, and in this study the comparison is made based on same anomalies. On the other hand, if an algorithm performs stably when it comes to AUC-ROC measures, it is likely to detect some same records as anomalies.

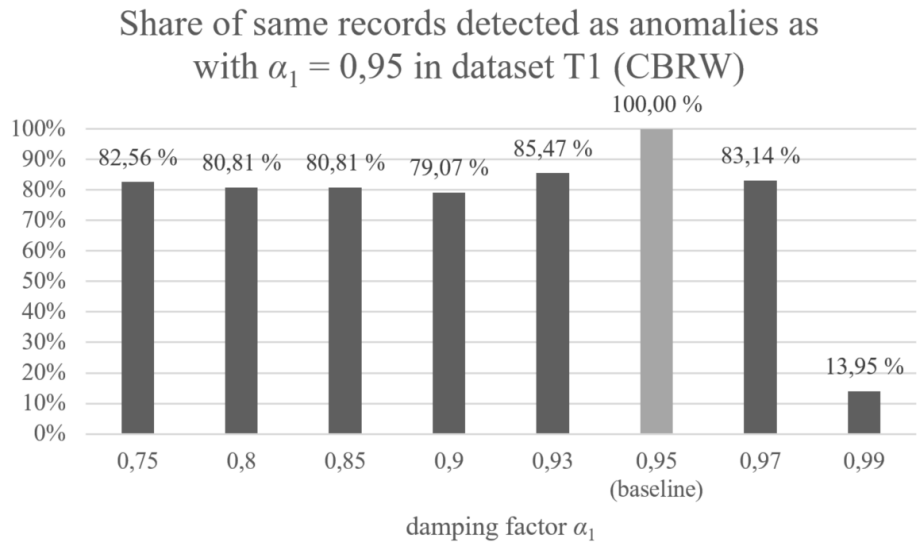

Figure 9. Sensitivity analysis for CBRW α_1, dataset T1

As shown in Figure 10, the sensitivity analysis results for CBRW α_1 with dataset T2 differ from the results with dataset T1. With dataset T2, CBRW performs very stably, detecting exactly the same records as anomalies with damping factor $\alpha_1 \in [0{,}8;\ 0{,}99]$. Only with $\alpha_1 = 0{,}75$ some of the anomalies detected are different. This result is in line with statement of Pang et al. (2016a) of CBRW performing stably with $\alpha_1 \in [0{,}85;\ 0{,}99]$.

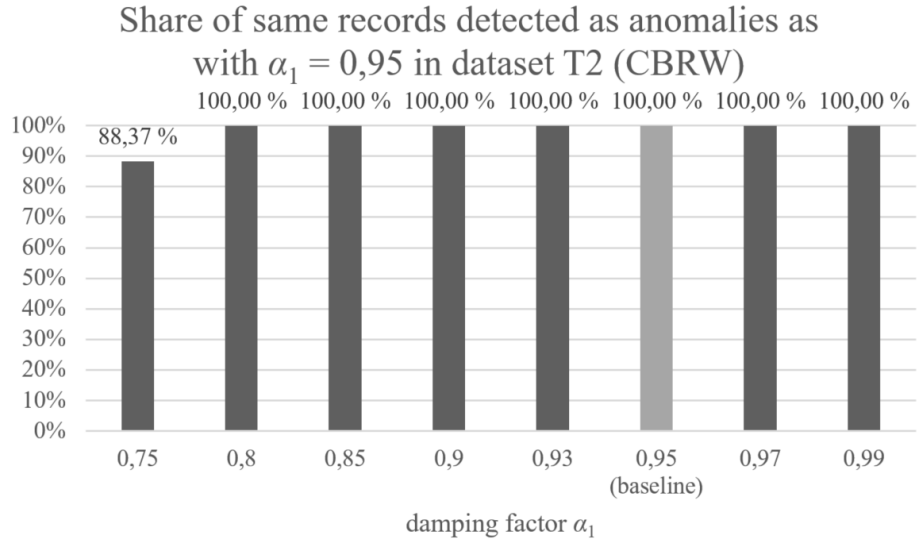

Figure 10. Sensitivity analysis for CBRW α_1, dataset T2

The results of sensitivity analysis for CBRW damping factor α_1 with dataset T3 are presented in Figure 11. As shown, CBRW performs very stably also in this dataset with different values of $\alpha_1 \in [0{,}75; 0{,}99]$, detecting at least 98,50 % and at most 99,29 % of same anomalies as with the baseline α_1 value. Therefore, it can be concluded that CBRW is rather insensitive to its input parameter damping factor α_1 value. Only in one of the three datasets, the parameter value affects the results to a noteworthy degree.

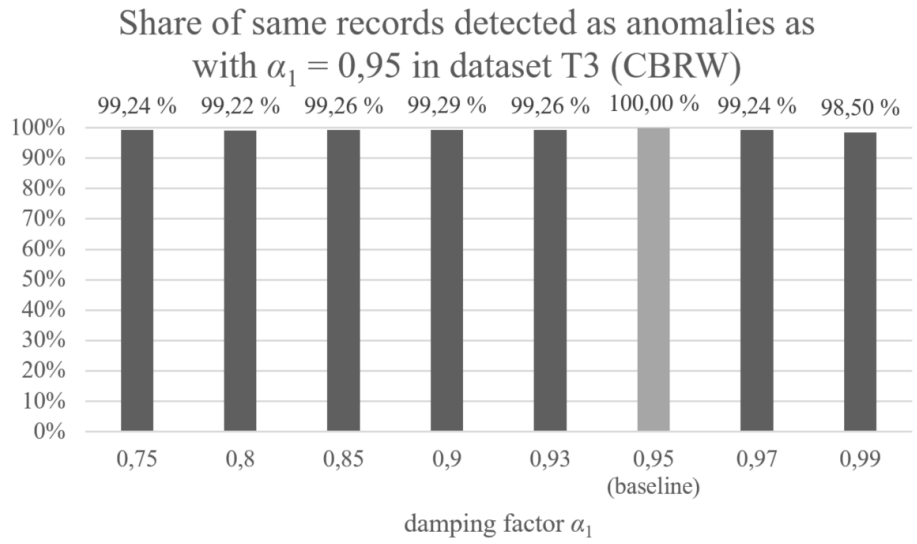

Figure 11. Sensitivity analysis for CBRW α_1, dataset T3

Because SCAN is a stochastic algorithm that can give different results from one run to another, its sensitivity is not as straightforward to analyze as it is in the case of CBRW. Therefore, SCAN is first analyzed for the stability of the algorithm without changing the parameter values. It was already analyzed how often same records occur as anomalies on 50 runs of SCAN by presenting and analyzing the frequency distributions. However, now the focus is on repeating the set of 50 runs five times and each time recording the most frequent anomalies in order to test whether SCAN is stable enough to be analyzed for the sensitivity of its parameter values.

To test the stability, the algorithm is run 50 times with the same, recommended parameter values selecting 0,5 % of records as anomalies and the most frequently occurring anomalies from the 50 runs are stored. The same is then repeated four times more and the results from all sets of 50 runs are stored. Eventually, the results of other sets are compared to the first set of

50 runs in terms of how big of a share of same anomalies were detected, in order to analyze how much the set of detected anomalies changes from one set of 50 runs to another.

Figure 12 presents the stability of SCAN in dataset T1 when ran 50 times. Each bar represents the share of same records detected as anomalies as on the first set of 50 runs of getting the most frequent anomalies. As shown, the anomalies detected by SCAN in dataset T1 vary significantly from one set of 50 runs to another. At the lowest, it detected only 23,26 % of same records as anomalies in a set compared to set 1. Therefore, it is not feasible to perform sensitivity analysis using the same setting as for CBRW, as it is not possible to analyze how much changing the parameter values affects the results, because the results are not stable even with same parameter values.

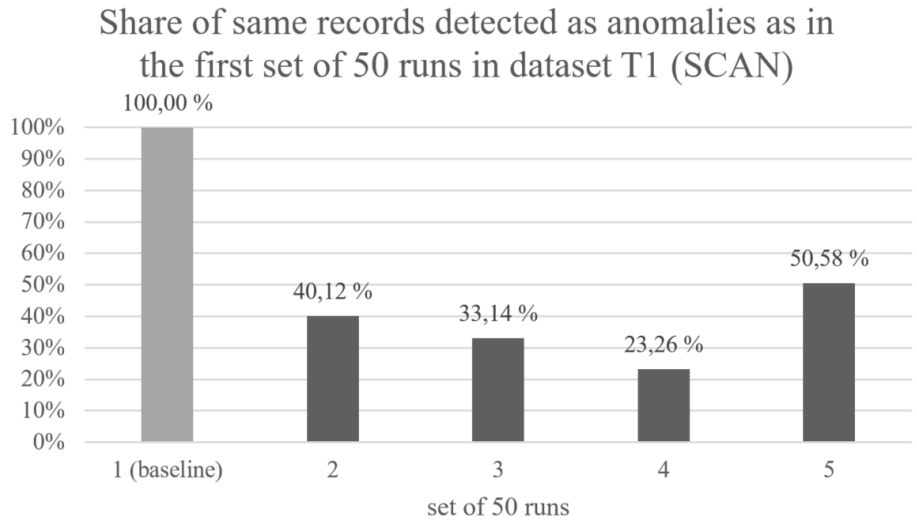

Figure 12. Stability of SCAN in dataset T1

Figure 13 presents the same experiment conducted with SCAN in dataset T2. As shown, conversely to dataset T1, SCAN performs very stably in dataset T2, detecting almost all the same anomalies between different sets of 50 runs. Accordingly, even though SCAN is stochastic, it is possible to get stable results in dataset T2 when considering the most often occurring anomalies from several runs. Thereby, there are clearly some records that tend to be more anomalous than others according to the algorithm. This is also visible in the earlier Figure 7, in which there are a considerable number of records detected as anomalies 24 times.

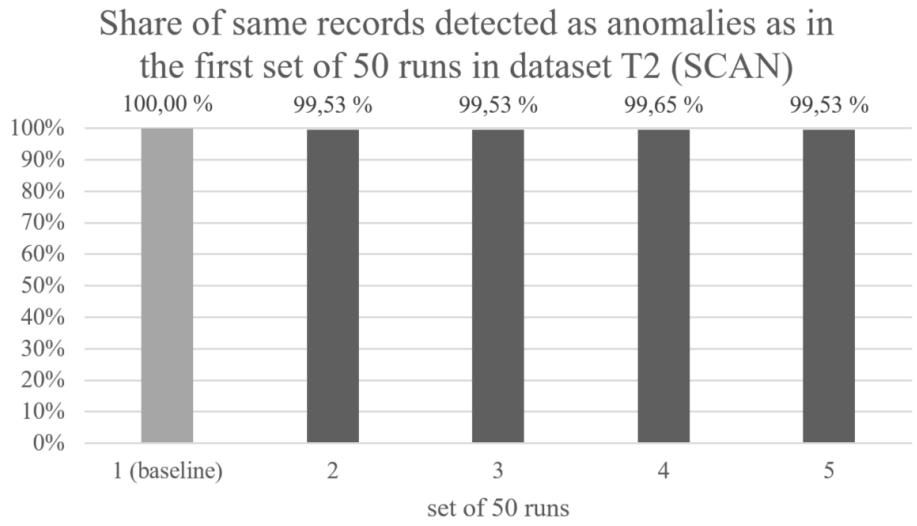

Figure 13. Stability of SCAN in dataset T2

Because the shares of same anomalies in sets 2, 3 and 5 are exactly the same and the share of same anomalies in set 4 is not far either, the most frequently occurring anomalies were also analyzed on whether they tend to be the same or not between the sets. The four anomalies that are found in set 1 but are not found in sets 2, 3 and 5 are the same. One of those four anomalies is found in set 4 but the other three are the same that are not found in set 4. However, the four (three when it comes to set 4) anomalies found in other sets, but not set 1, are different between the sets.

The process of obtaining these results first includes selecting the 849 records (0,5 % of records from the original dataset) with the highest anomaly scores from each run and then selecting the 849 most often occurring anomalies from all runs. In cases which the anomaly scores are exactly same or the number of occurrences as anomaly are the same, the anomalies are selected randomly. Therefore, it is likely that the anomalies differing between different sets are in fact initially having same anomaly scores or occurring the same number of times as anomalies, and the difference only comes from the randomness. Consequently, if the process of selecting the anomalies or the number of anomalies was changed, it might be possible to get 100 % same records as anomalies. In conclusion, SCAN can be considered as stable enough for sensitivity analysis in dataset T2.

To analyze the sensitivity of the input parameters of SCAN, the selection of anomalies was performed in a similar way as above, in order to be able to compare the results. The sensitivity analysis of SCAN's subset size factor α_2 was performed by running the algorithm with different $\alpha_2 \in [0{,}1;\ 0{,}35]$. As before, 0,5 % was used as the share of anomalies. The results were compared with the results of $\alpha_2 = 0{,}15$ used as a baseline, as recommended by Xu et al. (2019).

As shown in Figure 14, the change in the value of α_2 does not affect the results significantly, and the results are in line with the results of not changing the parameter value. Therefore, it can be argued that SCAN is insensitive to its parameter $\alpha_2 \in [0{,}1;\ 0{,}35]$ in dataset T2. This is in line with the empirical results of Xu et al. (2019) that suggest that SCAN performs stably within the aforementioned range of α_2.

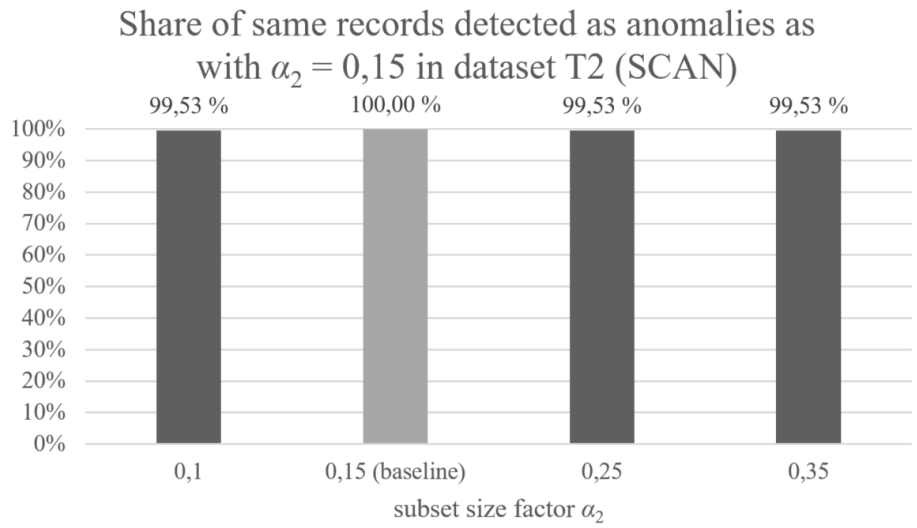

Figure 14. Sensitivity analysis for SCAN α_2, dataset T2

The sensitivity analysis of SCAN's representation dimensionality r was performed similarly to that of subset size factor α_2, but by running the algorithm with different $r \in [8;\ 512]$. The results were compared with the baseline results of $r = 128$. As shown in Figure 15, the value of r does not either have a high impact in the anomalies detected, which is again in line with the empirical results of Xu et al. (2019) that suggest that SCAN performs stably within that range of r.

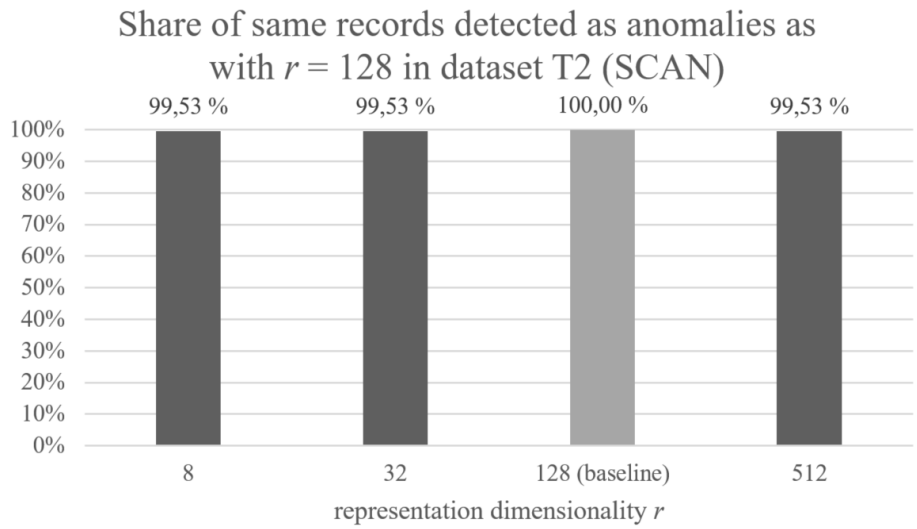

Figure 15. Sensitivity analysis for SCAN *r*, dataset T2

The stability of SCAN in dataset T3 with 50 runs is presented in Figure 16. As clearly shown, SCAN is very instable when it comes to dataset T3. It is detecting only at most 10,16 % and at least 2,08 % of same records as anomalies in a set compared to the baseline set 1. Therefore, it is again not feasible to perform sensitivity analysis using the same setting as for CBRW.

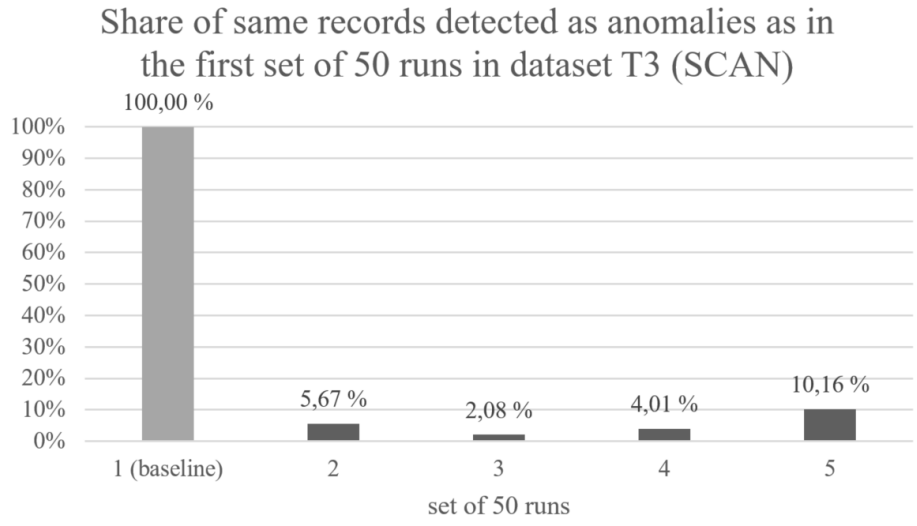

Figure 16. Stability of SCAN in dataset T3

Based on the sensitivity analyses, both algorithms seem to be insensitive to their parameters when it comes to dataset T2. CBRW is also considerably insensitive to its parameter damping factor α_1 when it comes to dataset T3. Nevertheless, CBRW does exhibit a bit more sensitivity to its parameter when it comes to dataset T1. Moreover, even running the stochastic algorithm SCAN on datasets T1 and T3 multiple times shows unstable performance, in comparison to stable performance on dataset T2.

These results indicate that the datasets T1 and T3 may not contain that clear anomalies, at least according to SCAN. On the other hand, the dataset T2 may contain more clear anomalies according to both CBRW and SCAN, even though previously concluded that the clear anomalies are different between algorithms. Furthermore, it is not possible to draw a conclusion of the sensitivity of SCAN to its parameters, but it can be concluded that CBRW is rather insensitive to its parameter.

7 CONCLUSIONS

The objective of this study was to compare and assess anomaly detection methods for detecting potential data quality issues in categorical data. The study first gave an overview of data quality, including discussion on how to ensure that data are of high quality, followed by introduction of the key concept of this study – anomaly detection. To find potential anomaly detection methods for comparison and assessment, previous literature was examined and introduced. Finally, selected algorithms were tested and evaluated. The study answers the research questions as follows:

1. *What has to be taken into consideration when selecting an anomaly detection method for categorical data?*

An important aspect to consider when selecting an anomaly detection method for categorical data is that there exists a wide range of methods that define an anomaly differently. Accordingly, one should select a method that is able to detect anomalies of the desired kind. Anomaly detection methods for categorical data are rather complex in general. Therefore, especially when applying anomaly detection to large high-dimensional datasets, the complexity of the method is an important factor to take into account. Furthermore, another important issue is the number of input parameters, as the choice of parameter values tends to affect the anomaly detection results – some methods being more sensitive than others.

2. *What are the state-of-the-art methods for anomaly detection in categorical data?*

Despite the prevalence of categorical data in practical applications, the research on anomaly detection in categorical data lacks attention compared to the same in numerical data. However, there exists a number of different methods that are based on the following ideas: density, marginal frequency, itemset frequency, Bayesian network or conditional frequency, information-theory, compression, clustering and coupling. Some algorithms, such as a coupling-based algorithm CBRW, are more often mentioned in literature to represent the state-of-the-art. Nevertheless, many of the latest proposed algorithms are coupling-based algorithms.

3. *What kind of anomalies can be found from the case data?*

The anomalies found from the case datasets differ between the selected algorithms. Moreover, anomalies detected in datasets T1 and T3 by the stochastic algorithm SCAN also differ significantly from one run to another, whereas in dataset T2, the algorithm rather often detects same records as anomalies – even with different parameter values. CBRW appears to be insensitive to its parameter value, detecting almost all the same anomalies regardless of the parameter value (within the tested range), but in dataset T1, it exhibits a slight sensitivity. These results indicate that the datasets T1 and T3 may not contain that clear anomalies, whereas the dataset T2 may contain some clear anomalies according to both CBRW and SCAN – even though the found anomalies differ between the algorithms.

The results of this study suggest that a factor that has more impact on the anomaly detection results is the selection of attributes – at least in the case of CBRW. In this study, excluding an attribute that gives most information to anomaly detection, changed the set of detected anomalies completely. Therefore, it is important to include only relevant attributes and exclude noisy attributes when applying anomaly detection.

SCAN detects as anomalies some records of which values differ in syntax from the majority of the records in dataset T3. That finding may indicate that there exist syntactic data quality issues, but to verify that, the records need to be analyzed by an expert in case company. However, there may exist less complex and more effective ways of finding syntactic data quality issues. Accordingly, the focus of this study was not in detecting syntactic data quality issues, but it turns out that while detecting more complex data quality issues, SCAN may also identify syntactic issues. When it comes to the more complex data quality issues, further investigation by experts in the case company is needed to determine whether the detected anomalies are data quality issues or actual legitimate data records.

The research area of anomaly detection in categorical data is not very mature, and there does not exist many exploitable algorithms available for the public. It is possible that any successful algorithms are not freely shared, but rather commercialized, or alternatively, there has not been

tremendous success in existing methods to spark the interest of scholars to publish new algorithms to the public. Therefore, in the lack of a complete set of easily available ready-to-use algorithms to choose from, in practice for a large-scale solution, the options are to either build own capabilities on the topic or buy the capabilities from outside.

It is recommended that the case company first further analyses the results of the anomaly detection algorithms applied in this study – investigating whether the anomalies detected by the algorithms are data quality issues or not. If either of the algorithms proves useful in the context of detecting data quality issues, it can directly be tested and used on small scale. However, since anomaly detection, or in general any other algorithm business, is not the core business of the case company, for a company-wide solution, it is recommended that the case company explores any possible solutions provided by outside vendors.

For evaluating possible anomaly detection solutions provided by vendors, the case company should first of all take into account that different anomaly detection methods base their selection of anomalies on different principles, and therefore are likely to detect different anomalies. Because the anomaly detection methods for categorical data are rather complex, it might not be feasible to analyze and determine the most suitable algorithm beforehand. Therefore, it is recommended to select a solution that comprises of multiple different algorithms that could then be run simultaneously – or otherwise test several different algorithms.

Another aspect from the complexity of the algorithms is that the more complex the algorithm, the higher its run time. Consequently, even if the technical architecture of the anomaly detection methods was out of the scope of this study, it is recommended that the case company demands from a vendor's solution a technically capable architecture providing reasonable run times with optimized processor usage and energy consumption. Furthermore, the case company should consider that the selection of attributes used by algorithms may affect the results significantly. Therefore, a user-friendly interface for selecting the desired attributes is a must requirement. It is also recommended that the user of the anomaly detection solution is capable of evaluating the relevance of attributes.

When interpreting the results of this study, a limitation that has to be taken into account is that the algorithms are only applied to a limited set of case company data, and their feasibility to any other data is not validated. Another limitation of this study is that the actual performance of the anomaly detection algorithms, in terms of whether they detect true anomalies or not, cannot be measured, because the case datasets do not contain data labels.

As a future research topic, one could consider comparing and evaluating a bigger set of different anomaly detection algorithms using real-world datasets. Because in this study it was possible to compare and evaluate only two anomaly detection algorithms, which detected rather different records as anomalies, it would be interesting to see how a bigger set of algorithms compares with each other. However, that kind of study would require expanding the criteria of algorithm selection and it may also require implementing the algorithms by oneself.

REFERENCES

Aggarwal, C.C. 2013. Outlier Analysis. Springer, New York.

Agrawal, R. & Srikant, R. 1994. Fast algorithms for mining association rules. Proceedings of the 20th International Conference on Very Large Data Bases, VLDB. Vol. 1215, pp. 487–499.

Ahmed, M. & Mahmood, A.N. 2015. Network Traffic Pattern Analysis Using Improved Information Theoretic Co-clustering Based Collective Anomaly Detection. International Conference on Security and Privacy in Communication Networks. Springer International Publishing, Cham. pp. 204–219.

Akoglu, L., Tong, H., Vreeken, J. & Faloutsos, C. 2012. Fast and reliable anomaly detection in categorical data. Proceedings of the 21st ACM international conference on information and knowledge management. pp. 415–424.

Batini, C. & Scannapieco, M. 2016. Data and Information Quality: Dimensions, Principles and Techniques. Springer International Publishing, Cham.

Boriah, S., Chandola, V. & Kumar, V. 2008. Similarity Measures for Categorical Data: A Comparative Evaluation. Proceedings of the 2008 SIAM International Conference on Data Mining (SDM). pp. 243–254

Breunig, M.M., Kriegel, H.-P., Ng, R.T. & Sander, J. 2000. LOF: Identifying density-based local outliers. *SIGMOD record*. Vol. 29, No. 2. pp. 93–104.

Calders, T. & Goethals, B. 2007. Non-derivable itemset mining. *Data Mining and Knowledge Discovery*. Vol. 14, No. 1, pp. 171–206.

Chandola, V., Banerjee, A. & Kumar, V. 2009. Anomaly detection: A survey. *ACM computing surveys*. Vol. 41, No. 3, pp. 1–58.

Dai, W., Yoshigoe, K. & Parsley, W. 2017. Improving Data Quality Through Deep Learning and Statistical Models. Information Technology - New Generations. Springer International Publishing, Cham. pp. 515–522

Das, K. & Schneider, J. 2007. Detecting anomalous records in categorical datasets. Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining. 2007 ACM. pp. 220–229.

Das, K., Schneider, J. & Neill, D. 2008. Anomaly pattern detection in categorical datasets. Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining. 2008 ACM. pp. 169–176.

Dasgupta, D. & Majumdar, N.S. 2002. Anomaly detection in multidimensional data using negative selection algorithm. Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02. Vol. 2, pp. 1039–1044.

Dasgupta, D. & Nino, F. 2000. A comparison of negative and positive selection algorithms in novel pattern detection. Smc 2000 Conference Proceedings. 2000 IEEE International Conference on Systems, Man and Cybernetics. "Cybernetics Evolving to Systems, Humans, Organizations, and Their Complex Interactions". Vol. 1, pp. 125–130.

Du, H., Ye, Q., Sun, Z., Liu, C. & Xu, W. 2021. FAST-ODT: A Lightweight Outlier Detection Scheme for Categorical Data Sets. *IEEE Transactions on Network Science and Engineering*. Vol. 8, No. 1, pp. 13–24.

Fu, Q. & Easton, J.M. 2017. Understanding data quality: Ensuring data quality by design in the rail industry. 2017 IEEE International Conference on Big Data. pp. 3792–3799.

Fujimaki, R., Yairi, T. & Machida, K. 2005. An approach to spacecraft anomaly detection problem using kernel feature space. Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining. pp. 401–410.

Ghoting, A., Otey, M.E. & Parthasarathy, S. 2004. LOADED: link-based outlier and anomaly detection in evolving data sets. Fourth IEEE International Conference on Data Mining (ICDM'04). pp. 387–390.

Goldberger, A.L., Amaral, L.A., Glass, L., Hausdorff, J.M., Ivanov, P.C., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C.-K. & Stanley, H.E. 2000. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. C*irculation*. Vol. 101, No. 23, pp. 215–220.

Goldstein, M. & Uchida, S. 2016. A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data. *PloS one*. Vol. 11, No. 4.

Grover, A. & Leskovec, J. 2016. node2vec: Scalable Feature Learning for Networks. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 855–864.

Hawkins, D. M. 1980. Identification of Outliers. Chapman And Hall, London.

He, Z., Deng, S. & Xu, X. 2005b. An Optimization Model for Outlier Detection in Categorical Data. Advances in Intelligent Computing. Springer, Berlin Heidelberg. pp. 400–409.

He, Z., Xu, X., Huang, J. & Deng, S. 2005a. FP-outlier: Frequent pattern based outlier detection. *Computer Science and Information Systems*. Vol. 2, No. 1, pp. 103–118.

He, Z., Deng, S., Xu, X. & Huang, J. 2006. A Fast Greedy Algorithm for Outlier Mining. Advances in Knowledge Discovery and Data Mining. Springer, Berlin Heidelberg. pp. 567–576.

Hodge, V. & Austin, J. 2004. A Survey of Outlier Detection Methodologies. *The Artificial Intelligence Review*. Vol. 22, No. 2, pp. 85–126.

Huang, J. 1997. A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining. *DMKD*.

Jain, A., De Simoni, G., Thoo, E., Ronthal, A., Chien, M., Feinberg, D., Zaidi, E., Parker, S., Walker, S. & Hawker, M. 2020. Cost Optimization Is Crucial for Modern Data Management Programs [online article]. Available at: https://www.gartner.com/en/documents/3986583/cost-optimization-is-crucial-for-modern-data-management- (accessed 31 July 2021).

Janakiraman, V.M. & Nielsen, D. 2016. Anomaly Detection in aviation data using extreme learning machines. Proceedings of the International Joint Conference on Neural Networks. pp. 1993–2000.

Jesus, G., Casimiro, A. & Oliveira, A. 2021. Using Machine Learning for Dependable Outlier Detection in Environmental Monitoring Systems. *ACM Transactions on Cyber-Physical Systems*. Vol. 5, pp. 1–30.

Jian, S., Pang, G., Cao, L., Lu, K. & Gao, H. 2019. CURE: Flexible Categorical Data Representation by Hierarchical Coupling Learning. *IEEE transactions on knowledge and data engineering*. Vol. 31, No. 5, pp. 853–866.

Kaslovsky, D. 2018. Coupled-Biased-Random-Walks. GitHub repository. Available at: https://github.com/dkaslovsky/Coupled-Biased-Random-Walks. (accessed 9 September 2021).

Kou, Y., Lu, C.T. & Chen, D. 2006. Spatial weighted outlier detection. Proceedings of the 2006 SIAM international conference on data mining. Society for Industrial and Applied Mathematics. pp. 614–618.

Koufakou, A. & Georgiopoulos, M. 2010. A fast outlier detection strategy for distributed high-dimensional data sets with mixed attributes. *Data Mining and Knowledge Discovery*. Vol. 20, No. 2, pp. 259–289.

Koufakou, A., Ortiz, E.G., Georgiopoulos, M., Anagnostopoulos, G.C. & Reynolds, K.M. 2007. A scalable and efficient outlier detection strategy for categorical data. 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007). Vol. 2, pp. 210–217.

Koufakou, A., Secretan, J. & Georgiopoulos, M. 2011. Non-derivable itemsets for fast outlier detection in large high-dimensional categorical data. *Knowledge and Information Systems*. Vol. 29, No. 3, pp. 697–725.

Lee, Y.W., Pipino, L., Funk, J.D. & Wang, R.Y. 2006. Journey to data quality. MIT press, Cambridge.

Li, S., Lee, R. & Lang, S.-D. 2007. Mining Distance-Based Outliers from Categorical Data. Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007). pp. 225–230.

Li, J., Zhang, J., Pang, N. & Qin, X. 2020. Weighted Outlier Detection of High-Dimensional Categorical Data Using Feature Grouping. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. Vol. 50, No. 11, pp. 4295–4308.

Liu, H., Wang, X., Lei, S., Zhang, X., Liu, W. & Qin, M. 2019. A rule based data quality assessment architecture and application for electrical data. Proceedings of the International Conference on Artificial Intelligence, Information Processing and Cloud Computing. pp. 1–6.

Mehrotra, K.G., Mohan, C.K. & Huang, H. 2017. Anomaly Detection Principles and Algorithms. Springer International Publishing.

Mohr, N. & Hürtgen, H. 2018. Achieving business impact with data. Report. Digital McKinsey.

Moore, A. & Lee, M.S. 1998. Cached Sufficient Statistics for Efficient Machine Learning with Large Datasets. *The Journal of artificial intelligence research*. Vol. 8, pp. 67–91.

Nian, K., Zhang, H., Tayal, A., Coleman, T. & Li, Y. 2016. Auto insurance fraud detection using unsupervised spectral ranking for anomaly. *The Journal of Finance and Data Science.* Vol. 2, No. 1, pp. 58–75.

Nisingizwe, M.P., Iyer, H.S., Gashayija, M., Hirschhorn, L.R., Amoroso, C., Wilson, R., Rubyutsa, E., Gaju, E., Basinga, P., Muhire, A., Binagwaho, A. & Hedt-Gauthier, B. 2014. Toward utilization of data for program management and evaluation: quality assessment of five years of health management information system data in Rwanda. *Global health action.* Vol. 7.

Olson, J.E. 2003. Data Quality: The Accuracy Dimension. Elsevier Science & Technology, San Francisco.

Otey, M.E., Ghoting, A. & Parthasarathy, S. 2006. Fast Distributed Outlier Detection in Mixed-Attribute Data Sets. *Data Mining and Knowledge Discovery*. Vol. 12, No. 2, pp. 203–228.

Pai, H.-T., Wu, F. & Hsueh, P.-Y.S. 2014. A relative patterns discovery for enhancing outlier detection in categorical data. *Decision Support Systems*. Vol. 67, pp. 90–99.

Pang, G., Cao, L. & Chen, L. 2016a. Outlier detection in complex categorical data by modelling the feature value couplings. IJCAI International Joint Conference on Artificial Intelligence. pp. 1902–1908.

Pang, G., Ting, K.M., Albrecht, D & Jin, H. 2016b. ZERO++: Harnessing the Power of Zero Appearances to Detect Anomalies in Large-Scale Data Sets. *The Journal of artificial intelligence research*. Vol. 57, pp. 593–620.

Pang, G., Xu, H., Cao, L. & Zhao, W. 2017. Selective value coupling learning for detecting outliers in high-dimensional categorical data. International Conference on Information and Knowledge Management, Proceedings. pp. 807–816.

Pang, G., Cao, L. & Chen, L. 2021. Homophily outlier detection in non-IID categorical data. *Data Mining and Knowledge Discovery*. Vol. 35, No. 4. pp. 1163–1224.

Rashidi, L., Hashemi, S. & Hamzeh, A. 2011. Anomaly detection in categorical datasets using bayesian networks. International Conference on Artificial Intelligence and Computational Intelligence. Springer, Berlin Heidelberg. pp. 610–619.

Redman, T.C. 2013. Data's Credibility Problem [online article]. Harvard Business Review. Available at: https://hbr.org/2013/12/datas-credibility-problem (accessed 28 October 2021)

Rettig, L., Khayati, M., Cudré-Mauroux, P. & Piórkowski, M. 2015. Online anomaly detection over Big Data streams. 2015 IEEE International Conference on Big Data. pp. 1113–1122.

Rossi, B., Chren, S., Buhnova, B. & Pitner, T. 2016. Anomaly detection in Smart Grid data: An experience report. 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC). pp. 2313–2318.

Sakpal, M. 2021. How to Improve Your Data Quality [online article]. Available at: https://www.gartner.com/smarterwithgartner/how-to-improve-your-data-quality (accessed 28 October 2021)

Salvador, S. & Chan, P. 2005. Learning States and Rules for Detecting Anomalies in Time Series. *Applied Intelligence (Dordrecht, Netherlands)*. Vol. 23, No. 3, pp. 241–255.

Sebastian-Coleman, L. 2013. Measuring Data Quality for Ongoing Improvement: A Data Quality Assessment Framework. Elsevier Science & Technology, San Francisco.

Singh, R. & Singh, K. 2010. A Descriptive Classification of Causes of Data Quality Problems in Data Warehousing. *IJCSI International Journal of Computer Science Issues*. Vol. 7, Issue 3, No. 2.

Smets, K. & Vreeken, J. 2011. The Odd One Out: Identifying and Characterising Anomalies. Proceedings of the 2011 SIAM international conference on data mining. pp. 804–815.

Song, X., Wu, M., Jermaine, C. & Ranka, S. 2007. Conditional Anomaly Detection. *IEEE Transactions on Knowledge and Data Engineering*. Vol. 19, No. 5, pp. 631–645.

Suri, N.R., Murty, M.N. & Athithan, G. 2012. An algorithm for mining outliers in categorical data through ranking. 2012 12th International Conference on Hybrid Intelligent Systems (HIS). pp. 247–252.

Suri, N.R., Murty, M.N. & Athithan, G. 2016. Detecting outliers in categorical data through rough clustering. *Natural Computing.* Vol. 15, No. 3, pp. 385–394.

Taha, A. & Hadi, A. 2019. Anomaly Detection Methods for Categorical Data: A Review. *ACM computing surveys*. Vol. 52, No. 2, pp. 1–35.

Tang, G., Pei, J., Bailey, J. & Dong, G. 2015. Mining multidimensional contextual outliers from categorical relational data. *Intelligent Data Analysis*. Vol. 19, No. 5, pp. 1171–1192.

Tripathi, A.M. & Baruah, R.D. 2020. Contextual Anomaly Detection in Time Series Using Dynamic Bayesian Network. Intelligent Information and Database Systems. Springer International Publishing, Cham. pp. 333–342.

Vaziri, R., Mohsenzadeh, M. & Habibi, J. 2016. TBDQ: A Pragmatic Task-Based Method to Data Quality Assessment and Improvement. *PloS One*. Vol. 11, No. 5.

Vilenski, E., Bak, P. & Rosenblatt, J.D. 2019. Multivariate anomaly detection for ensuring data quality of dendrometer sensor networks. *Computers and Electronics in Agriculture*. Vol. 162, pp. 412–421.

Wang, R.Y. & Strong, D.M. 1996. Beyond Accuracy: What Data Quality Means to Data Consumers. *Journal of management information systems*. Vol. 12, No. 4, pp. 5–33.

Wei, L., Qian, W., Zhou, A., Jin, W. & Yu, J.X. 2003. HOT: Hypergraph-based outlier test for categorical data. Advances in Knowledge Discovery and Data Mining. pp. 399–410

Wong, W.-K., Moore, A., Cooper, G. & Wagner, M. 2002. Rule-based anomaly pattern detection for detecting disease outbreaks. Eighteenth national conference on artificial intelligence. 2002 American Association for Artificial Intelligence. pp. 217–223.

Wu, S. & Wang, S. 2013. Information-Theoretic Outlier Detection for Large-Scale Categorical Data. *IEEE transactions on knowledge and data engineering*. Vol. 25, No. 3, pp. 589–602.

Xu, H. 2019. EMAC_SCAN. GitHub repository. Available at: https://github.com/xuhongzuo/EMAC_SCAN. (accessed 5 November 2021).

Xu, H., Wang, Y., Wu, Z. & Wang, Y. 2019. Embedding-Based Complex Feature Value Coupling Learning for Detecting Outliers in Non-IID Categorical Data. Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 33, No. 01, pp. 5541–5548.

Yamanishi, K., Takeuchi, J., Williams, G. & Milne, P. 2004. On-Line Unsupervised Outlier Detection Using Finite Mixtures with Discounting Learning Algorithms. *Data Mining and Knowledge Discovery*. Vol. 8, No. 3, pp. 275–300.

Yu, J.X., Qian, W., Lu, H. & Zhou, A. 2006. Finding centric local outliers in categorical/numerical spaces. *Knowledge and information systems*. Vol. 9, No. 3, pp. 309–338.

Zhao, X., Liang, J. & Cao, F. 2014. A simple and effective outlier detection algorithm for categorical data. *International Journal of Machine Learning and Cybernetics*. Vol. 5, No. 3, pp. 469–477.

Zheng, G., Brantley, S., Lauvaux, T. & Li, Z. 2017. Contextual Spatial Outlier Detection with Metric Learning. Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 2161–2170.