Rule-Based Knowledge Discovery via Anomaly Detection in Tabular Data

Asara Senaratne*,†, Peter Christen†, Graham Williams† and Pouya Ghiasnezhad Omran†

School of Computing, The Australian National University, Canberra, Australia.

Abstract

In this paper, we propose a novel approach to unsupervised detection of abnormal records in tabular data. We first characterize records in a tabular dataset using a set of features and then employ a one-class support vector machine classifier to characterize records as either normal or abnormal. We select the features that are most relevant in characterizing normal and abnormal records and apply clustering to identify groups of records that have similar characteristics according to these features. Using information-based measures, in the final step we identify the purest abnormal clusters to provide a descriptive representation that allows a user to better understand and identify abnormal records in the dataset. We evaluate our approach on datasets from three different domains, historical birth certificates, social network posts, and COVID-19 data. This evaluation demonstrates that our approach is well suited to identify anomalies in tabular data in an unsupervised manner while outperforming the baseline.

Keywords

One-class support vector machine, k-means clustering, unsupervised learning, data quality enhancement

1. Introduction

The tabular representation of data is intuitive and aids in representing large amounts of data in an engaging, easy-to-read, and structured manner. Even to date, tabular data are one of the most popular forms of data representation in businesses and even among researchers, as data tables are simple to prepare, understand, and analyze [1]. Heterogeneous tabular data, often constructed by integrating data collected from multiple sources, can contain both textual and numerical attributes. When data collection and recording are based on manual data entry or automated means of data collection, such as with sensors, data anomalies due to mistyping or misinterpretation of values, or due to a device malfunction, are not uncommon [2]. That is, we can consider the origin of data quality problems to be the structural heterogeneity of sources, human mistakes, or failing extractors. These data quality problems can range from

In A. Martin, K. Hinkelmann, H.-G. Fill, A. Gerber, D. Lenat, R. Stolle, F. van Harmelen (Eds.), Proceedings of the AAAI 2023 Spring Symposium on Challenges Requiring the Combination of Machine Learning and Knowledge Engineering (AAAI-MAKE 2023), Hyatt Regency, San Francisco Airport, California, USA, March 27-29, 2023.

*Corresponding author.

†These authors contributed equally.

asara.senaratne@anu.edu.au (A. Senaratne); peter.christen@anu.edu.au (P. Christen); graham.williams@anu.edu.au (G. Williams); p.g.omran@anu.edu.au (P. G. Omran)

0000-0002-3080-7847 (A. Senaratne); 0000-0003-3435-2015 (P. Christen); 0000-0001-7041-4127 (G. Williams); 0000-0002-4473-3877 (P. G. Omran)

CEUR Workshop Proceedings (CEUR-WS.org)

contradicting values to inconsistent entries, and outliers creating abnormalities in data. It is vital to detect such abnormalities as they are a valuable source of knowledge, which can either be corrected or passed on for manual investigations by human experts. While an abnormality may not necessarily be erroneous, we consider a data error as an abnormality in data [3].

For example, consider a dataset of birth certificates, where each row corresponds to the birth-related information of a person. In such a dataset, it is common to observe missing and invalid values which possibly arose during data entry, or the data may not have been purposely recorded in the birth certificates due to stillbirths, unknown parents, and so on. While it is easy for error detection approaches to identify such data quality errors, a solution beyond mere error detection is required to identify unusual occurrences in a dataset. For example, identifying prenuptial pregnancy, babies with adopted parents, or unmarried parents require inter-attribute assessment, which is not a capability of many existing error detection techniques. While these scenarios are not errors in data, their identification can reflect on the lifestyle and social conditions of the people who lived during the period covered by a dataset.

Over the past decades, research in data quality enhancement has resulted in several error detection and data cleaning approaches. Each of the developed algorithms usually addresses the detection and repair of only a specific error type without giving consideration to complex cross-attribute correlations for the detection of unusual occurrences. Hence, assessing data quality and identifying anomalous occurrences in real-world data are merely treated as two different problems. These problems require the application of multiple error detection algorithms based on pre-defined rules or statistical analysis to cover all anticipated data errors, and separate deployment of outlier detection techniques to identify unusual records that are worth investigating. However, running multiple such algorithms in parallel is challenging. Also, not all error detection and outlier detection strategies will be equally suitable for every dataset. Consequently, there arises the need for a holistic approach that can detect abnormal data, while also performing error detection. Anomaly detection is a data analysis task that detects anomalous data from a given dataset [4]. It is an evolving area of research as it involves discovering interesting and rare patterns in data [5].

In this paper, we propose RULEAD (RULE-based knowledge discovery via Anomaly Detection in tabular data), an unsupervised anomaly detection approach to detect abnormal records in tabular data. RULEAD performs data profiling to extract binary features, after which this feature matrix is learned to produce a set of rules describing anomalies such that it aids the reasoning process of domain experts. The ultimate goal of this research is to identify anomalies in data to generate new knowledge while also uncovering data quality issues. We evaluate our approach on datasets from three different domains: historic vital records, the COVID-19 dataset, and Reddit social network posts.

Our contributions are as follows: (1) We propose the RULEAD method for anomaly detection in tabular data with capabilities for data profiling; (2) RULEAD is unsupervised, domain-independent, and it detects abnormal records while also detecting erroneous data in tabular data, without human involvement; and (3) RULEAD produces a set of rules describing anomalies such that the output is understandable by non-technical domain experts; (4) We experimentally show how well our proposed approach performs anomaly detection on the experimental datasets, and how it outperforms the baseline, a meta data driven error detection method [1], in terms of the quality of the results and run time.

2. Related Work

While anomaly detection in tabular data is less popular compared to error detection in the literature, there are various works on data-cleaning approaches which utilize a combination of statistical, logical, or probabilistic methods. DBoost [6] is a user-guided outlier detection framework that relies on inference and statistical modeling of heterogeneous data to flag suspicious fields in database tuples. A major obstacle in data analysis is dirty data in the form of missing, duplicate, incorrect, or inconsistent values. Hence, SampleClean [7] combines statistical estimation theory, approximate query processing, and data cleaning to propose algorithms for estimating query results when only a sample of data is cleaned. Similarly, Scare [8] is a data repairing approach that is based on maximizing the likelihood of replacement data given a data distribution, which can be modeled using statistical machine learning techniques. This is a novel approach combining machine learning and likelihood methods for cleaning dirty databases by value modification. The authors have developed a quality measure of the repairing updates based on the likelihood of benefit and the number of changes applied to the database. Following the same research direction, TABBIE [9] is a table embedding model trained to detect corrupted cells.

The problem of holistic data cleaning is further addressed by Bohannon et al. in [10], where they define a database repair as a set of value modifications to repair constraints in an attempt to find low-cost changes that, when applied, will cause the constraints to be satisfied. The approach proposed in [11] is another method for the automatic repair of dirty data based on a set of user defined quality rules. Similarly, NADEEF [12] is a data cleaning platform based on user-specified data quality rules, BigDansing [13] is a big data cleaning system where users express data quality rules both declaratively and procedurally, and HoloClean [14] is a framework for holistic data repairing driven by probabilistic inference.

These repairing methods leverage various denial constraints, such as functional dependencies, matching dependencies, and inclusion constraints, to detect and repair erroneous records [1]. There are techniques proposed to automatically discover the constraints and formulae held in spreadsheets, as automatic constraint discovery can enable auto-completion, error checking, formula suggestion, rich import, and data compression. The work by Kolb et al. [15] is an attempt to investigate whether machine learning and knowledge discovery techniques can be used to learn constraints (formulae and other relations) in spreadsheet data in an unsupervised way. TaCLe [16] is a similar tabular constraint learner that aims to reconstruct the spreadsheet formulae held in the tables. However, all these techniques mainly focus on qualitative cleaning strategies. To the best of our knowledge, there is no other work in the literature that aims to detect anomalies in tabular data.

3. Proposed Approach

Our aim is to discover otherwise unidentified abnormal and erroneous records in tabular data, on the basis that they are rare occurrences in the context of a given dataset. For example, an abnormal first name might be one that is made of several long words and contains hyphens, while an abnormal telephone number would be one with numbers, special characters, and

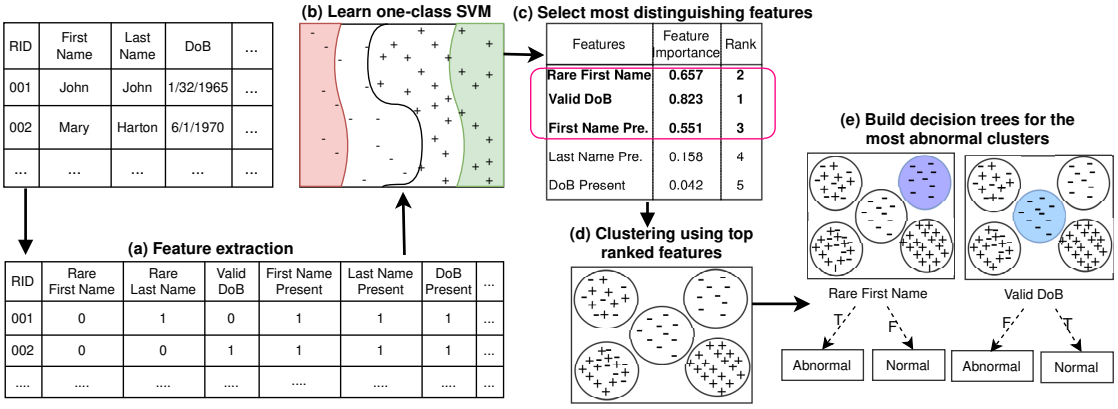

Figure 1: Overview of RULEAD, as detailed in Section 3.

letters. In this section, we describe in detail the steps of RULEAD as visualized in Figure 1.

Consider a tabular dataset D with a schema S. Let $\mathcal{A}$ be the set of attributes in S. Each tuple or record $t \in D$ consists of cells. We specify a cell of the tuple t of the attribute $a \in \mathcal{A}$ as $t[a]$. The value of each cell is represented by $v_{t,a}$. A record t in D is considered abnormal, if at least one $v_{t,a} \in t$ shows a significant deviation from the rest of the data in t or a.

3.1. Feature Extraction

Based on a set of feature generation functions $\mathcal{F}$, we automatically generate a set of binary features from the attribute values $\mathcal{A}$ of a single record t or by comparing values across multiple attributes (inter-attributes) of t in D. For example, in vital records such as birth, marriage, and death certificates, the frequency of occurrence of a name or occupation, the rareness of co-occurrence of the occupations of the parents of a baby, or the age of a mother at birth, are all possible features. The essence of generating binary features is for explainability of the final output, and ease of processing in a one-class SVM. Following is a list of feature types we generate:

1. **Frequency-based features** determine how frequent or rare $v_{t,a}$ is in a. These features primarily aim to determine any outliers in attributes.
2. **Presence check features** check for the presence or absence of the attribute values.
3. **Meta data-based features** perform data profiling [17] to determine the validity/invalidity of the attribute values after learning S of D via inferencing. RULEAD infers the widely used data type of an attribute by learning the values in it. This learning will then be used to identify wrong semantic data type affiliations such as misfielded values. For example, consider abnormal values such "g", "femalemale" in the *Gender* column, and partial dates such as "06/07/xxx", "Aug" in the *Date of Birth (DoB)* column. These are a few of the anomalies aimed to be detected by the features under this category.
4. **Multi-attribute features** are constructed considering multiple attributes together to capture inter-attribute inconsistencies. For example, the functional dependency $X \rightarrow Y$ denotes that values in the attribute Y are functionally dependent on value combinations

in the attribute combination X. We construct a multitude of features under this category considering the correlations among attributes such as *Weight*, *Height* and *BMI* attributes, ability to derive one attribute from another stored attribute such as *DoB* and *Age* attributes, related attributes such as *Postcode* and *State* where one attribute implies the value of another, and comparison attributes such as *DoB* and *Date of Marriage (DoM)*.

By deploying the feature generation functions $\mathcal{F}$ to automatically generate each feature type described above, we generate a feature matrix $\mathbf{F}$, where we have one feature vector, $\mathbf{f}$, per record t (the number of rows of $\mathbf{F}$ is $|D|$ and the number of columns is $|\mathcal{F}|$). To prune the feature space, we remove features with a single value. The step of feature extraction is shown as step (**a**) in Figure 1.

3.2. Learning Normality

While binary classification problems require training data with examples from both classes, a variation of the popular Support Vector Machine (SVM) classifier can be used to learn the decision boundary based only on a set of data points and their distribution [18], without the requirement of ground truth data for training. The one-class SVM was originally developed to identify dense areas of high-dimensional distributions. Data points are mapped into a feature space such that a given set of data points is separated from the origin with a maximum margin [19]. A one-class SVM learns a non-linear decision function where that data points that have a non-negative distance, $0 \leq d$, from the decision boundary are in the region capturing the majority of data points. This is considered as the *normal* class. The negative distances, $d < 0$, identify data points outside this region and are considered to be the *abnormal* class.

A one-class SVM achieves this separation by using a kernel function [20] (for example, a linear function, a Radial Basis Function (RBF), a polynomial, or a sigmoid), where the selection of that function depends on the nature of the data. The choice of kernel function also impacts the computational complexity in training a SVM [18]. In our approach, the input to the one-class SVM is a feature matrix, $\mathbf{F}$, as generated in the previous step. For each feature vector $\mathbf{f} \in \mathbf{F}$, the SVM returns the numerical distance $\mathbf{f}.d$ from the decision boundary. We then classify feature vectors as normal if their distance is $\mathbf{f}.d \geq 0$ or abnormal if their distance is $\mathbf{f}.d < 0$. Specifically, we generate the set of normal feature vectors, $\mathbf{N} = \{\mathbf{f} \in \mathbf{F} : 0 \leq \mathbf{f}.d\}$ and abnormal feature vectors, $\mathbf{A} = \{\mathbf{f} \in \mathbf{F} : 0 > \mathbf{f}.d\}$, where $\mathbf{F} = \mathbf{N} \cup \mathbf{A}$. In Figure 1, learning of the one-class SVM is shown as step (**b**).

3.3. Feature Selection and Ranking

This step finds the features $f \in \mathcal{F}$ that best distinguish the normal from the abnormal feature vectors as identified by the one-class SVM. We employ two attribute selection methods that are commonly used in the decision tree induction algorithms [21].

As we do not have training data (ground truth) in the form of known normal and abnormal records, we employ an approach inspired by ensemble classification methods [22], where we select two sub-sets of feature vectors to be used to calculate the ability of features to distinguish normal from abnormal feature vectors. For a given ratio r, with $0 < r < 1$, we select a set of normal and a set of abnormal feature vectors, $\mathbf{N}_r$ and $\mathbf{A}_r$. Assuming the one-class SVM has

classified less feature vectors in $\mathbf{F}$ as abnormal than normal ($|\mathbf{A}| < |\mathbf{N}|$), and the number of abnormal feature vectors, $a = |\mathbf{A}|$, then we select $r \cdot a$ normal feature vectors into $\mathbf{N}_r$ and the same number of abnormal feature vectors into $\mathbf{A}_r$.

These feature vectors are selected to have the largest absolute distances, $|\mathbf{f}.d|$, from the SVM decision boundary. As the feature vectors closer to the decision boundary are less normal or less abnormal compared to the feature vectors further away from the boundary, we select several such sub-sets of feature vectors with different ratio values, r, to be combined in a weighted fashion.

Each pair of sub-sets of selected feature vectors, $\mathbf{N}_r$ and $\mathbf{A}_r$, with $\mathbf{N}_r \subset \mathbf{N}$ and $\mathbf{A}_r \subset \mathbf{A}$, for different values of r, is then used to identify the features that are best suited to distinguish normal from abnormal feature vectors in $\mathbf{N}_r$ and $\mathbf{A}_r$. We employ the two attribute selection measures, also known as impurity measures, Gini and entropy [21]. These measures provide a score for each feature indicating how much information is gained by separating normal from abnormal feature vectors. A higher information gain score means a feature is better able to distinguish between normal and abnormal feature vectors. For each pair of subsets $\mathbf{N}_r$ and $\mathbf{A}_r$, we will therefore obtain a list $\mathbf{g}_r$, containing pairs of features and their information gain scores, or entropy $(f_{r,i},\ g_{r,i})$.

With different sized subsets $\mathbf{N}_r$ and $\mathbf{A}_r$ of selected normal and abnormal feature vectors, potentially different features will have the highest information gain scores. To obtain a robust set of features that are useful to distinguish normal from abnormal records, we calculate a weighted sum for each feature which indicates how well it distinguishes normal from abnormal. We specifically give higher weights to features that have been selected based on smaller training sets (that consist of the more normal and abnormal feature vectors), where r is set to a small value. Lower weights are used for features that have been selected based on larger training sets (with less separation), where r is set to a larger value. For each feature $f_i \in \mathcal{F}$ we calculate an overall weight as:

$$w_i = \sum m_r \cdot g_{r,i}, \tag{1}$$

where m_r is the weight assigned to features identified using the subsets $\mathbf{N}_r$ and $\mathbf{A}_r$, and $g_{r,i}$ is the information gain score or entropy of feature f_i as calculated based on the subsets $\mathbf{N}_r$ and $\mathbf{A}_r$. The features f_i with the highest weights w_i are best suited to distinguish normal from abnormal feature vectors. In Section 4, we present the features ranked using this method for all experimental datasets.

The output of this third step of RULEAD, step (**c**) in Figure 1, is a ranked list of features that are best suited to distinguish the normal and abnormal feature vectors in $\mathbf{F}$. In the following step, we apply clustering on the feature vectors in $\mathbf{F}$ to identify groups of similar feature vectors.

3.4. Clustering Feature Vectors

Given the objective of our approach is to automatically identify abnormal records in tabular data, our next task is to identify groups of records that are similar to each other based on the most discriminating feature vectors, and then identify the groups that are most abnormal. We accomplish this using clustering of feature vectors using the identified subset of features that are best able to distinguish normal from abnormal feature vectors.

We first select a subset of features for clustering, $\mathcal{F}_c \subset \mathcal{F}$ based on their weights w_i as calculated in the previous step using Equation 1. This selection can either be based on a certain threshold t (select all features with a weight $w_i \geq t$), or a certain number of the features with the highest weights w_i. An important aspect is that if too many features are selected then the clustering process described next might suffer from the curse of dimensionality [23] where no clear distinction between normal and abnormal feature vectors can be identified.

We generate the feature matrix $\mathbf{F}_c$, where each feature vector $\mathbf{f} \in \mathbf{F}_c$ only contains the features in $\mathcal{F}_c$ selected based on their weights. Each feature vector in the original feature matrix $\mathbf{F}$ will be represented in $\mathbf{F}_c$ by a lower dimensional feature vector, i.e. $|\mathbf{f}_c| < |\mathbf{f}|$. We then apply k-means clustering [24] on the feature matrix $\mathbf{F}_c$, where we use different values of k, obtaining a set of clusters $\mathbf{C}_k = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k\}$.

From the one-class SVM, we know for each feature vector $\mathbf{f} \in \mathbf{F}_c$ whether it has been classified as normal or abnormal. We can therefore calculate the quality of each cluster as its purity with regards to the abnormal class of the feature vectors allocated to that cluster, $p_a(\mathbf{c}_i) = c_A/|\mathbf{c}_i|$, where c_A is the number of abnormal feature vectors in cluster $\mathbf{c}_i$. As illustrated in step (**d**) in Figure 1, the output of this step is a clustering $\mathbf{C}_k$ containing k clusters, each consisting of feature vectors from $\mathbf{F}_c$, where for each cluster we have also calculated its abnormality purity measure, p_a. In the final step of our approach, described next, we generate descriptive representations of the clusters in $\mathbf{C}_k$ with highest values of p_a.

3.5. Describing Abnormal Clusters

The final step of our approach aims to create a descriptive representation of the feature vectors in the clusters $\mathbf{c}_i \in \mathbf{C}_k$ generated in the previous step that contain mostly abnormal feature vectors. We first identify those clusters $c_i \in \mathbf{C}_k$ that have a high abnormal purity, $p_a(\mathbf{c}_i)$. The original feature matrix, $\mathbf{F}$, is then augmented with a target feature that identifies a feature vector as being in cluster $\mathbf{c}_i$ or not. This provides a two-class training dataset (for each cluster) for a decision tree classifier from which we can generate rules to describe the cluster [25]. The resulting trained tree is converted into a rule set that describes the feature vectors within the abnormal cluster $\mathbf{c}_i$. These are the paths (rules) in the tree from the root node to the leaf nodes that correspond to the abnormal class.

The aim is to obtain a descriptive representation of the abnormal cluster rather than an accurate classification model. The descriptions explain, in language easily accessible to domain experts, why a cluster of abnormal feature vectors is different from all other feature vectors generated from the records in the dataset. Furthermore, the training of the decision tree is specifically not based on the normal and abnormal labels generated by the one-class SVM, but on the feature vectors within a selected abnormal cluster $\mathbf{c}_i$ (the *positive* class) against all other feature vectors (the *negative* class).

The final output of this step (step (**e**) in Figure 1) is a collection of rules that describe groups of abnormal records in a tabular dataset. The intent is that these can be reviewed by the domain experts in order to determine whether they represent aberrant data or otherwise interesting and unexpected cases.

4. Experimental Evaluation

We evaluate our approach using two different categories of datasets. The first contains three datasets where one is from the domain of historical vital records, the second is a Reddit dataset, and the third dataset is a COVID-19 dataset. For these three datasets, we provide the results obtained using our approach followed by a discussion of the results in Section 4.2. The second category contains the three datasets named salaries, flights, and hospital that we obtained from the baseline approach [1]. We use these three datasets for a comparative evaluation of our approach with the baseline in Section 4.3.

We implemented our approach using Python 3.0 and the machine learning package *Scikit-learn* [26]. We ran all experiments on a 64-bit MacBook Pro with an Apple M1 processor, 8 GB of memory, and MacOS Ventura v13.0. The source code is available online for reproducibility[1].

4.1. Parameter Settings

We used various parameter settings to investigate the behavior of our approach. We trained the one-class SVM using a RBF kernel due to the non-linearity of data and efficiency of RBF in comparison to other kernels such as polynomial. We set the parameters γ (kernel coefficient) and ν (the upper bound on the fraction of training errors and a lower bound on the fraction of support vectors) of RBF to $\gamma = 0.1$ and $\nu = 0.1$ [18], respectively.

As discussed in Section 3.3, we then selected several subsets of the most normal and most abnormal feature vectors, based on the one-class SVM classification. We set the ratio parameter $0 < r < 1$ to select subsets of sizes 250, 500, 1,000, 2,000, and 3,000 feature vectors each into $\mathbf{N}_r$ and $\mathbf{A}_r$, respectively. We then used the Gini and entropy attribute selection measures [21] to calculate scores for each feature using different sized subsets. The weights we assigned to feature scores from the different sized subsets are $w_{250} = 3$, $w_{500} = 2$, $w_{1,000} = 1$, $w_{2,000} = 0.75$, $w_{3,000} = 0.5$. We assigned these weights in such a way that we can obtain a robust set of features that are useful to distinguish abnormal from normal records. Finally, we conducted k-means clustering with the number of clusters set to $k = [10, 20, 30, 40, 50]$.

4.2. Experimental Datasets

We evaluate our abnormality detection approach using datasets from the domains of historical vital records, social networks, and COVID-19. The first is a Scottish dataset [27] containing 17,614 birth records covering the population of the Isle of Skye (IoS) over the period from 1861 to 1901. Each record is a birth certificate and contains personal information about the baby and its parents, such as their names, address, and so on. The Reddit dataset[2] represents the directed links of a Reddit post to a Reddit community (known as sub-reddits) [28]. This dataset was extracted from publicly available Reddit posts between January 2014 to April 2017. The third dataset[3] (named 'Israel') is a publicly available dataset from the Israeli Ministry of Health. It contains data about individuals who have got tested for COVID-19 [29]. As none of these

[1] https://github.com/AsaraSenaratne/RULEAD
[2] https://snap.stanford.edu/data/soc-RedditHyperlinks.html
[3] https://data.gov.il/dataset/covid-19

Table 1
Dataset summaries.

	Isle of Skye births	Reddit posts	COVID-19 data
Number of records	17,614	35,776	278,849
Number of attributes	37	18	10
Number of features extracted	104	86	52

datasets have labelled data, we assess the ability of RULEAD in detecting anomalies by manually evaluating the output.

Table 1 summarizes the sizes and the numbers of features we extracted from the three datasets, and Table 2 provides a summary of the dataset attributes. RULEAD automatically generated 104, 86, and 52 binary features for Isle of Skye, Reddit, and COVID-19 datasets, respectively. The constructed features contain a combination of data quality and semantic features that aim to identify anomalous records whilst also detecting any erroneous data. We show the top most weighted features that we selected from our feature selection method for the Isle of Skye dataset in Table 3, for the Reddit dataset in Table 4, and for the COVID-19 dataset in Table 5.

Table 2
Attribute summaries of the datasets.

Isle of Skye births
- Information about the baby: First and last name, DoB, address. - Information about the parents: Father's first and last name, occupation; mother's first, last, and maiden name, occupation; DoM and place of marriage.
Reddit posts
- Source and target sub-reddits: Link starting and ending points. - Post identifier (ID): The post in the source sub-reddit that starts the link. - Post Label: Binary value noting if the source post is negative towards the target. - Post Properties: A vector of 86 numerical features.
COVID-19 data
- Personal information: gender (male or female), age information (age above 60 or not). - Test result: whether the person has a negative or positive test result for COVID-19. - Source of infection: if positive, did the patient contract it overseas or from a known contact. - Presence of five initial clinical symptoms: cough, cold, shortness of breath, sore throat, and fever.

Based on the feature importance obtained from the feature selection and ranking, we selected the top ranked features for each of the datasets. Table 3 lists the top ranked Isle of Skye features and their weights. The top ranked features determine whether the child's last name matches the last name of the father and maiden name of the mother, whether the child has a first name recorded, how common the last name is within the population of data it occurs, whether the child was born before the marriage of the parents, and the commonality of the child's date of birth. Upon inspection, rareness and commonality appear to be a better source of detecting

data quality errors. Reviewing the availability of names and the match between them pinpoint to family conditions whilst also highlighting data entry errors. If a child was stillborn, it may not have been given a first name (the feature *first name present* will be 0). If at birth the child's parents were not married, the child may not have a father mentioned in the birth certificate. A child's date of birth can be before parents' marriage date in the case of prenuptial pregnancy.

Table 3
Top ranked Isle of Skye features.

Features	Weight
Child's last name matches father's last name	33
First name present	27.25
Child's last name matches mother's maiden name	27
Mother's last name present	23.5
Common father's last name in population	21.75
Common last name in population	20
Child's DoB before parent's DoM	14.25
Common DoB in population	12

The Reddit features reflect characteristics of posts written within a sub-reddit. The top ranked features listed in Table 4 mostly resemble the characteristics of a written post, such as word length, number of words, characters, white spaces, stop words, unique words, special characters and the number of words occurring in a sentence.

Table 4
Top ranked Reddit features.

Features	Weight
Average word length	72.5
Number of words	60
Fraction of white spaces	58
Number of unique works	37.5
Fraction of special characters	27.25
Number of unique stop words	23.75
Number of sentences	16
Average number of words per sentence	12.5

The top ranked features of the COVID-19 dataset mostly aid in determining patterns involved with the spread of the disease, over reflection of data quality. As listed in Table 5, these features determine whether a person is above 60 years or not, whether the person is a male or female, whether the person has both cough and sore throat as symptoms, does the person have fever, and whether a person tests positive even without any symptoms.

Table 6 provides a summary of the output of the one-class SVM, including a summary of the count of feature vectors provided as input to the SVM, and the counts of vectors classified as normal and abnormal, and the range within which these records lay within the hyper plane.

Using these identified top ranked features, we then clustered the entire feature matrix (as

Table 5
Top ranked COVID-19 features.

Features	Weight
Is age above 60	41
Is male	40.75
Having cough and sore throat	40
Has fever	39
No symptoms but positive to COVID	38

Table 6
Output summary of the one-class SVM classifier model.

Dataset	Total number of records	Number of normal vectors	Number of abnormal vectors	Range of anomaly scores
Isle of Skye	17,614	14,084	3,530	-55.04 to 30.79
Reddit	46,718	37,374	9,344	-3439.15 to 732.8
COVID-19	278,849	273,823	5,026	-8005.02 to 888

described in Section 3.4). The purpose of applying clustering is to identify the most unusual clusters that are purely abnormal or nearly pure. For the next step of building decision rules, we selected clusters with an abnormal purity of $p_A() \geq 0.8$. We selected 0.8 as the threshold with the intention of selecting only highly pure clusters. As described in Section 3.5, the abnormal feature vectors selected from each pure to near pure clusters were further classified against all the other feature vectors to obtain the rules that make them more abnormal. We show the rules we obtained for Isle of Skye, Reddit, and COVID-19 datasets in Table 7, Table 8, and Table 9, respectively.

Table 7
Abnormal decision rules generated from the Isle of Skye dataset features.

1. (first name missing) AND (child's last name matches mother's maiden name)
2. (child's last name matches mother's maiden name) AND (father's last name is rare in population)
3. (rare first name in parish) AND (rare mother's maiden name in parish) AND (rare gender in population)
4. (child's last name does not match father's last name) AND (child's last name does not match mother's maiden name) AND (child's last name does not match mother's last name)
5. (rare parish in population)

It is plausible to categorize a birth record as anomalous if it has no first name. This can either mean there is a data entry error or the birth was a stillborn. These explanations are what we hope to extract from domain experts. Also, the rare occurrence of a parish, date of birth, and gender means they can actually be outliers in the data or be data entry errors. The first rule in Table 7 defines those groups of records with a missing first name and the child's last

name matching mother's maiden name possibly referring to those children with a stillbirth and without father's identity, which can also be the case of the second rule. As for the third rule, this could refer to records with data entry errors as they have a rare gender. The fourth rule represents the group of children without a surname which can be the case of adopted children. The last rule highlights the group of records with births that have happened in a less populated parish, thus making the parish rare within the dataset.

Table 8
Abnormal decision rules generated from Reddit dataset features.

1. (high no. of words) AND (low average word length) AND (high no. of stop words)
2. (high no. of words) AND (low no. of characters)
3. (low no. of stop words) AND (low no. of unique stop word)
4. (high no. of words) AND (high fraction of stop words)
5. (low no. of characters)
6. (high no. of characters without counting white space) AND (low no. of unique stop words)

As the decision rules for the Reddit feature vectors in Table 8 show, too many occurrences of a particular word or words, and stop words are considered abnormal. A post with a high number of words that are short is considered abnormal, and at the same time posts with high word counts but with less stop words are considered anomalous too. The rules obtained are agreeable as any sentiment analysis process would first rely on these features to determine the nature of a post. Because any junk post doesn't really convey a message and hence, the usage of vocabulary can be unsettling. For example, the reddit post *"Even tho i read the new karma, im not sure how to get it.... i never needed it since i just scroll witht mkn maslf visible but now i wanna hv 10 karma points.... cn sm1 gv me lyk 5?"* is considered abnormal and reflected by rule number 2 in Table 8.

Table 9
Abnormal decision rules generated from COVID-19 dataset features.

1. (age above 60) AND (is male)
2. (no symptoms) AND (positive result)
3. (missing source of infection)
4. (has sore throat) AND (has fever) AND (negative result)

While there are not many clusters of abnormal records identified in the COVID-19 dataset, the clusters reflected by the four rules in Table 9 very well define those records exhibiting abnormal properties. The first rule highlights the group of abnormal records representing older males. From the population statistics of the COVID-19 dataset, 82% of the people who have come forward for testing are females. In the remaining 18%, 3% are older males. While the number of older males who have been tested for COVID-19 is comparatively less, we can assume that this rareness is either because they have not come forward for testing, or they are less susceptible to COVID-19, or there is a higher mortality rate among this group of people. The second and fourth rules represent those obvious anomalous scenarios of being positive to COVID-19 without any symptoms, or obtaining a negative result despite being sick. It is also

rare to find records without the source of infection in the dataset, which either represent those who have not reported their source of infection, or those unaware of it.

4.3. Benchmark Comparison

We compare RULEAD with the meta-data driven error detection approach proposed by Visengeriyeva et al. [1] using the three datasets provided by its authors. These datasets are available online[4] together with the corrupted versions. The baseline work adopts ensemble learning and incorporates metadata extracted from a dataset for error detection. This work is capable of detecting outliers, inconsistencies, conflicting records, duplicates, and violations of syntactic and semantic patterns. While there is limited work in the domain of unsupervised anomaly detection in tabular data, due to the diversity in the types of errors this approach can consider, and due to adoption of ensemble learning, we find this work the most suitable to be considered as the baseline of RULEAD.

The hospital dataset is available on the US Department of Health and Human Services. This data set comprises 10,000 records with 18 attributes of mainly textual and categorical data types, such as addresses, ZIP codes, state codes, and hospital names. This dataset is supposedly error-free. The authors of [1] have generated a dirty version of this dataset using the BART system [30]. They have configured BART to insert denial constraints violations by changing values in data fields. The inserted error percentage is 9.2%.

The salaries dataset[5] is another real-world dataset that contains the names, job titles, and salaries of San Francisco city employees on an annual basis from 2011 to 2014. This dataset consists of 75,000 different data points. The biggest fraction of this dataset are numerical values. The dirty variant of SALARIES is again produced by introducing 2.33% errors with the BART system [30]. The authors of [1] have configured BART to produce numerical outliers spread on six payment attributes: basepay, overtime pay, other pay, benefits, total pay, total pay benefits.

The flights dataset represents a fusioned dataset from 38 deep web sources related to the flights domain. The collected information of over 1,200 flights over a one month period (December 2011) is accompanied by the gold standard, created by the authors of [31]. The resulting dataset comprises 74,000 records with an error percentage of 61.85%.

Table 10
Summary of the benchmark datasets.

Particulars	Hospital	Salaries	Flights
Number of attributes	18	13	9
Number of records	10,000	75,000	74,000
Percentage of errors generated	9.2%	2.3%	61.85%
Number of features extracted by RULEAD	30	22	15

We compare our approach with the baseline in terms of precision, recall, and run time. While we ran the source code of the baseline work as provided by the authors[6], we also ran our

[4] https://github.com/visenger/clean-and-dirty-data
[5] https://www.kaggle.com/datasets/kaggle/sf-salaries
[6] https://github.com/visenger/DetectEr

approach on the above three datasets and evaluated against the ground truth available. Table 11 provides a comparative evaluation of the results obtained by our approach and the baseline.

Table 11
Comparative evaluation of the baseline. The abbreviation *R.T* indicates the *run time* in *seconds*.

Approach	Hospital			Salaries			Flights		
	Precision	Recall	R.T.	Precision	Recall	R.T.	Precision	Recall	R.T.
Baseline	87.02	87.00	1,506	77.68	77.60	2,410	74.29	73.73	2,431
RULEAD	**90.12**	**89.11**	**1,220**	**78.21**	**78.00**	**911**	**77.55**	**76.09**	**900**

As per Table 11, RULEAD performs better in identifying the anomalies at a lower run time in comparison to the baseline under consideration. As feature generation is the step taking the highest amount of time in our proposed approach, the hospital dataset has the longest run time due to high number of attributes in comparison to the other two datasets, despite their numbers of records. In contrast, the baseline approach [1] has a long run time as the number of records increases. As BART is used to synthetically generate errors in the three benchmark datasets, we manually evaluated a few of the errors that RULEAD failed to identify. A summary of the analysis is provided below:

1. While RULEAD can detect duplicate entries in a dataset, it cannot identify records belonging to the same entity with contradicting data.
2. In a categorical variable where the number of categories is small (such as for a *Gender* attribute), if an error is introduced in such a way that the new value becomes an outlier, RULEAD can identify such records. However, a change in a value of a field such as *First Name*, where there are many unique values, will not be triggered as anomalous by our approach.
3. RULEAD identifies abnormal records, not individual cells. Hence, minor changes introduced to a cell which does not make the change apparent will not be detected by RULEAD. For example, changing a salary value from 100,000 to 110,000, where the range of the salary column is 90,000 to 150,000, will not be identified as an anomaly.

5. Conclusion and Future Work

We presented RULEAD, an unsupervised anomaly detection approach to detect abnormal records in datasets. Unlike other work in this domain, we do not perform error detection, but anomaly detection considering the attributes and their inter-dependencies. Our approach can provide users with a view on the anomalous nature of their data including reasons for such a decision in an efficient and unsupervised manner.

Our algorithm consists of the following steps: feature extraction from attributes, learning normality to determine normal and abnormal feature vectors, ranking and selection of the most distinguishing features, clustering using the selected features, and finally providing a symbolic representation of the decision rules behind the anomalies detected. These rules can be used by domain experts to gain further insights into the records in a graph. A record becomes abnormal

when the data associated with it is unusual (does not comply with what is expected in the attribute), missing, contradicting, or invalid. While it is easier to detect missing and erroneous data, much focus should be given to the detection of anomalous records that look normal in nature.

As our experiments using the three experimental datasets showed, the top ranked most distinguishing features from the three scenarios have helped to extract the most abnormal clusters with high abnormal purity and meaningful decision rules which can be presented to a domain expert for further analysis. The rules identified from these abnormal clusters explain the reasons why records are abnormal. One important aspect of our approach is that, rather than treating attributes in isolation, RULEAD can consider multiple attributes together to make connections and comparisons among them. Thus, making hidden scenarios discoverable.

As for future work, we aim to develop approaches to determine the most effective set of features from a group of features created to train the one-class SVM, such that curse of dimensionality is eliminated. Finally, we also aim to propose an approach for automatic correction of the identified errors based on the rules discovered.

References

[1] L. Visengeriyeva, Z. Abedjan, Metadata-driven error detection, in: ICSSDM, 2018, pp. 1–12.

[2] P. Christen, R. Schnell, Thirty-three myths and misconceptions about population data: from data capture and processing to linkage, IJPDS 8 (2023).

[3] A. Senaratne, P. Christen, G. Williams, P. G. Omran, Unsupervised identification of abnormal nodes and edges in graphs, JDIQ 15 (2022) 1–37.

[4] L. Akoglu, H. Tong, J. Vreeken, C. Faloutsos, Fast and reliable anomaly detection in categorical data, in: CIKM, ACM, Maui, 2012, p. 415–424.

[5] M. Ahmed, A. N. Mahmood, J. Hu, A survey of network anomaly detection techniques, JNCA 60 (2016) 19–31.

[6] C. Pit-Claudel, Z. Mariet, R. Harding, S. Madden, Outlier detection in heterogeneous datasets using automatic tuple expansion (2016).

[7] S. Krishnan, J. Wang, M. J. Franklin, K. Goldberg, T. Kraska, T. Milo, E. Wu, Sampleclean: Fast and reliable analytics on dirty data., IEEE Data Eng. Bull. 38 (2015) 59–75.

[8] M. Yakout, L. Berti-Équille, A. K. Elmagarmid, Don't be scared: use scalable automatic repairing with maximal likelihood and bounded changes, in: SIGMOD, ACM, 2013, pp. 553–564.

[9] H. Iida, D. Thai, V. Manjunatha, M. Iyyer, Tabbie: Pretrained representations of tabular data, arXiv preprint arXiv:2105.02584 (2021).

[10] P. Bohannon, W. Fan, M. Flaster, R. Rastogi, A cost-based model and effective heuristic for repairing constraints by value modification, in: SIGMOD, 2005, pp. 143–154.

[11] X. Chu, I. F. Ilyas, P. Papotti, Holistic data cleaning: Putting violations into context, in: ICDE, IEEE, 2013, pp. 458–469.

[12] M. Dallachiesa, A. Ebaid, A. Eldawy, A. Elmagarmid, I. F. Ilyas, M. Ouzzani, N. Tang, Nadeef: a commodity data cleaning system, in: SIGMOD, ACM, 2013, pp. 541–552.

[13] Z. Khayyat, I. F. Ilyas, A. Jindal, S. Madden, M. Ouzzani, P. Papotti, J.-A. Quiané-Ruiz, N. Tang, S. Yin, Bigdansing: A system for big data cleansing, in: SIGMOD, ACM, 2015, pp. 1215–1230.
[14] T. Rekatsinas, X. Chu, I. F. Ilyas, C. Ré, Holoclean: Holistic data repairs with probabilistic inference, Proc. VLDB Endow. 10 (2017) 1190–1201.
[15] S. Kolb, S. Paramonov, T. Guns, L. De Raedt, Learning constraints in spreadsheets and tabular data, Machine Learning 106 (2017) 1441–1468.
[16] S. Paramonov, S. Kolb, T. Guns, L. De Raedt, Tacle: Learning constraints in tabular data, in: CIKM, ACM, 2017, pp. 2511–2514.
[17] Z. Abedjan, L. Golab, F. Naumann, T. Papenbrock, Data profiling, Synthesis Lectures on Data Management 10 (2018) 1–154.
[18] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, R. C. Williamson, Estimating the support of a high-dimensional distribution, Neural Computation 13 (2001) 1443–1471.
[19] H. Yang, I. King, M. R. Lyu, Multi-task learning for one-class classification, in: IJCNN, IEEE, 2010, pp. 1–8.
[20] V. Vapnik, The nature of statistical learning theory, Springer, 2000.
[21] J. R. Quinlan, C4.5: Programs for machine learning, Morgan Kaufmann, San Mateo, 1993.
[22] Y. Ren, L. Zhang, P. N. Suganthan, Ensemble classification and regression-recent developments, applications and future directions, IEEE Computational Intelligence Magazine 11 (2016) 41–53.
[23] R. Bellman, A new type of approximation leading to reduction of dimensionality in control processes, Journal of Mathematical Analysis and Applications 27 (1969) 454–459.
[24] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Symposium on mathematical statistics and probability, volume 1, Oakland, 1967, pp. 281–297.
[25] G. Williams, Evolutionary hot spots data mining, in: PAKDD, Springer, Beijing, 1999, pp. 184–193.
[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, Scikit-learn: Machine learning in python, Journal of Machine Learning Research 12 (2011) 2825–2830.
[27] A. Reid, R. Davies, E. Garrett, Nineteenth-century scottish demography from linked censuses and civil registers: A 'sets of related individuals' approach, History and Computing 14 (2006) 61–86.
[28] S. Kumar, W. L. Hamilton, J. Leskovec, D. Jurafsky, Community interaction and conflict on the web, in: WWW, Lyon, 2018, pp. 933–943.
[29] Y. Zoabi, S. Deri-Rozov, N. Shomron, Machine learning-based prediction of covid-19 diagnosis based on symptoms, npj Digital Medicine 4 (2021) 1–5.
[30] P. C. Arocena, B. Glavic, G. Mecca, R. J. Miller, P. Papotti, D. Santoro, Messing up with bart: error generation for evaluating data-cleaning algorithms, VLDB Endowment 9 (2015) 36–47.
[31] X. Li, X. L. Dong, K. Lyons, W. Meng, D. Srivastava, Truth finding on the deep web: Is the problem solved?, arXiv preprint arXiv:1503.00303 (2015).