Anomaly Explanation : A Review

Véronne Yepmo, Grégory Smits, Olivier Pivert

Anomaly Explanation: A Review

Véronne Yepmo Tchaghe[a,*], Grégory Smits[a,*], Olivier Pivert[a,*]

[a] *Université Rennes 1, IRISA - UMR 6704, F-22305 Lannion, France*

Abstract

Anomaly detection has been studied intensively by the data mining community for several years. As a result, many methods to detect anomalies have emerged, and others are still under development. But during the recent years, anomaly detection, just like a lot of machine learning tasks, is facing a wall. This wall, erected by the lack of trust of the final users, has slowed down the usage of these algorithms in the real-world situations for which they are designed. Having the best empirical accuracy is not enough anymore; there is a need for algorithms to explain their outputs to the users in order to increase their trust. Consequently, a new expression has emerged recently: eXplainable Artificial Intelligence (XAI). This expression, which gathers all the methods that provide explanations to the output of algorithms has gained popularity, especially with the outbreak of deep learning. A lot of work has been devoted to anomaly detection in the literature, but not as much to anomaly explanation. There is so much work on anomaly detection that several reviews can be found on the topic. In contrast, we were not able to find a survey on anomaly explanation in particular, while there are a lot of surveys on XAI in general or on XAI for neural networks for example. With this paper, we want to provide a comprehensive review of the anomaly explanation field. After a brief recall of some important anomaly detection algorithms, the anomaly explanation methods that we discovered in the literature will be classified according to a taxonomy that we define. This taxonomy stems from an analysis of what is really important when trying to explain anomalies.

Keywords: Anomaly explanation, Anomaly detection, Outlier interpretation, Interpretability, Explainable Artificial Intelligence (XAI)

1. Introduction

What is anomaly detection? To provide an answer to this question, let us observe Figure 1 below that shows a 2D dataset:

*Corresponding author

Email addresses: veronne.yepmo-tchaghe@irisa.fr (Véronne Yepmo Tchaghe), gregory.smits@irisa.fr (Grégory Smits), olivier.pivert@irisa.fr (Olivier Pivert)

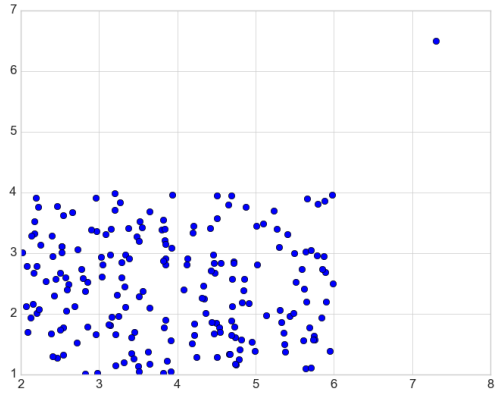


Figure 1: A dataset

Without needing to take a closer look at the picture, one data point catches our attention: the data point located at (7.3 , 6.5), as it is detached from the others. The first thing that comes to mind when seeing this picture is *"This data point is really different from the others, this is not normal. There must be an error somewhere, it is not supposed to be there"*. That data point is called an **outlier**. More formally, an outlier can be defined as an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism [1]. Outliers are also called abnormal data points, irregularities or anomalies, in contrast to normal data points or regularities which seem to follow the same distribution. **Outlier detection** is the task aiming at discovering those deviating data points automatically. Its applications are numerous:

- one classical example is spam detection where a mail server has to identify if an incoming e-mail is a spam (undesirable e-mail) or not, in order to put it into the spam folder;
- in the banking domain, fraudulent credit card transactions are anomalies as they are not performed by the owner of the card. Identifying those is of great benefit for the bank and the cardholder. At this stage, we have to make a difference between **outliers** which are just deviating data points (like if the owner of the card makes a really high punctual payment in comparison to his habits) or noise -that can be a negative value for an amount- due for example to an error in the system, and **anomalies** which effectively reflect the fact that someone else used the card to make a payment. We will say a bit more about that in the next paragraphs;
- unusual behaviours in networks traffic must be identified to fight against attacks that can compromise a system;

- in High Performance Computing (HPC) architectures, or more generally in engineering systems, sensors are used to collect information about different components of the system. Analyzing the records of these sensors, usually in real-time, can help identify faulty behaviours of some components, and correct them afterwards. For example, a very high temperature of a component could indicate that the cooling system is not working correctly;

- in medicine, MRI photographs can be processed to identify cancerous cells;

- in astronomy, images provided by telescopes are studied by machines to detect the apparition of new celestial objects. In this field, the expression *novelty detection* is often used to refer to the identification of new outliers;

- in international trade, the prices tagged on the invoices of some transactions can be abnormal (lower or higher than the real price of the product involved in the transaction). This so-called **trade misinvoicing** is illegal, and the money obtained from it usually finances terrorism and corruption. In addition to that, it causes huge losses to the countries involved [2].

The notion of anomaly can be ambiguous and heavily dependent on the context, especially as even the notion of regularity can be ambiguous. For example, in HPC there may be times when the system is under heavy strain (we will call these *-1-*). The CPU will work more during those times and that will cause an increase of its temperature, as compared to moments when it is not used much (called *-0-*). In addition to that, we can have times (*-2-*) during which the temperature of the CPU is higher than or close to the temperature during those moments of high demand, indicating that there is really something wrong with the cooling system. Without previous knowledge about the system (to know that sometimes the temperature can be high because the CPU is very much in demand), one can consider that the temperatures collected during *-1-* are abnormal, especially when these periods are few in comparison to *-0-*. Those temperatures in *-1-* would therefore be flagged as anomalous (like the ones in *-2-*, obviously). On the other hand, one can consider that only the situation *-2-* is abnormal, but that requires knowing that the situation *-1-* can happen and is not related to a defect in the system. In this case, like in the bank situation that we explained previously, the temperatures in *-1-* are outliers and the temperatures in *-2-* are anomalies. Nevertheless, since this distinction requires some knowledge about the system (the context), in general in the literature both are referred to and classified as anomalies, and the observations in *-2-* are called *contextual anomalies* in the most stringent papers.

Picture 2 below also illustrates an example of ambiguity when dealing with anomalies. There are 500 round data points, 50 triangles and 25 squares.

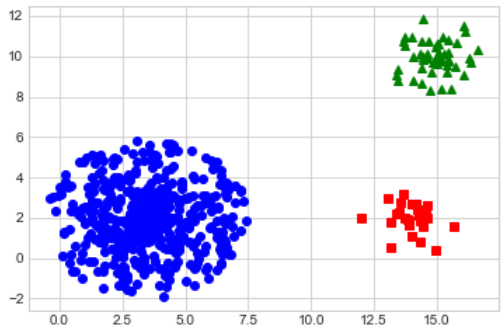


Figure 2: Another dataset

Are the entire cluster of triangles and the entire cluster of squares anomalies? Are anomalies only the data points which are not really close from the centers of the clusters of triangles or squares?
Additional knowledge could be needed to provide answers to these questions with certitude, and as humans we can have access to this knowledge by analyzing the ambiguous data points or by just plotting the dataset like in Figures 1 and 2 (which is not even enough in the case of Figure 2). But the machine does not always have all this knowledge. Furthermore, in the examples above we have 2-dimensional datasets which are easy to visualize. It is hardly the case in real-world situations where datasets may have hundreds of dimensions. Consequently, we could not even be able to provide the additional knowledge needed. An anomaly detection algorithm just tells for each data point if it is abnormal or not, sometimes with a score indicating the degree to which it thinks the data point is anomalous, and that is all. Even us computer scientists, we are, in most cases, not able to explain why the algorithm identified a specific data point as unusual relatively to others. It would not be fair to ask end-users, to whom the anomaly detection system looks like a black-box, to blindly trust its output, especially when the system is used in sensitive domains like medicine. If in addition to the anomaly score the machine could at least provide explanations on why it flagged a data point as anomalous, a user could know without much additional effort if that anomaly is relevant in the context or not. Plus, explanations could improve the trust (and consequently the usage of the system) of the users towards the system as the latter would not be an opaque box anymore.

This work does not intend to make an extensive review of the anomaly detection field or an extensive review of the XAI field. For anomaly detection, many detailed surveys exist ([3, 4, 5, 6]), and sometimes they are specific to a field like in [7]. In the case of explanations, with the outbreak of XAI, there has been a surge of general reviews these last years ([8, 9]); but to the best of our knowledge there is none devoted to the anomaly explanation methods.

When explaining the results of a classification task, we are interested in telling why the explained instance is similar to the other instances belonging to the same class. Indeed, classes share common properties that we want to identify in order to explain an instance of the class. In contrast, since anomalies are irregular and diverse, we are interested in telling how they differ from the regular instances. From this perspective, there are many types of explanations that we can provide, and they will be enumerated and studied. This work will emphasize on anomaly explanation methods. Its main contribution is a taxonomy of the existing anomaly explanation methods and the limits of each identified category. This will be displayed in section 3. But before that, a non exhaustive summary of the principal anomaly detection methods will be made in the next section.

2. Anomaly Detection

There is no unified taxonomy of anomaly detection methods in the literature. At a low level of granularity, a distinction can be made between algorithms based on the labelling of the dataset. Supervised anomaly detection methods use a labelled dataset during training, and identifying anomalies is therefore a binary classification task in which there is high imbalance in the dataset, since anomalies are few in comparison to regular data. When labels are absent, anomaly detection is performed in an unsupervised manner: there is no training, the data are just fed to the algorithm which identifies the outliers. The latter is more convenient since labelling a dataset is a daunting task and it can be difficult to have access to already labelled data. Plus, all the anomalies may not be known before building the algorithm: new anomalies different from all the previous ones can appear and should be correctly identified as anomalies. Between the supervised and unsupervised settings, authors sometimes add in their reviews the semi-supervised setting in which only regular instances are used during the training. In that case, a model of the normal instances is learned and outliers are the instances which do not fit the model. But since outliers are few in the data set, and nowadays most of these methods are robust enough to provide good results even with the presence of outliers in the training set, we will classify them into the unsupervised methods. Ultimately, what we will include in the set of unsupervised anomaly detection methods in this work are the ones which do not require training (because they are completely unsupervised) and the ones requiring training, but robust enough to not be perturbed by the presence of anomalies in the training set. The unsupervised setting is the most realistic one when dealing with anomaly detection for the reasons we mentioned earlier, we will therefore focus on it.

There is no unified taxonomy for unsupervised anomaly detection either. In [5] for example, the authors make a distinction between nearest-neighbor-based, clustering-based, statistical, subspace-based and classifier-based methods; whereas in [10] the authors consider three sets of methods: density-based, distance-based and model-based. From our perspective, nearest-neighbor-based, distance-based and density-based methods can be put together, since distances are computed to evaluate densities and they all rely on distances computations.

Clustering-based methods do not belong to the previous group, because in contrast to the previous methods there is an explicit notion of clusters, even though distances between data points are still computed. Model-based methods should be a distinct category to group robust semi-supervised methods and methods for which a model of the data points is learned. The clustering-based methods belong to this category, since a clustering is a model of the data set. We add to the two previous groups the neural-network-based methods containing all the deep learning anomaly detection algorithms.

Basically, we propose to divide the anomaly detection methods into three groups: distance-based methods, model-based methods and neural-network-based methods. As mentioned, this paper does not intend to provide an exhaustive review of the existing algorithms. The focus will be on the most promising/used ones of each category.

2.1. Distance-Based Methods

In this category there are all the methods relying on distance computation to identify anomalies. This distance computation can be used, for example, to compute densities and flag as outliers data points which are located in low density regions.

Local Outlier Factor (LOF) [11] compares the density of a data point to the density of its k nearest neighbors, with the hypothesis that for an inlier those two quantities will be approximately the same. The density of a data point x in this context is the inverse of the average (on the neighbors of x) of the maximum distance among the distance between x and its neighbor and the distance from that neighbor to its farthest neighbor. This local treatment is efficient in scenarios where there are clusters of different densities in the data set: even for sparse clusters the data points which are deep inside the cluster will have approximately the same density as their closest neighbors. As a result, their LOF will be close to 1. The LOF of a data point x is given by:

$$LOF_k(x) = \frac{\sum_{y \in N_k(x)} \frac{l_k(y)}{l_k(x)}}{|N_k(x)|}, \tag{1}$$

where $N_k(x)$ is the set of k-nearest neighbors of x and $l_k(x)$ is the *local reachability density of* x defined by:

$$l_k(x) = \frac{|N_k(x)|}{\sum_{y \in N_k(x)} max(d(y,x), d_k(y))}. \tag{2}$$

In Equation 2, $d_k(x)$ is the distance D such that there is at least k data points y for which $d(y,x) \leq D$ and there is at most $k-1$ data points z for which $d(z,x) < D$. In other words, it is the distance between x and its $k^{th}-$nearest neighbor.

The LOF of an outlier does not have a specific range of values, but it is bounded. The formulas to compute the bounds are given in [11]. The incidence of k on the LOFs of the data points is not clear. Increasing (resp. decreasing)

the value of k does not always increase (resp. decrease) the values of the LOF. As a result, the authors of [11] propose a method to determine a range for the values of k. For the lower bound of k, even though they specify that the value could be application-dependent, they state that picking 10 to 20 works well in general. Finally, the authors suggest to compute the LOFs of the data points for the different values of k in the range found and to take aggregates like the maximum, the minimum or the mean to find the final values of the LOFs. However, taking the minimum may erase the outlying nature of a data point completely and taking the mean may dilute the outlying nature of a data point [11], and that is why they used the maximum in their experiments. Because LOF uses the Euclidian distance to select the nearest neighbors of a data point, its density estimation can be incorrect when features have a linear correlation as highlighted in [5]. To solve that issue, Connectivity-based Outlier Factor (COF) was introduced in [12]. COF uses the *chaining distance* instead of the Euclidian distance and performs in a similar way to LOF for the computation of the outlier scores. Other variants of LOF have been proposed in the literature, and they are presented extensively in [5].

2.2. Model-Based Methods

The idea behind clustering methods for anomaly detection is to cluster the data set and then to flag as anomalies data points which do not belong to any cluster. For that purpose, in an unsupervised setting, the clustering method must be robust enough to not be sensitive to the presence of outliers in the data set: sensitive methods will try as much as possible to insert the outliers into clusters, which can lead those instances to be flagged as normal instances, or will simply throw away these outliers. Robust methods, like *FindOut* [13], do not force the outliers into clusters. An evident drawback of this conception is that if there are clusters of anomalies in the data set they will be considered as regular instances. This problem can be solved by a post-inspection of the clusters: dense large clusters are considered normal and sparse or small clusters are considered anomalous. In [14], for example, the authors use the k-means clustering algorithm to cluster a data set containing network traffic information. Then, an identification of the normal and anomalous clusters is made, and data points which do not belong to any cluster are flagged as normal or outliers depending on the type (regular or anomalous) of the cluster they are closer to. But if the instance is located at a distance greater than a predefined threshold from a normal cluster, it is classified as anomalous.

After projecting the data in a higher dimensional space using a *kernel*, One-Class Support Vector Machines (One-Class SVMs), which were first introduced in [15], try to draw a boundary around the data instances by solving an optimization problem. A decision function is then extracted from this boundary. The value of the function will be $+1$ for the data points inside the region delimited by the boundary, and -1 for the others. From this description, it is obvious that One-Class SVMs are a semi-supervised outlier detection method, as a model of the normal points is learned. But because One-Class SVMs as described in [16] are robust enough so that they can deal with the presence of

anomalies in the training data, they are considered unsupervised and outlined in this work. In [16], the authors propose two enhanced versions of One-Class SVMs, namely Robust One-Class SVMs and η One-Class SVMs to deal with outlier detection in a completely unsupervised way. The two enhancements are similar to the classical One-Class SVMs, except that there is an explicit asumption that outliers are present in the data. For Robust one-class SVMs, slack variables already present in the classical One-Class SVMs optimization objective are modified to take into account outliers. For η One-Class SVMs, there is an outlier suppression mechanism through a variable η which represents the normality of a data point. For both methods, an outlier score based on the distance of the data point to the decision boundary is computed. Normal data points have a score between 0 and 1, and, the more outlying a data point, the larger its score.

Isolation Forest (IF) [10] is based on the idea that outliers are isolated in the feature space. Starting from a random sample of the dataset, the method selects randomly one attribute a among the set of attributes A, then selects randomly a split value v in the attribute range. The sample is then partitioned into two subsets according to that split value: the data points for which the value of a is less than v and the data points for which the value of a is greater than or equal to v. This process is repeated recursively on each partition and a binary tree is obtained. Each node of the tree is a splitting step and, consequently, each node has two children representing the two subsets obtained after the split. The tree building process will stop for a node when no partition can be made anymore (when the size of the sample in the node is 1) or when a predefined depth threshold h_{lim} is reached. A set of t trees is generated this way with different random samples in order to obtain a forest. After building the forest, the anomaly score of a data point is computed using the average depth of the data point in the trees of the forest:

$$s(x) = 2^{-\frac{E(h(x))}{c(\Psi)}}, \tag{3}$$

where $E(h(x))$ is the average depth of the data point over the t trees. $c(\Psi)$ is a normalization factor corresponding to the average path length of unsuccessful searches in a binary tree with Ψ nodes. If the average depth is equal to $c(\Psi)$, meaning that in every tree the search of the data point was unsuccessful -because h_{lim} was reached-, then the anomaly score will be 0.5 which is consistent with the fact that we are not sure about whether the data point is abnormal or not. The data point will be categorized as an anomaly if its anomaly score is greater than a predefined threshold ϵ ($\epsilon = 0.5$ by default). Some limits of the Isolation Forest have been highlighted, which has lead to some improvements of the method. One limit, displayed in [17], is the inconsistency of the anomaly scores produced by IF in some situations, inconsistency towards the distribution of the data points. To solve that problem, the authors of [17] proposed a variant of the Isolation Forest called Extended Isolation Forest which uses hyperplanes with random slopes instead of lines parallel to the axes during the construction of the trees. Just like in the classical Isolation Forest where two split parameters are stored

(the feature and the split value), two parameters are also stored in the Extended Isolation Forest: the slope and the intercept. This idea of using hyperplanes was already proposed by SCiForest [18], with a deterministic selection of the split points in order to detect local clustered anomalies.

2.3. Neural Networks Based Methods

One of the earliest works about outlier detection with deep learning is [19]. In the latter, the authors use a Replicator Neural Network (RNN) with three hidden layers to perform outlier detection.

AutoEncoders (AEs), which have been previously used for dimensionality reduction and have a similar structure to RNNs are also exploited for outlier detection. Just like a Replicator Neural Network, the autoencoder receives as input a data point and tries to reconstruct it. First, a set of layers called the *encoder* transforms the input into another data point with less features in the space known as *latent space*. After that, another set of layers called the *decoder* tries to transform the lower dimensional data into the original input. During the training, the neural network will try to minimize the reconstruction error which is the difference between the output x' and the input x. With a perfect autoencoder the output is always the original data point ($x' = x$), and the reconstruction error is therefore zero. To obtain the lower dimensional data points we just have to get the corresponding data points in the latent space (after the encoding step). Figure 3 below shows an example of autoencoder:

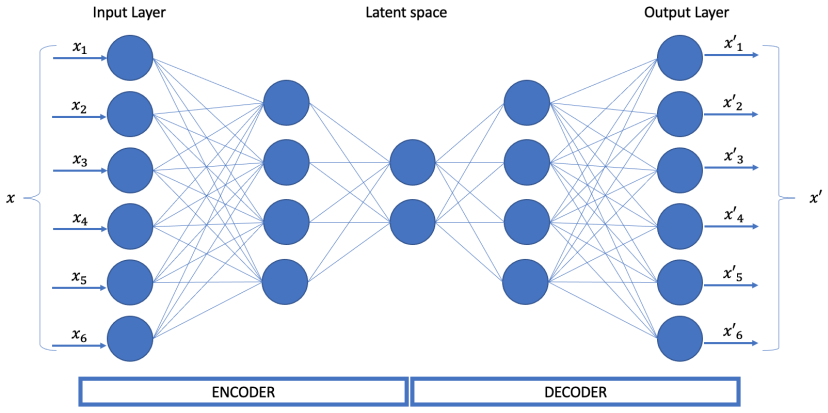


Figure 3: Example of autoencoder: the input space has 6 dimensions and the latent space has 2 dimensions.

The usage of autoencoders for outlier detection assumes that outliers will always have a higher reconstruction error than normal data points, and this reconstruction error therefore constitutes a measure of outlierness. This assumption is justified by the fact that the autoencoder learns a model of normality: it should be able to reconstruct perfectly the regular instances, and outliers which

deviates from regular data points should be reconstructed poorly. In theory, an autoencoder is a semi-supervised anomaly detection method as it should be trained only with regular instances so that the outliers can be easily detected with their high reconstruction error later. In [20] for example, the authors used an autoencoder for anomaly detection in a semi-supervised way. But, with more robust architectures taking into account the presence of outliers in the training data, autoencoders can be classified in the unsupervised approaches for anomaly detection. In [21], an ensemble of autoencoders, each with different random connections between the layers is used for anomaly detection. Each autoencoder of the ensemble is trained on a different sample of the dataset. Finally, the median score over the ensemble is used as the final anomaly score for an instance. An ensemble of autoencoders is also used in [22]. But here each autoencoder of the ensemble performs anomaly detection on different features of the feature space. Autoencoders are not the only dimensionality reduction algorithms used for outlier detection.

Using an ensemble of autoencoders is not the only way to make the method robust enough to be used in an unsupervised way; autoencoder variants like Variational AutoEncoders (VAEs) can be used. In [23] for example, the authors used a VAE to detect anomalies in network traffic. VAEs are similar to autoencoders, but instead of finding a lower dimensional representation of the input in the latent space, they try to find the distribution from which the input has been generated. It means that during the encoding step, the VAE will find the parameters of the distribution that generated the input, and during the decoding step, the VAE will sample a data point from the distribution found during the encoding step, and decode it. The goal here is not only to minimize the reconstruction error between the output and the input, but also to make the computed distributions close to the standard normal distribution. VAEs are generative neural networks. Other types of generative neural networks like Generative Adversarial Networks (GANs) are also used for anomaly detection. A GAN consists of two components: a generator and a discriminator. The generator tries to generate instances which are close to real instances (trying to learn the distribution of the data points) and the discriminator tries to make the distinction between real instances and fake instances produced by the generator. The generator and the discriminator are opposed to each other; the generator wants to generate realistic "fake" instances that will fool the discriminator. A GAN was used in [24] in combination with an AE to detect anomalies in medical images in a semi-supervised way. Another AE variant, an Adversarial AutoEncoder (AAE) was used in [25] to detect anomalies in wireless spectra. An Adversarial AutoEncoder is a mix of a classical autoencoder and a GAN: the autoencoder still tries to reconstruct the instances, the generator generates instances that seem to come from the latent space of the autoencoder; finally the discriminator of the GAN has to find out if the instance that it faces comes from the latent space of the autoencoder or if it has been generated by the generator.

The topic of deep anomaly detection is really wide and covering it entirely is beyond the scope of this work. More detailed surveys can be found in [26]

and [6].

2.4. Discussion

Neural-network-based methods are suitable for the identification of anomalies in complex data types like images and sequence data (time series, videos, audios, text...), as neural networks are able to capture complex relationships in the data. For more classical data like tabular data, the time and effort put in the training of a neural network can be discouraging: it requires a large training dataset, more parameters need to be set in comparison with other methods and the training time is not insignificant, although there are specialized ecosystems for neural networks training nowadays. Distance-based methods rely on distance computations which are time consuming. Even when the distances are only computed between neighboring data points, the neighbors have to be identified and the size of the neighborhood is generally a crucial parameter of the method. In addition to that, in high dimensional spaces distance-based methods suffer from the curse of dimensionality. Plus, if we don't have tabular data, the classical Euclidian distance may not be suitable anymore, and an appropriate distance metric should be selected or developed, which is not always an easy task. On the other hand, distance-based methods do not require any training and local techniques, like LOF, are able to discover local outliers (which are outliers relatively to a small subset of data points, in opposition to global outliers which are outlying relatively to all the other data points). The choice of the number of clusters in clustering-based methods is of paramount importance and there are few methods robust enough to deal with the presence of outliers in the dataset, which is not surprising since the main focus of clustering is to capture regularities. But clustering methods are able to discover different types of anomalies in the data, which is an advantage. The Isolation Forest algorithm requires no distance computation and has very few hyper-parameters, which makes it very appealing for anomaly detection. Plus, it has been proved to be very effective and was specifically thought for outlier detection in contrast to other methods which where originally designed for other purposes. But it is not the best choice when there are categorical attributes because categorical attributes are not ordered.

In terms of output, distance-based and model-based methods return a score representing how sparse is the space surrounding the data point. Neural-network-based methods in contrast return a score which represents the quality of the reconstruction of an instance by the network.

Finally, as being said earlier, all the methods listed in this part only return anomaly scores. There is no clue on why a data point is an outlier based on its characteristics. In the next section, a complementary issue will be explored: anomaly explanation.

3. Anomaly Explanation

For an algorithm which aims at recognizing in a set of images which ones are cat images and which ones are dog images, the most natural way to tell users

why the algorithm tagged a picture as a cat instead of a dog is to return the group of pixels that helped the algorithm to make the difference. This group of pixels can represent the whiskers of the cat on each image for example. In this way, the user will notice that the whiskers are an attribute that the cat possesses, and not the dog, and will therefore understand why the algorithm decided that it is a cat picture. In general, identifying the features/attributes which contributed the most to the decision of an algorithm is a good start and a classical method to provide explanations. Anomaly detection is also concerned. In Figure 4 below, to mark the square data point as anomalous, we can look only at the feature f_1 for all the instances: in comparison to the regular data points in blue for which the values of the attribute f_1 vary between -1 and 8, it takes the value 12.

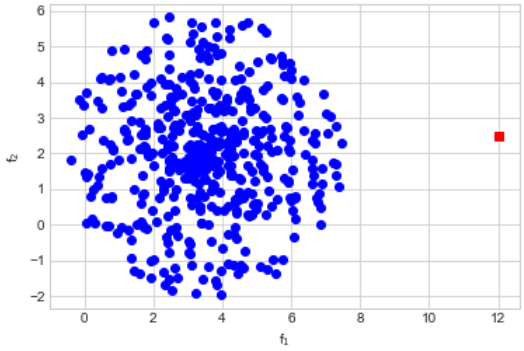


Figure 4: Anomaly explanation by feature importance: attribute f_1 helps us to tag the square data point as anomalous

The same cannot be told for the feature f_2 since the square instance has a value of 2.5 for that attribute, which is normal when comparing it with the values of f_2 the regular instances. Consequently, to explain that anomaly to the user, we can just say that attribute f_1 contributed to the abnormality of the square data point. This first category of anomaly explanation is **feature importance**.

But just telling which features are important is sometimes not enough. In Figure 5 below, when trying to explain the abnormality of the square data point using feature importance, we will observe that both features have equal importance, because one attribute does not help the algorithm to identify the anomaly more than the other: the isolated instance has a regular value for each of the features taken independently. It is the combination of the values for both attributes which makes the data point irregular. On the other hand, in Figure 1, the outlying instance has an abnormal value for both features. In this two cases, explanation by feature importance will just return the two attributes, and that is no information at all. In two dimensions, like in our examples, it is easy for the user to plot and observe. But, again, if we are in higher dimension,

which is almost always the case, displaying a list of features with more than two having the same importance is not really helping the user.

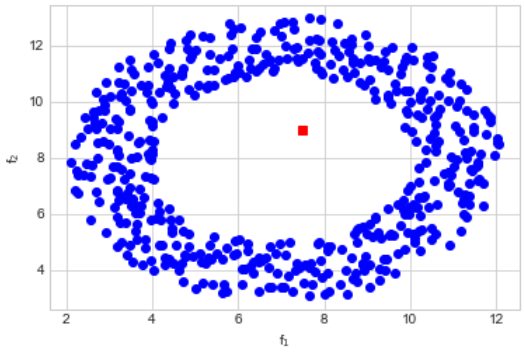


Figure 5: Anomaly explanation by feature values: the square data point is anomalous because $f_1 = 7.5$ and $f_2 = 9$, and that combination of values is not normal for an instance.

In these situations, it would have been more understandable to say, for instance, that the data point in Figure 5 is anomalous because it has a value for the feature f_1 around 7.5 and a value for the feature f_2 around 9. For Figure 1, it would be better to say that the data point is irregular because the first attribute has a value greater than 6 and the second attribute has a value greater than 4. This second category of explanation is the **anomaly explanation by feature values**.

Again, when the number of features involved in the explanation is increasing, it is difficult to use this kind of explanations because we can have several conditions on the features. In addition to that, with the two previous categories of explanations, we just have information about the anomaly. We do not know concretely what is the difference between anomalies and regular data points. With the example in Figure 5, after discovering that the instance is anomalous because $f_1 = 7.5$ and $f_2 = 9$, the user can ask if a data point with $f_1 = 8$ and $f_2 = 7$ would be anomalous (without plotting the data set of course). Explanations by feature importance and by features values do not provide an answer to this question. An answer would be provided if the anomaly was explained by directly comparing it to regular data points. This has been done since the beginning of this section with figures, but visually: from them, one can directly spot the irregular data point because there is a visual comparison with regularities. This third category of explanations will be called **anomaly explanation by data points comparison**.

Data points comparison provides richer explanations to anomalies. But there can be different types of anomalies in the data set, with each type sharing some characteristics, like different fraud profiles in credit card fraud detection. In this case, we must tell why an instance is abnormal and why it is different from other abnormal instances. In addition to that, if there are different clusters of regular

data points in the data set, and each of these clusters has some anomalies as shown in Figure 6 below where there are 3 clusters and 4 anomalies (x_1, x_2, y and z), the most complete explanation that we can provide is telling that x_1 and x_2 are anomalies for the cluster of round instances and why it is the case, that y is an anomaly for the the triangles and why, and finally that z is an anomaly for the squares and why.

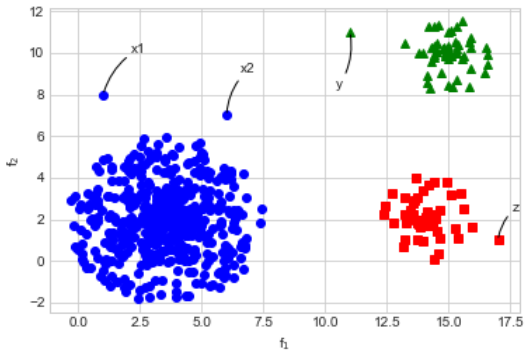


Figure 6: Anomaly explanation by structure analysis

To provide this kind of detailed explanations, an analysis of the intrinsic structure of the data set is required, followed by a comparison of the anomaly(ies) with this intrinsic structure. This last category of explanations will be called **explanation by structure analysis**. It starts at the anomaly detection level by identifying groups of anomalies or anomalies with respect to different groups of regular data points.

To sum up, we propose the following four categories for anomaly explanation methods:

- explanation by feature importance,
- explanation by feature values,
- explanation by data points comparison,
- explanation by structure analysis.

In the next subsections we will show that existing works can be inserted into the categories of this taxonomy.

Noteworthy is that although we were not able to find a survey devoted to anomaly explanation methods, in some papers dealing with the topic the authors tried to categorize anomaly explanation methods. The most recurrent taxonomies are model-specific vs model-agnostic methods, and local vs global methods. Model-specific methods are the ones built for a particular machine

learning algorithm, while model-agnostic methods can be used with any algorithm. Local methods explain why a specific data point is anomalous while global methods explain why anomalies are irregular globally, or why a group of anomalies are irregular. These taxonomies are really coarse, and that is why we provide a more refined one and analyse the appropriate use of each element. However, because they provide additional information about anomaly explanation methods, when listing the works of the literature we will also insert each of them into these categories.

To illustrate this section, we will consider the following example: in Table 1, we have a list of products along with their brand, model, unit weight and unit price. We want to identify the anomalous products, using the information in Table 2. The latter correspond to the real properties of the products.

Table 1: List of products

ID	**Brand**	**Model**	**Unit weight(g)**	**Unit price(USD)**
1	Apple	iPhone X	174	550
2	Apple	iPhone 11	194	600
3	Apple	iPhone 12	300	500
4	Samsung	Galaxy S20	163	850
5	Samsung	Galaxy S21	169	900
6	Samsung	Galaxy Note 20	250	900
7	Xiaomi	MI 11	100	500
8	Xiaomi	MI 10S	208	300
9	Xiaomi	POCO F2 Pro	260	800

Table 2: True characteristics of the products

Brand	**Model**	**Unit weight(g)**	**Unit price range(USD)**
Apple	iPhone X	174	[500-600]
Apple	iPhone 11	194	[800-1000]
Apple	iPhone 12	164	[1100-1500]
Samsung	Galaxy S20	163	[800-900]
Samsung	Galaxy S21	169	[900-1200]
Samsung	Galaxy Note 20	192	[550-700]
Xiaomi	MI 11	196	[450-600]
Xiaomi	MI 10S	208	[100-350]
Xiaomi	POCO F2 Pro	210	[200-300]

From the two tables above, it can be seen that the anomalies are:

- the products *2* because of its low price,
- the product *3* because of its high weight and its low price,

- the products *6* and *9* because of their high weight and their high price,
- and the product *7* because of its low weight.

3.1. Anomaly Explanation By Feature Importance

In this part, we will make a distinction between the methods which identify the important features without other explanations, and the methods which weight the features or provide an ordering of the features according to their importance.

3.1.1. Non-weighted feature importance

The earliest work on anomaly explanation is a non-weighted feature importance approach. In [27], the authors identify outliers in subspaces of the features space using a distance-based anomaly detection method. In our example, the outlier *2* can be identified in the subspace $(model, unitprice)$. This serves as explanation since the identified anomalies are outliers in the specific subspaces found, meaning that the features constituting the subspace are those that discriminate the most the instance. The authors introduce the notions of *strongest*, *weak* and *trivial* outliers. An outlier is non-trivial in a subspace A if it is not an outlier in any subspace included in A. A strongest outlier is an outlier in a strongest outlying feature space (if no outlier exists in any subspace included in A, then A is a strongest feature space). A weak outlier is a non-trivial not strongest outlier. Algorithms are provided to identify (and thus explain) strong and weak outliers. This anomaly explanation method is model-specific because it is designed for distance-based methods. It is also local because it helps explaining one outlier at a time.

Like the work in [27], some methods also explain anomalies by finding the set of features that isolates them. In [28], the authors explain a given anomaly by identifying the subspace of features that best separates that outlier from the rest of the dataset. More generally, anomaly detection methods which identify outliers in subspaces of the original feature space like Subspace Outlier Degree (SOD) [29], or in subspaces of a transformation of the original feature space like Correlation Outlier Probability (COP) [30] can be considered as anomaly explanation methods using feature importance. Indeed, the features in the subspaces obtained are the most important for the identification of the anomaly. These methods do not quantify the importance of each feature and are thus non-weighted feature importance anomaly explanation methods.

The authors of [31] use focus plots to explain a group of outliers. Focus plots are 2-dimensional feature plots. The explanation algorithm tries to find the set of features pairs that best discriminate the outliers in the group. All possible combinations of pairwise plots are generated, and, for each pair of features the outlier scores of the data points in the group are computed using only the two features in the pair. The pair that gives the highest anomaly score is kept. Some heuristics are used to limit the search in the features space. This method named *LookOut* is model-agnostic. But, as highlighted in [32], outliers can be diverse, and trying to explain a set of random outliers using *LookOut*

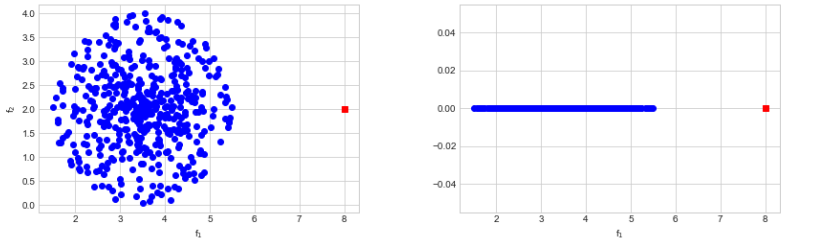
is not efficient as the algorithm will try to make a compromise between the outliers to produce the final focus plots. The latter may therefore not include the best focus plot for each outlier individually. For example, the best focus plot for outlier *2* is $(model, unitprice)$ and the best focus plot for outlier *3* is $(unitweight, unitprice)$. If we want to explain these two outliers using *LookOut*, the method may select the first focus plot, which is not optimal for outlier *3*. As a result, the authors of [32] proposed a method to explain clusters of outliers, clusters based on the behaviour of the outliers, instead of random groups of outliers: the outliers are clustered according to the features that separate the most each of them from the other data points, and finally the features pairs which discriminate the most a cluster of outliers from the other instances are returned. It is also possible for the final user to retrieve the features pairs that best discriminate all the outliers of the dataset.

More generally, there is a set of datamining methods called Group Outlying Aspects Mining (GOAM) which try to identify the features which make a certain group of instances distinct from the other instances. In this case the instances do not have to be outliers; they could be regular data points and the user just wishes to know with which combination of features they are the most distinct from the others. For more details, the reader can refer to [33].

In [34], the authors explain anomalies in images using metadata. Anomaly detection is first performed using Principal Component Analysis (PCA). PCA is a dimensionality reduction method projecting the data points into a lower dimensional space that maximizes the variance between the data points. Each feature in the computed space is a linear combination of the features of the original space. PCA is also used for anomaly detection with the assumption that outliers will be separated from the other instances in the computed space. After the identification of anomalies, tags are generated for each picture in the data set; every tag is a word describing the picture, and these tags constitute its metadata. Then, the tags corresponding to the greatest number of anomalies are identified and returned as global explanations of anomalies. The identification of important tags, importance with regard to anomaly detection, is made using algorithms like PRIM (Patient Rule Induction Method) whose objective is to find regions in high-dimensional input space with large values of a real output variable [35]. This explanation can be used with any anomaly detection algorithm; it is therefore a model-agnostic method. It is an explanation by feature importance since the features space has just changed from the space of pixels of the images to the space of metadata, but ultimately the most relevant features/metadata are returned.

With Sequential Feature Explanations (SFEs) [36], a sequence of features is presented to a simulated analyst for a specific outlier. It is therefore a local explanation method. If after using only the first feature in the sequence the analyst cannot conclude that the data point is anomalous, it will use the two first features and so on, until the data point is found outlying using a sequence of features. The explanation for the outlier will be the smallest sequence of features that the analyst has used to conclude that the data point is an outlier. SFEs are employed with distance-based anomaly detection methods, more specifically

with density-based methods that estimate a probability density function over the data set. In our example, when trying to explain the outlier *6*, the method can suggest the feature *model* first. It is not enough to conclude that the data point is anomalous. It can then suggest the feature *brand*. It is still not enough to conclude using the two first features. After suggesting the feature *unitweight*, we can conclude that the data point is anomalous using the triplet $(model, brand, unitweight)$. The latter is finally returned as an explanation. With the example in Figure 7a below, (f_1) is returned as an explanation:



(a) A 2-dimensional dataset (b) Dataset from (a) projected on the axis f_1

Figure 7: Sequential Feature Explanations: feature f_1 alone is enough to conclude that the square data point is irregular (b). As a result, the SFE for that anomaly, if f_1 was the first feature presented, is: (f_1).

3.1.2. Weighted feature importance

Local Outlier Detection with Interpretation (LODI) [37] and Local Outliers with Graph Projection (LOGP) [38] identify outliers in subspaces of the original feature space and in subspaces of a transformation of the original feature space respectively, like SOD and COP that were introduced in the previous section. But LODI and LOGP provide weights quantifying the importance of each identified feature.

SHAP (SHapley Additive exPlanations) [39] is a model-agnostic method which explains the prediction of an instance by computing the contribution of each feature to the prediction. It has many variants like Kernel SHAP, Deep SHAP which is a model-specific explanation method tailored for deep neural networks, or Tree SHAP designed for tree models. SHAP values do not only say which features contributed to the anomaly and by how much, but also which features tend to make the instance regular and by how much. As an example, the feature *unitprice* will receive a higher SHAP value than the feature *unitweight* for outlier *9*. They both contribute to making the instance anomalous, but the feature *unitprice* contributes the most because it is further away from the regular values than *unitweight* is, for that instance. The SHAP values of features *brand* and *model* would be approximately the same, as they both make the instance regular and none does it better than the other. In [40], the authors use Kernel SHAP locally to explain anomalies detected by an AutoEncoder: after detecting an anomaly because of its high reconstruction

error, the top features (the features having the highest reconstruction errors for the anomaly) are identified. For each top feature, the SHAP values -which indicate how the prediction of a model's output changes when a feature's value changes- of all the other features are computed. The features are then divided into two groups based on the SHAP values computed: the features contributing to the anomaly (the features pushing the instance towards an anomalous state on the top feature selected) and the features offsetting the anomaly (the features trying to make the value of the top feature selected normal). Finally, for each top feature, the features contributing the most to the anomaly and the features offsetting the most the anomaly are returned. The authors of [41] produce similar explanations to time series anomalies using an extension of Kernel SHAP, the anomalies having been identified by a GRU-AutoEncoder.

SHAP values are based on Shapley values which come from game theory. Shapley values represent the contribution of each feature in the prediction of an instance. They are usually hard to compute, and it is the reason why they are often approximated using SHAP values for example. Shapley values were also exploited for anomaly explanation by feature importance in [42], but using PCA as anomaly detector. In [43], the computation of the shapley values was generalized to provide explanations to any semi-supervised anomaly detector.

DIFFI (Depth-based Feature Importance for the Isolation Forest) [44] is a model-specific method providing explanations to the output of an Isolation Forest. It gives feature importance scores based on the results of the Isolation Forest. According to DIFFI, an important feature should induce the isolation of anomalies at small depth, and should also produce higher imbalance on anomalous data points. After building the Isolation Forest, DIFFI processes each tree separately to assign feature importance scores to each feature for a specific tree and then aggregates the scores to compute the feature importance scores for the whole forest, even if the aggregation formula is not clearly stated. In addition to these feature importance scores, DIFFI also provides local feature importance scores which help identify the features that contributed the most to detecting a specific anomaly. The global scores we described earlier identify the features that contributed the most to isolating the anomalies in the samples that helped building the forest.

Neural-network-based anomaly detection methods possess the advantage that they can leverage explanation methods designed for neural networks:

- in [23], the authors extract the gradients of the features from a trained Variational AutoEncoder to explain why a data point is anomalous. The idea behind is that if a small variation of a feature's value for an outlier causes a huge variation of its anomaly score, then that feature is highly responsible of the outlierness of that instance. It is thus a local, model-specific anomaly explanation method;

- in [45], the authors rewrite One-Class SVMs models in terms of neural networks and then perform anomaly detection using the neural network obtained. To provide explanations to the output of the neural network,

a Layer-wise Relevance Propagation (LRP) with a Deep Taylor Decomposition is used to obtain the most important features for identifying the outliers. It is a local, model-specific explanation method. Layer-wise Relevance Propagation was also used in [46], although anomaly detection was performed in a supervised way using a neural network;

- in [47], attention mechanism is used with LSTM to detect anomalies in system logs. An analysis of the attention weights is performed afterwards in order to identify the most important features for anomaly detection globally.

ACE, *Anomaly Contribution Explainer* [48] is a model-agnostic method close to LIME [49] which explains the prediction of an anomaly detection algorithm by feature importance. To compute the contribution of each feature to the anomaly score of an instance, ACE builds a local linear model around the instance using its neighbors and their anomaly scores as computed by the anomaly detection algorithm.

Discussion

Feature importance is the most used type of anomaly explanation method. Indeed, numerous works belonging to this category were identified in the literature. The output of these techniques can be a list of features (ordered or not) possibly with a weight indicating the importance of the feature, a pair of features or a list of feature pairs, or a plot displaying how the outlier is separated from the others in a features subspace.

Anomaly explanation by feature importance can be used with any anomaly detection method. For distance-based and clustering-based methods, the identification of subspaces that best separates outliers and normal data points is easy. Neural-network-based anomaly detection methods can benefit from the explanation methods designed for neural networks like LRP or local gradients. For other anomaly detection methods, model-agnostic methods like SHAP exist.

Anomaly explanation methods based on feature importance do not only provide information about why a specific data point is anomalous, but they can also give a global understanding of the anomalies by identifying the features that explain a set of anomalies or all the anomalies. It is clear that the set of anomalies to explain should be chosen carefully to avoid conflicts. Furthermore, feature importance can help identify different groups of anomalies, like in [23] where the authors propose a clustering of the anomalies based on the features gradients to identify the types of anomalies present in the data set. But anomaly explanation by feature importance is too coarse. Plus, if the original features are transformed prior to the anomaly detection, feature importance scores will not be meaningful to the final users as they will not recognize the features presented by the explanation system. This transformation can be made using an algorithm like PCA, either to reduce the dimensionality of the dataset or to avoid the leak of sensitive information.

3.2. Anomaly Explanation By Feature Values

All the explanations coming from decision-tree-based anomaly detection algorithms lie in this category. Explanations are in the Disjunctive Normal Form (DNF), and each literal of the DNF is a conjunction of predicates. Each predicate is a condition on the value of a feature which has the form fsv where f is a feature, s is one of the signs $<, \leq, =, >, \geq$ and v is a feature value. As an illustration, an explanation by feature values of outlier *9* can be: $unitweight \geq 210$ and $unitprice \geq 300$.

In [50] the authors use a random forest to identify anomalies in HPC systems. The algorithm identifies the trees which classified the data point as anomalous; then going from the leaves to the root of each tree, it finds the conditions which helped to flag the data point as anomalous. The conditions regarding the same feature are consolidated afterwards, in order to have the fewest possible number of predicates. Those conditions are then displayed to a human analyst who identifies the most relevant ones and can throw out the least interesting in order to prune the decision trees, so that only relevant anomalies could be identified later.

In [51], after using One-Class SVMs to detect outliers, the space containing the **inliers** is divided into hyper-cubes recursively using a clustering algorithm (k-means++ in this case) until there is no outlier in any hyper-cube; then rules are extracted from the boundaries of each hyper-cube. Each rule is a conjunction of predicates specifying the condition of belonging to one hyper-cube and thus, being a regular data point. Finally the list of rules is returned. It is important to note that although the proposed method has been applied on One-Class SVMs, it is a model-agnostic method as it could be used with any outlier detection algorithm. Figure 8 provides an illustration of the method:

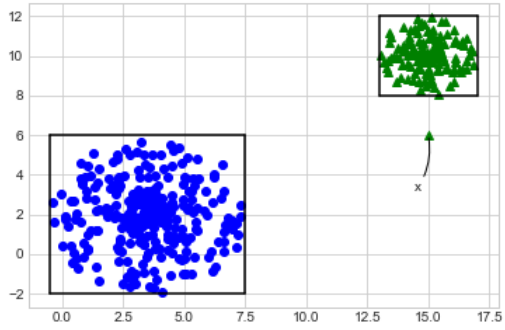


Figure 8: Rules extraction according to [51]: the space containing the inliers is divided into two hyper-cubes (rectangles in this case, because the space has two dimensions). The explanation returned will be: ($f_1 \leq 7.5$ AND $f_1 \geq -0.5$ AND $f_2 \leq 6$ AND $f_2 \geq -2$) OR ($f_1 \leq 17$ AND $f_1 \geq 13$ AND $f_2 \leq 12$ AND $f_2 \geq 8$). x does not respect the rules: it is an anomaly because of that.

In [52] the authors perform anomaly detection using a LSTM neural network. They then approximate the neural network by a decision tree in order to retrieve the explanations. Approximating a hardly explainable model by another, more easily explainable one is a common practice to provide explanations. The target model is generally a tree-based model because it is easier to extract explanations from such models, and the rules generated are generally more human-interpretable.

The Explainer [53] is a model-agnostic anomaly explanation method. After identifying the anomalies using any anomaly detection algorithm, each outlier is explained by exploiting a random forest composed of decision trees built using that outlier and a subset of regular instances. The authors propose two explanation methods: *minimal explanation* in which only one tree is used to extract the rules and *maximal explanation* in which a set of trees is used. Each decision tree aims at separating the outlier from the regular instances. Decision rules are extracted from each tree of the forest to explain the abnormality of the data point in the form of a conjunction of predicates. For the maximal explanation, the rules for all the trees concerning the outlier are aggregated to obtain one compact DNF. To provide global explanations, the detected anomalies are clustered, then the trees for all the anomalies of a specific cluster are aggregated into one forest and explanations are extracted.

Counterfactual explanations can also be classified among this type of anomaly explanation methods. Counterfactual explanations indicate which features values to change (and how) in order to obtain a different prediction for an instance. For example, a counterfactual explanation of the outlier *2* will indicate that the unit price must be increased by 200 to obtain a regular instance. Counterfactual factual explanations in the context of anomaly detection are explored in [54]. The authors generate counterfactuals with an autoencoder-based anomaly detection.

Discussion

The output of this kind of explanations is typically a set of rules on the features as we stated in the introduction of this section. But it can also be a text in natural language, like in [55] where the authors identify anomalies in time series data using a neural network: anomaly detection is performed in a supervised manner and, when a time series is classified as anomalous, the parts of the time series that contributed to the anomaly are identified; then, these parts are checked against some predefined rules. Those parts are finally compared to some statistics about the time series and textual explanations are generated with the information retrieved (statistical features comparison + rules checking).

Anomaly explanation by feature values is tailored for model-based anomaly detection methods, in particular with tree-based methods as stated at the beginning of the section. In that case, the rules are easily extracted (less easily when there are many trees, but still manageable). For other model-based anomaly detection methods like One-class SVM, it is also possible to extract explanations relying on the values of the features, and it has been done in the literature;

but this requires more work than with tree-based methods. After using a neural network to identify the anomalies, using explanation by features values is very difficult; in the work that was mentioned, the rules extraction was not straightforward.

The rules can easily become unreadable due to their number. As a result, some authors chose to return a short list of rules, each rule having a limited number of predicates. This can be sub-optimal because some less important (but still important) information about why an instance is anomalous can be ignored. Another flaw of this type of explanations is that, unlike feature importance, it is a bit complicated to explain anomalies globally. In addition to that, extracting and consolidating rules is more complex in terms of time processing. However, rules remain the most natural way of explaining anomalies, and translating rules into natural language is relatively easy.

3.3. Anomaly Explanation By Data Points Comparisons

Angle-Based Outlier Detection (ABOD) [56] is an unsupervised anomaly detection method providing explanations. To detect outliers, the algorithm will compare the variance of the angles between data points, with the hypothesis that when an instance is regular, the set of angles between that instance and its neighbors has a high variance because it is surrounded by other instances in many directions. The angles between an outlier and its neighbors will not vary that much because the outlier is positioned outside of some sets of points that are grouped together [56]. To give explanations on why an instance is outlying, ABOD finds its closest instance in the nearest cluster, then computes and returns the difference vector between the two data points. Figure 9 below provides an example. The authors of ABOD do not provide a more detailed justification on the choice of the closest data point, so nothing prevents it from being also an outlier and in this case the explanation will not be correct. In addition to that, as remarked in [57], only the closest neighbor is used for the explanation. The other instances in the dataset could contain more insights on why a given instance is anomalous.

In [58], anomalies in network payloads (data contained in a packet, request or connection) are explained by computing the difference between the vector representing the anomaly and a vector which is the average of the regular instances. The difference vector is then plotted for each feature in order to identifying the anomaly features having a value really far from the average regular data points.

In [59], clustering is used to detect anomalies: after the clustering, the most smallest cluster in terms of cardinality is considered anomalous. Then, the anomalous cluster is compared to the other clusters in terms of features. This comparison is reported to the final user as a text enumerating the features (along with the percentages) on which the clusters are different. A global difference percentage between pairs of clusters is also given. The pairs of clusters which are the most different can also be returned with the features differences percentages.

Kernel-based Supervised Hashing (KSH) [60] constructs a group of hash functions which will map the original data points to lower dimensional expressions in a hash code space. To build the hash functions, KSH uses a labelled

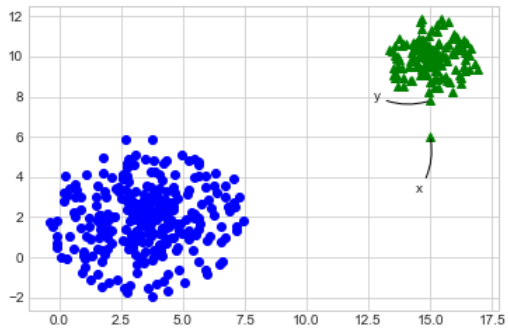


Figure 9: Anomaly explanation using ABOD: $y(15, 7.8)$ is the closest instance to $x(15, 6)$ in the x's nearest cluster. The difference $y - x = (0, 1.8)$ is therefore returned as an explanation for the abnormal data point x.

training set. Data points having the same label will be similar/neighbors in the hashing space. To find out if a given data point is anomalous or not, KSH will search for its (10) nearest neighbors in the hash code space after hashing the data point. The class (anomalous or not) of the instance will be the majority class among its neighbors, and these neighbors are returned as an explanation of the abnormality.

In [61], the authors explain the abnormal value of a feature in the result of an aggregate query on a database by the abnormal value of the same feature in another tuple. An abnormally high number of publications by an author during a year can be explained by the fact that he had an abnormally low number of publications the year before due to rejection, and the publications that were previously rejected were accepted the following year.

In [62], anomaly detection is performed in a semi-supervised way using *GAN omaly* [63] which consists of a Generative Adversarial Network whose generator is an AutoEncoder coupled with an encoder. To provide explanations on why an instance is anomalous, two methods are proposed: display the normal instance closest to the anomaly, or generate a synthetic normal instance that is similar to the anomaly but without the features that make the anomaly outlying. The authors also propose a feature importance anomaly explanation method by inspecting the hidden layers of the GAN to find the most relevant attributes.

Discussion

The possible outputs of anomaly explanation by data points comparisons methods are the closest or the set of closest instances (irregular or not) of an anomaly, possibly with the differences (visual or not) between the instances.

This kind of explanations is very suitable for distance-based anomaly detection methods. Since the latter already rely on distance computation, it is easy, after the identification of anomalies, to evaluate the difference between regular

data points and outliers. It is applicable to cluster-based methods too, because they also rely on distances computations. More generally, it can be used with any anomaly detection algorithm. The difficulty of use comes from the choice of an appropriate distance/similarity metric. That being said, performing anomaly explanation using data points comparisons may turn out to be complicated with neural-network-based anomaly detection methods and their sometimes complex data types.

Displaying similar instances and showing the differences between the anomalous instance and similar instances allow the user to concretely perceive why a data point is irregular. But these explanations are very limited by the choice of a distance/similarity metric and require distances computation to find similar instances. We then find ourselves in a situation where, even if we avoid using distances computation to identify anomalies, we cannot escape them to provide explanations, which is not always applicable/desirable. Plus, this kind of explanation is not really relevant, when used alone.

3.4. Anomaly Explanation By Structure Analysis

This last category of explanations takes into account the structure of the dataset. Analyzing the structure means discovering in the dataset groups of regular data points, groups of irregular data points, instances which deviate from each group and instances that are in groups where they are not supposed to be. In the example from table 1, products can be grouped according to the model in order to identify and explain the anomalies of each model. For example, outlier *2* is an outlier for the model *iPhone 12* because its price is lower than usual, for products of this model. An explanation by structure analysis should provide this information. Besides that, regular products can be grouped according to the true price range, in order to obtain different ranges of products. For example in 1, high-end products can be those which true prices range in the interval $[800 - 1500]$, low-end products those which true prices range from 100 to 400 and, an intermediate range of products can contain those for which $unitprice \in [450 - 700]$. With this breakdown, an explanation by structure analysis for the outlier *2* is that according to its unit price it is a mid-range product, but it is not normal because products of this model are supposed to be high-end products. This kind of explanations can be provided by analyzing in details (possibly manually) the detected anomalies, but the goal is to simplify the process as much as possible, for humans and for the computer. Identifying the anomalies and giving directly this type of detailed explanations could be very useful. Some works have been identified along these lines, but this type of explanation is sorely lacking references.

In [64], the authors use a variant of the linearised fuzzy c-medoids algorithm to cluster the anomalies after detecting them using another anomaly detection algorithm. They were able to obtain distinct fraud profiles, but they did not reach the explanation step.

With x-PACKS [65], a subspace clustering is first performed on the dataset containing anomalies and normal data points. After that step, hyper-rectangles containing the maximum number of anomalous data points and the minimum

number of regular instances are obtained. Then each hyper-rectangles is refined into a hyper-ellipsoid in order to enclose as many outliers as possible and as few regular instances as possible. Finally, rules on every feature of the ellipsoid are generated and constitute the explanations for the set of anomalies contained in the ellipsoid. The explanations are computed after the anomalies identification which can be made using any algorithm; it is therefore a model-agnostic method.

The work that most closely belongs to this category is [66]. In this one, the authors derive a similarity measure from an Isolation Forest; a clustering (or more precisely an Agglomerative Hierarchical Clustering) of the regularities and the irregularities is then performed based on the similarity measure defined. After that, each cluster of anomalies is compared to each cluster of regular points based on their distinctive properties and, linguistic summaries, describing not only the properties of each cluster but also the differences between clusters, are generated.

Discussion

As this type of explanations was not extensively explored in the literature, the output can ideally be a text in natural language giving as much details as possible on the anomalies, or even a set of rules.

This kind of explanations is very suitable for model-based anomaly detection methods, especially with cluster-based methods because clusters explanation methods exist. It is more difficult to perform anomaly explanation by structure analysis with neural-network-based anomaly detection methods since the structure of the dataset is not really analyzed when using neural networks. Providing this kind of explanations starts at the anomaly detection level with the identification of different types of anomalies in the data set and the identification of local anomalies.

Anomaly explanation by structure analysis provides the most detailed information about why instances are anomalous. But it has not been deeply explored yet. The works identified as belonging to this category are either incomplete (do not return understandable explanations to the user) or a sequence of steps (anomaly detection -> clustering -> structure analysis of the clusters). No method in the literature has been able yet to provide a unified algorithm going directly from the detection to the detailed explanations. Also, all the methods identified here explain anomalies in groups. But structure analysis should also be able to explain why a specific data point is anomalous, and not only why a set of instances are anomalous.

3.5. Summary Of Anomaly Explanation

Table 3 below summarizes the works dealing with anomaly explanation. For each category of explanations, the outputs are recalled. The anomaly detection methods which can be used with each explanation type are specified, along with their difficulty of use (1=easy and 3=difficult). Then, each work is classified as local, global, model-agnostic or model-specific anomaly explanation method. The pros and cons of each category are also recalled. The complexity of each

category cannot be specified directly, since it depends on the method used (detection and/or explanation method) and on the desired level of detail. When using anomaly explanation by feature importance, if each feature is taken independently of the others, the complexity is linear; but if the interactions between the features are considered, the complexity will be exponential as a function of the number of attributes. As a result, performing dimensionality reduction prior to explanation by feature importance is generally a good idea. Anomaly explanation by data points comparisons offers a good compromise in terms of complexity, by limiting the number of comparisons between the data points. In this case, the nearest neighbor(s) must be computed first; and, if the distances are already computed by the detection method, there is no additional complexity when generating explanations. In conclusion, not only the detection method, but also the level of detail of the explanations has an impact on the complexity.

4. Conclusion

The goal of this work was to make a review of the anomaly explanation field. Four categories of anomaly explanation methods were defined: explanation by feature importance, explanation by feature values, explanation by data points comparison and explanation by structure analysis. The first category, anomaly explanation by feature importance, has been intensively explored in the literature. In contrast, a lot can be done in order to explain anomalies by structure analysis, which is the most desirable type of explanations because of the details provided. To obtain these richer explanations, feature importance, feature values and data points comparisons can be combined with the analysis of the dataset structure. This is a proof that the four categories are not mutually exclusive. The examination of the data structure can be made automatically by an algorithm (a clustering algorithm for example, as it has been done in the literature), or by exploiting the knowledge of an expert. It is necessary not to neglect the importance of an expert's knowledge when designing an anomaly detection/explanation system, as it can improve the quality of the system. Even though the most wanted setting in anomaly detection/explanation is a completely autonomous, human-intervention-free one, integrating expert knowledge can first help to identify real anomalies as stated in the introduction. Later, at the explanation step, it can help to describe the structure of the regularities and of some irregularities in order to facilitate the analysis of the dataset structure. The expert knowledge integration for structure analysis has not been seriously explored yet.

Acknowledgement

This research takes part of the SEA DEFENDER project funded by the French DGA (Directorate General of Armaments).

Table 3: Summary of anomaly explanation methods

			References				Pros & Cons	
Explanation type	**Possible outputs**	**Detection methods (ease of use)**	**Local**	**Global**	**Model-agnostic**	**Model-specific**	**Pros**	**Cons**
Feature importance	list of features (ordered or not, weighted or not), pair of features, list of features pairs, plot	Distance-based(1), Model-based(1), Neural-network-based(1)	[27], [28], [29], [30], [36], [37], [38], [39], [40], [41], [42], [43], [44], [23], [45], [46], [48]	[31], [32], [34], [41], [44], [47]	[31], [32], [34], [39], [43], [48]	[27], [28], [29], [30], [36], [37], [38], [40], [41], [42], [44], [23], [45], [46], [47]	easy to use with any detection method;can provide a global understanding of the anomalies;can help to identify different groups of anomalies	not detailed enough;less understandable when the original features have been transformed; can return no explanation at all
Feature values	decision rules, natural language	Distance-based(2), Model-based(1), Neural-network-based (3)	[50], [52], [53], [54], [55]	[51], [53]	[51], [53]	[50], [52], [54], [55]	precise explanations of where the anomalies are located; close to, and easy to translate into natural language	rules can easily become unreadable;more complicated to explain anomalies globally; complexity
Data points comparisons	closest instance (irregular or not), difference between two instances (graphical or not)	Distance-based(1), Model-based(2), Neural-network-based(3)	[56], [58], [60], [61], [62]	[59]	[58]	[56], [59], [60], [61], [62]	helps to perceive the difference between a regularity and an irregularity	limited by the choice of the distance or similarity metric;not really relevant when used alone
Structure analysis	decision rules, natural language	Distance-based(1), Model-based (1), Neural-network-based (3)		[65], [66]	[65]	[66]	detailed explanations;helps to perceive the difference between different kinds of instances, and between regularities and irregularities	can overwhelm the user if the dataset is complex and the explanations are not condensed enough

References

[1] D. M. Hawkins, Identification of outliers, Vol. 11, Springer, 1980.

[2] J. S. Zdanowicz, Trade-based money laundering and terrorist financing, Review of law & economics 5 (2) (2009) 855–878.

[3] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM computing surveys (CSUR) 41 (3) (2009) 1–58.

[4] G. O. Campos, A. Zimek, J. Sander, R. J. Campello, B. Micenková, E. Schubert, I. Assent, M. E. Houle, On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study, Data mining and knowledge discovery 30 (4) (2016) 891–927.

[5] M. Goldstein, S. Uchida, A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data, PloS one 11 (4) (2016) e0152173.

[6] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, K.-R. Müller, A unifying review of deep and shallow anomaly detection, Proceedings of the IEEE.

[7] M. Ahmed, A. N. Mahmood, J. Hu, A survey of network anomaly detection techniques, Journal of Network and Computer Applications 60 (2016) 19–31.

[8] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM computing surveys (CSUR) 51 (5) (2018) 1–42.

[9] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, et al., Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai, Information Fusion 58 (2020) 82–115.

[10] F. T. Liu, K. M. Ting, Z.-H. Zhou, Isolation-based anomaly detection, ACM Transactions on Knowledge Discovery from Data (TKDD) 6 (1) (2012) 1–39.

[11] M. M. Breunig, H.-P. Kriegel, R. T. Ng, J. Sander, Lof: identifying density-based local outliers, in: Proceedings of the 2000 ACM SIGMOD international conference on Management of data, 2000, pp. 93–104.

[12] J. Tang, Z. Chen, A. W.-C. Fu, D. W. Cheung, Enhancing effectiveness of outlier detections for low density patterns, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2002, pp. 535–548.

[13] D. Yu, G. Sheikholeslami, A. Zhang, Findout: Finding outliers in very large datasets, Knowledge and Information Systems 4 (4) (2002) 387–412.

[14] G. Münz, S. Li, G. Carle, Traffic anomaly detection using k-means clustering, in: GI/ITG Workshop MMBnet, 2007, pp. 13–14.

[15] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, J. C. Platt, et al., Support vector method for novelty detection., in: NIPS, Vol. 12, Citeseer, 1999, pp. 582–588.

[16] M. Amer, M. Goldstein, S. Abdennadher, Enhancing one-class support vector machines for unsupervised anomaly detection, in: Proceedings of the ACM SIGKDD workshop on outlier detection and description, 2013, pp. 8–15.

[17] S. Hariri, M. C. Kind, R. J. Brunner, Extended isolation forest, arXiv preprint arXiv:1811.02141.

[18] F. T. Liu, K. M. Ting, Z.-H. Zhou, On detecting clustered anomalies using sciforest, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2010, pp. 274–290.

[19] S. Hawkins, H. He, G. Williams, R. Baxter, Outlier detection using replicator neural networks, in: International Conference on Data Warehousing and Knowledge Discovery, Springer, 2002, pp. 170–180.

[20] A. L. Alfeo, M. G. Cimino, G. Manco, E. Ritacco, G. Vaglini, Using an autoencoder in the design of an anomaly detector for smart manufacturing, Pattern Recognition Letters 136 (2020) 272–278.

[21] J. Chen, S. Sathe, C. Aggarwal, D. Turaga, Outlier detection with autoencoder ensembles, in: Proceedings of the 2017 SIAM international conference on data mining, SIAM, 2017, pp. 90–98.

[22] S. Chaurasia, S. Goyal, M. Rajput, Outlier detection using autoencoder ensembles: A robust unsupervised approach, in: 2020 International Conference on Contemporary Computing and Applications (IC3A), IEEE, 2020, pp. 76–80.

[23] Q. P. Nguyen, K. W. Lim, D. M. Divakaran, K. H. Low, M. C. Chan, Gee: A gradient-based explainable variational autoencoder for network anomaly detection, in: 2019 IEEE Conference on Communications and Network Security (CNS), IEEE, 2019, pp. 91–99.

[24] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, U. Schmidt-Erfurth, f-anogan: Fast unsupervised anomaly detection with generative adversarial networks, Medical image analysis 54 (2019) 30–44.

[25] S. Rajendran, W. Meert, V. Lenders, S. Pollin, Unsupervised wireless spectrum anomaly detection with interpretable features, IEEE Transactions on Cognitive Communications and Networking 5 (3) (2019) 637–647.

[26] G. Pang, C. Shen, L. Cao, A. v. d. Hengel, Deep learning for anomaly detection: A review, arXiv preprint arXiv:2007.02500.

[27] E. M. Knorr, R. T. Ng, Finding intensional knowledge of distance-based outliers, in: Vldb, Vol. 99, Citeseer, 1999, pp. 211–222.

[28] B. Micenková, R. T. Ng, X.-H. Dang, I. Assent, Explaining outliers by subspace separability, in: 2013 IEEE 13th international conference on data mining, IEEE, 2013, pp. 518–527.

[29] H.-P. Kriegel, P. Kröger, E. Schubert, A. Zimek, Outlier detection in axis-parallel subspaces of high dimensional data, in: Pacific-asia conference on knowledge discovery and data mining, Springer, 2009, pp. 831–838.

[30] H.-P. Kriegel, P. Kröger, E. Schubert, A. Zimek, Outlier detection in arbitrarily oriented subspaces, in: 2012 IEEE 12th international conference on data mining, IEEE, 2012, pp. 379–388.

[31] N. Gupta, D. Eswaran, N. Shah, L. Akoglu, C. Faloutsos, Beyond outlier detection: Lookout for pictorial explanation, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2018, pp. 122–138.

[32] H. Liu, F. Ma, Y. Wang, S. He, J. Chen, J. Gao, Lp-explain: Local pictorial explanation for outliers, in: 2020 IEEE International Conference on Data Mining (ICDM), IEEE, 2020, pp. 372–381.

[33] S. Wang, H. Xia, G. Li, J. Tan, Group outlying aspects mining, in: International Conference on Knowledge Science, Engineering and Management, Springer, 2018, pp. 200–212.

[34] D. Qi, J. Arfin, M. Zhang, T. Mathew, R. Pless, B. Juba, Anomaly explanation using metadata, in: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2018, pp. 1916–1924.

[35] W. Polonik, Z. Wang, Prim analysis, Journal of Multivariate Analysis 101 (3) (2010) 525–540.

[36] M. A. Siddiqui, A. Fern, T. G. Dietterich, W.-K. Wong, Sequential feature explanations for anomaly detection, ACM Transactions on Knowledge Discovery from Data (TKDD) 13 (1) (2019) 1–22.

[37] X. H. Dang, B. Micenková, I. Assent, R. T. Ng, Local outlier detection with interpretation, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2013, pp. 304–320.

[38] X. H. Dang, I. Assent, R. T. Ng, A. Zimek, E. Schubert, Discriminative features for identifying and interpreting outliers, in: 2014 IEEE 30th international conference on data engineering, IEEE, 2014, pp. 88–99.

[39] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 30, Curran Associates, Inc., 2017, pp. 4765–4774. URL https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf

[40] L. Antwarg, B. Shapira, L. Rokach, Explaining anomalies detected by autoencoders using shap, arXiv preprint arXiv:1903.02407.

[41] I. Giurgiu, A. Schumann, Additive explanations for anomalies detected from multivariate temporal data, in: Proceedings of the 28th acm international conference on information and knowledge management, 2019, pp. 2245–2248.

[42] N. Takeishi, Shapley values of reconstruction errors of pca for explaining anomaly detection, in: 2019 international conference on data mining workshops (icdmw), IEEE, 2019, pp. 793–798.

[43] N. Takeishi, Y. Kawahara, On anomaly interpretation via shapley values, arXiv preprint arXiv:2004.04464.

[44] M. Carletti, M. Terzi, G. A. Susto, Interpretable anomaly detection with diffi: Depth-based feature importance for the isolation forest, arXiv preprint arXiv:2007.11117.

[45] J. Kauffmann, K.-R. Müller, G. Montavon, Towards explaining anomalies: a deep taylor decomposition of one-class models, Pattern Recognition 101 (2020) 107198.

[46] K. Amarasinghe, K. Kenney, M. Manic, Toward explainable deep neural network based anomaly detection, in: 2018 11th International Conference on Human System Interaction (HSI), IEEE, 2018, pp. 311–317.

[47] A. Brown, A. Tuor, B. Hutchinson, N. Nichols, Recurrent neural network attention mechanisms for interpretable system log anomaly detection, in: Proceedings of the First Workshop on Machine Learning for Computing Systems, 2018, pp. 1–8.

[48] X. Zhang, M. Marwah, I.-t. Lee, M. Arlitt, D. Goldwasser, Ace–an anomaly contribution explainer for cyber-security applications, in: 2019 IEEE International Conference on Big Data (Big Data), IEEE, 2019, pp. 1991–2000.

[49] M. T. Ribeiro, S. Singh, C. Guestrin, " why should i trust you?" explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016, pp. 1135–1144.

[50] E. Baseman, S. Blanchard, N. DeBardeleben, A. Bonnie, A. Morrow, Interpretable anomaly detection for monitoring of high performance computing systems, in: Outlier Definition, Detection, and Description on Demand Workshop at ACM SIGKDD. San Francisco (Aug 2016), 2016.

[51] A. Barbado, Ó. Corcho, R. Benjamins, Rule extraction in unsupervised anomaly detection for model explainability: Application to oneclass svm, arXiv preprint arXiv:1911.09315.

[52] F. Song, Y. Diao, J. Read, A. Stiegler, A. Bifet, Exad: A system for explainable anomaly detection on big data traces, in: 2018 IEEE International Conference on Data Mining Workshops (ICDMW), IEEE, 2018, pp. 1435–1440.

[53] M. Kopp, T. Pevnỳ, M. Holeňa, Anomaly explanation with random forests, Expert Systems with Applications 149 (2020) 113187.

[54] S. Haldar, P. G. John, D. Saha, Reliable counterfactual explanations for autoencoder based anomalies, in: 8th ACM IKDD CODS and 26th COMAD, 2021, pp. 83–91.

[55] M. Munir, S. A. Siddiqui, F. Küsters, D. Mercier, A. Dengel, S. Ahmed, Tsxplain: Demystification of dnn decisions for time-series using natural language and statistical features, in: International Conference on Artificial Neural Networks, Springer, 2019, pp. 426–439.

[56] H.-P. Kriegel, M. Schubert, A. Zimek, Angle-based outlier detection in high-dimensional data, in: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, 2008, pp. 444–452.

[57] T. Mokoena, Why is this an anomaly? explaining anomalies using sequential explanations, Ph.D. thesis (2019).

[58] K. Rieck, P. Laskov, Visualization and explanation of payload-based anomaly detection, in: 2009 European Conference on Computer Network Defense, IEEE, 2009, pp. 29–36.

[59] M. Mejia-Lavalle, Outlier detection with innovative explanation facility over a very large financial database, in: 2010 IEEE Electronics, Robotics and Automotive Mechanics Conference, IEEE, 2010, pp. 23–27.

[60] Z. Li, G. Liu, S. Wang, S. Xuan, C. Jiang, Credit card fraud detection via kernel-based supervised hashing, in: 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), IEEE, 2018, pp. 1249–1254.

[61] Z. Miao, Q. Zeng, C. Li, B. Glavic, O. Kennedy, S. Roy, Cape: explaining outliers by counterbalancing, Proceedings of the VLDB Endowment 12 (12) (2019) 1806–1809.

[62] A. Smith-Renner, R. Rua, M. Colony, Towards an explainable threat detection tool., in: IUI Workshops, 2019.

[63] S. Akcay, A. Atapour-Abarghouei, T. P. Breckon, Ganomaly: Semi-supervised anomaly detection via adversarial training, in: Asian conference on computer vision, Springer, 2018, pp. 622–637.

[64] M.-J. Lesot, A. Revault d'Allonnes, Credit-card fraud profiling using a hybrid incremental clustering methodology, in: E. Hüllermeier, S. Link, T. Fober, B. Seeger (Eds.), Scalable Uncertainty Management, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 325–336.

[65] M. Macha, L. Akoglu, Explaining anomalies in groups with characterizing subspace rules, Data Mining and Knowledge Discovery 32 (5) (2018) 1444–1480.

[66] A. K. Shukla, G. Smits, O. Pivert, M.-J. Lesot, Explaining data regularities and anomalies, in: 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2020, pp. 1–8.