Contribution to Anomaly Detection and Explanation

Véronne Yepmo Tchaghe

COLLEGE MATHS, TELECOMS
DOCTORAL INFORMATIQUE, SIGNAL
BRETAGNE SYSTEMES, ELECTRONIQUE

Université de Rennes

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE RENNES

ÉCOLE DOCTORALE N° 601
Mathématiques, Télécommunications, Informatique, Signal, Systèmes, Électronique
Spécialité : *Informatique*

Par
Véronne YEPMO TCHAGHE

Contribution to Anomaly Detection and Explanation

A Unified Method based on Isolation Forest

Thèse présentée et soutenue à ENSSAT Lannion, France, le 20 Décembre 2023
Unité de recherche : UMR 6074 - Institut de Recherche en Informatique et Systèmes Aléatoires

Rapporteurs avant soutenance :

Raja CHIKY	Professeure, 3iL Ingénieurs, Limoges
Christophe MARSALA	Professeur, Sorbonne Université, Paris

Composition du Jury :

Président :	Christophe MARSALA	Professeur, Sorbonne Université, Paris
Examinateurs :	Peggy CELLIER	Maîtresse de Conférences HDR, INSA Rennes
	Raja CHIKY	Professeure, 3iL Ingénieurs, Limoges
	Anne LAURENT	Professeure, Université de Montpellier
	Christophe MARSALA	Professeur, Sorbonne Université, Paris
Dir. de thèse :	Olivier PIVERT	Professeur, ENSSAT, Lannion
Co-dir. de thèse :	Grégory SMITS	Professeur, IMT Atlantique, Brest

À mon père, qui voulait que je sois docteure en médecine.
À ma mère, éducatrice hors pair.

Acknowledgements

A Ph.D. is a journey. As I approach the culmination of this three-year odyssey, I would like to extend my heartfelt gratitude to those who have played a pivotal role in making this experience as enjoyable as it has been for me.

First of all, I would like to thank my Ph.D. supervisors, Grégory SMITS and Olivier PIVERT, who accepted me as their Ph.D. student. Besides the work aspect, I believe that it is important to be surrounded by great people. And I was lucky enough to have as main supervisor one of the kindest persons I know. Grégory is not only a hardworking and brilliant researcher. He is also an amazing human-being. And I thank him for these three years. I extend my appreciation to Olivier for always having the right comment during meetings.

Secondly, I wish to express my gratitude to Raja CHIKY and Christophe MARSALA who kindly accepted to review this manuscript. I would also like to thank Anne LAURENT and Peggy CELLIER for agreeing to be part of the jury of my defense.

Thirdly, I would like to thank Marie-Jeanne LESOT, my co-author and unofficial thesis advisor. I am grateful for the interesting scientific discussions we had together, and also the non-scientific ones.

Furthermore, my heartfelt thanks go to the members of the SHAMAN team (former and current). A special thank to my office mate and dear friend Wafaa, my favorite Lebanese, for these amazing three years we spent together. Thanks to Angélique, SHAMAN's administrative assistant, for all the train and hotel reservations.

Before I started my Ph.D., I had the opportunity take a peak into the world of research through a research Master after my Engineering degree. This chance was provided by Engelbert MEPHU, whom I thank for allowing me to discover the fascinating world of research.

I cannot forget my family and friends for their support, especially Rostan NANA who sent me this Ph.D. offer by e-mail back in September 2020. Thanks to my sister and brothers Rubenne, Nell, Vianney and Vernes for their unconditional support throughout my studies.

Finally, I would like to thank everyone who, at one point or another, asked me how

the Ph.D. was going and wished me luck, including that Uber driver who drove me to the Halifax airport.

Table of Contents

List of Figures

List of Tables

Notations and Abbreviations

The notations used throughout the dissertation are summarized here.

Notation	Meaning
$\mathcal{D}$	Data set of N points
$\mathcal{A} = \{A_1, \ldots, A_d\}$	Descriptive attributes
$dom(A)$	Domain of attribute A
I^A	Interval on feature $A \in \mathcal{A}$
$x \in \mathcal{D}$	Data point
$x.A$	Value for data point x on attribute A
C	Set of discarded attributes
t	Size of the forest $\mathcal{F} = \{T_1, \ldots, T_t\}$
Ψ	Size of the sample used to build a tree
h_{lim}	Depth limit
$\eta_i(x)$	Terminal node containing x in the i-th tree
α	Margin width percentage
$marg$	Margin width size
$s(x)$	Anomaly score of x
$\mathcal{C}$	Partition of $\mathcal{D}$ in k clusters $\{C_1, \ldots, C_k\}$
T_{l_i}	Tree containing the leaf l_i
$c(x, C)$	Contextual score of instance x with respect to cluster C
$\nabla_x^{l_i}$	Deepest common ancestor between the paths from the root of T_{l_i} to l_i and $\eta_i(x)$
γ	Anomaly score threshold
τ	Edge weight threshold

The following abbreviations are employed within the thesis. This list is in alphabetical order.

Abbreviation	**Meaning**
AE	Autoencoder
AHC	Agglomerative Hierarchical Clustering
AI	Artificial Intelligence
ARI	Adjusted Rand Index
ATON	Attention-guided Triplet deviation network for Outlier interpretatioN
CADI	Contextual Anomaly Detection using Isolation-Forest
CBLOF	Cluster-Based Local Outlier Factor
COIN	Contextual Outlier INterpretation
COP	Correlation Outlier Probability
DL	Deep Learning
DLN	Depth-Limit Node
DN	Dense Node
ECOD	Empirical-Cumulative-distribution-based Outlier Detection
EIF	Extended Isolation Forest
GAN	Generative Adversarial Network
HPC	High Performance Computing
IF	Isolation Forest
IN	Isolation Node
LOF	Local Outlier Factor
LRP	Layer-wise Relevance Propagation
LSTM	Long Short Term Memory
ML	Machine Learning
PCA	Principal Component Analysis
SHAP	SHapley Additive exPlanations
SOD	Subspace Outlier Degree
XAI	eXplainable Artificial Intelligence

Résumé en français

Les résultats de recherche décrits dans ce manuscrit s'inscrivent dans le cadre d'un projet collaboratif avec un éditeur de logiciels spécialisé dans l'agrégation des données décrivant des activités de commerce maritime international (marchandises, parties prenantes, navires, routes, règlementations, etc.). Un enjeu majeur de ce domaine d'activité est de fournir des outils d'aide à l'analyse de ces données. Plus précisément, la problématique métier abordée dans ce projet est de détecter automatiquement des cas de sur et de sous-facturation qui constituent le premier vecteur de blanchiment d'argent. Les données sur les marchandises issues du commerce maritime sont regroupées en quelques catégories de spectre très large, comme la catégorie *téléphonie*. Les données analysées au sein d'une catégorie sont donc très hétérogènes, allant de l'accessoire de téléphone à 1€ au téléphone satellitaire militaire valant plusieurs milliers d'euros. Déterminer si une valeur, telle qu'un prix, est anormalement élevée ou faible nécessite de disposer d'une connaissance sur les groupes de produits (accessoires, smartphone haut de gamme, téléphonie militaire, etc.). Un score d'anormalité associé automatiquement à une marchandise n'est pas une information suffisante lorsqu'il s'agit de construire des outils d'aide à la décision qui seront utilisés par des êtres humains. Il est nécessaire de fournir des explications sur l'origine de ce score et des connaissances additionnelles sur les données comparées.

La transposition de cette problématique en question de recherche a conduit à étudier les méthodes de détection d'anomalies, i.e. de données suspectes, et de génération d'explications des raisons pour lesquelles un point est considéré comme anormal. Ces explications doivent être contextuelles, c'est-à-dire prendre en compte la structure des données régulières, structure issue d'un partitionnement. Le prix mentionné n'est pas anormal uniquement parce qu'il est trop bas, mais parce qu'il est trop bas *pour* un téléphone haut de gamme. Alors que la génération d'explications contextuelles pourrait être effectuée en utilisant trois méthodes distinctes liées à chaque problématique (détection d'anomalies, partitionnement des données, explication d'anomalies) comme cela a été fait dans la littérature, cette thèse propose une méthode unifiée réalisant les trois tâches et qui s'appuie sur la notion de forêt d'isolation.

Contexte et travaux connexes

Dans le chapitre 1, la terminologie de la détection d'anomalies est tout d'abord rappelée. Une distinction entre les notions de donnée aberrante, bruit, anomalie, nouveauté et donnée régulière est faite. Puis, quelques notions sur la détection d'anomalies proprement dite sont rappelées, notamment les concepts d'anomalies locale et globale, et celui d'anomalies conditionnelles. Le chapitre se poursuit avec un état de l'art des méthodes automatiques de détection d'anomalies. Une analyse aboutissant à une taxonomie des méthodes d'explication d'anomalies est ensuite présentée, taxonomie qui aura été notre première contribution au domaine. En comparaison avec les taxonomies existantes, qui se focalisent sur la méthode ayant généré les explications, notre taxonomie se concentre sur l'information véhiculée par l'explication générée. Nous établissons ainsi une distinction entre les explications par importance d'attribut, les explications par valeurs d'attributs, les explications par comparaison de points et les explications par analyse de la structure intrinsèque des données. Les explications par importance d'attribut associent à chaque attribut ou dimension un poids quantifiant son importance dans l'identification de l'anomalie. Les explications par valeurs d'attributs sont constituées des régions dans lesquelles se trouvent les anomalies. Les explications par comparaison de points renvoient comme explication un point ou un groupe de points représentatifs de la normalité ou de l'anormalité. Les explications par analyse de la structure intrinsèque des données quant à elles établissent un lien avec les groupes de données régulières dans le jeu de données. Ce dernier type d'explications, qui a été peu exploré dans la littérature, est le plus pertinent dans le cadre du projet SEA Defender à cause du rapprochement avec des groupes de données régulières. Même si l'explication d'anomalies a été moins explorée dans la littérature que l'explication des classifieurs et des réseaux de neurones, elle ne saurait être considérée comme moins importante. Au contraire, étant donné le caractère imprévisible des anomalies, fournir des explications quant à leur anormalité est d'un grand intérêt pour les utilisateurs, avant tout pour se rassurer quant au fait que l'anomalie identifiée en est vraiment une. Le chapitre 1 présente également un état de l'art des méthodes de partitionnement dites robustes. Alors que les méthodes de partitionnement classiques sont perturbées par la présence d'anomalies dans le jeu de données, les méthodes de partitionnement robustes tiennent compte des données anormales et sont par conséquent moins vulnérables à la présence de ces dernières.

CADI

Dans le chapitre 2, nous présentons notre approche unifiée intitulée CADI pour Contextual Anomaly Detection using Isolation-Forest. Même si le nom de la méthode pourrait renvoyer aux méthodes de détection d'anomalies conditionnelles évoquées dans le chapitre 1, le contexte ici est bel et bien relatif au rapprochement d'anomalies de la structure des données régulières. CADI s'appuie sur une version revisitée de l'algorithme des forêts d'isolation. Une forêt d'isolation est un ensemble d'arbres construits en divisant l'espace de données de manière récursive et aléatoire, avec l'hypothèse que les anomalies seront isolées dans les feuilles plus rapidement que les données régulières. Les forêts d'isolation demeurent un algorithme très efficace de détection d'anomalies qui possède très peu d'hyper-paramètres et est interprétable à l'échelle des arbres. Tandis que les forêts d'isolation classiques utilisent des séparations complètement aléatoires, CADI possède un critère de sélection des séparations. Les séparations conservées et utilisées pour construire les arbres de la forêt sont celles qui ne séparent pas les points appartenant au même cluster. Pour évaluer cela, une marge est définie autour de chaque séparation tirée aléatoirement : si beaucoup de points se retrouvent dans cette marge, elle est potentiellement en train de traverser un cluster. Elle est donc abandonnée au profit d'une autre. Ce critère assez simple permet de ne pas rajouter trop de complexité à la méthode classique des forêts d'isolation qui, moins complexe que la plupart de ses compétiteurs, est aussi très efficace. La modification de la sélection complètement aléatoire des séparations donne lieu à trois types de feuilles dans les arbres de notre nouvelle forêt: des feuilles contenant des points isolés, des feuilles ayant atteint la profondeur maximale des arbres et enfin des feuilles contenant des points qui ne peuvent plus être séparés. Ces dernières, appelées feuilles DN (pour **Dense Node**), contiennent donc des portions de clusters. Elles sont ensuite fusionnées après l'identification des anomalies pour retrouver une partition des données régulières. La combinaison des feuilles DN, qui est inspirée du *grid-based clustering*, permet non seulement de découvrir des clusters non-elliptiques, mais aussi de découvrir automatiquement le nombre de groupes dans le jeu de données, tout en évitant de calculer des distances entre les paires de points. Après l'identification des anomalies puis celle des clusters de données régulières, les anomalies sont rapprochées des clusters identifiés. Ce rapprochement est effectué en analysant les arbres de la forêt et en comptant le nombre de séparations entre les feuilles DN composant chaque cluster et l'anomalie à rapprocher. Le chemin entre chaque feuille DN et l'anomalie est à nouveau

exploité pour générer des explications contextuelles des anomalies.

Expérimentation

Dans le chapitre 3, les expériences menées pour évaluer notre approche sont présentées. Chaque composante, la détection d'anomalies, la reconstruction de la structure intrinsèque des données, puis la génération d'explications contextuelles, est évaluée. S'agissant de la détection d'anomalies, des jeux de données classiques de détection d'anomalies sont exploités. CADI est comparée à la version initiale des forêts d'isolation, ainsi qu'à d'autres algorithmes non supervisés de détection d'anomalies. La sensibilité de l'hyper-paramètre additionnel exploité par CADI et contrôlant la largeur de la marge autour des séparations est également évaluée. Il en ressort que la méthode est robuste au choix de cet hyper-paramètre, mais aussi que CADI est capable de mieux identifier les anomalies locales dans les jeux de données. Avant d'évaluer le partitionnement produit par CADI, nous montrons d'abord que les informations contenues dans les feuilles des arbres, en particulier les feuilles DN, peuvent être combinées pour obtenir une partition des données régulières. Pour ce faire, une série de tests est réalisée sur des données synthétiques contenant des anomalies et des clusters. Il en résulte que les feuilles DN contiennent des points proches entre eux et séparés des points contenus dans les autres feuilles DN. La partition extraite suite à la combinaison des feuilles DN est ensuite évaluée et comparée à celles produites par d'autres algorithmes de partitionnement robuste. S'agissant de l'évaluation de la contextualisation des anomalies (rapprochement aux clusters) ainsi que des explications, il a fallu générer des données synthétiques avec des clusters, des anomalies locales ainsi que les vraies informations sur le(s) cluster(s) dont se rapproche chaque anomalie et les vraies informations concernant les attributs communs et discriminants entre chaque anomalie et chaque cluster.

La conclusion de ce manuscrit résume nos contributions puis présente les perspectives de ce travail. Parmi les perspectives, l'évaluation de l'impact de la profondeur limite des arbres est particulièrement intéressante, ainsi que ses liens avec le choix des valeurs des hyper-paramètres. D'autre part, l'évaluation des explications générées reste un problème récurrent dans la communauté de l'intelligence artificielle explicable, qui mérite d'être exploré. Finalement, CADI n'aura pas pu être testée sur des données issues des transactions du commerce international. Une fois ces données disponibles, il serait enrichissant d'appliquer l'approche et de voir dans quelle mesure des connaissances ex-

pertes sur l'interprétation des données issues du commerce international pourraient être apportées à chaque étape du processus.

Introduction

Context

Artificial Intelligence (AI) is becoming an increasingly important part of our lives. From Deep Blue defeating Garry Kasparov in 1997 to autonomous vehicles, it has come a long way and shows no signs of halting its ascent since. AI is now everywhere. Machine Learning (ML) and its subset, Deep Learning (DL), are components of AI in which the algorithm learns/discovers patterns from examples. The first step before applying these and other data mining algorithms is usually to pre-process the data set. One stage of data pre-processing, data cleaning, involves eliminating abnormal observations. These abnormal observations, called outliers, lower the performance of ML algorithms. Discarding them therefore often results in a performance improvement. However, some outliers may indicate a problem in the data source, such as sensors fault, which deserves attention. Outliers can also contribute to the understanding of the normal data. As a result, there is a task devoted to the identification of these instances in a data set: outlier/anomaly detection.

Definition 1: Outlier

An observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism. [Haw80]

In most applications, the data is created by one or more generating processes. When the generating process behaves unusually, it results in the creation of outliers [Agg16]. While all outliers are deviating instances, including *noise*, anomalies are the most deviating observations which are of interest to an analyst.

Anomaly detection has many applications in the real-world. One classic example is spam detection where a mail server has to decide if an incoming mail is an unwanted mail or not. 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. Another application of anomaly detection is intrusion detection in networks: unusual behaviours in networks traffic must be identified to fight against intruders who 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 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. Finally, in medicine, MRI photographs can be analyzed to identify cancerous cells.

An anomaly detection algorithm flags an instance as anomalous or not, sometimes with a score indicating how abnormal the instance is. However, with the ubiquity of AI, there is a need for algorithms to provide, in addition to their outputs, the reason why they produced that output. The expression eXplainable Artificial Intelligence (XAI) was born, coined by DARPA [GA19]. This expression gathers all the methods that provide explanations to the output of algorithms. It has gained popularity, especially with the outbreak of deep learning. Nowadays, there is hardly any major AI conference without at least an XAI session. A great deal of work has been devoted to anomaly detection in the literature, but not as much to anomaly explanation.

Motivation: The Sea Defender Project

The Sea Defender Project supported by the French Directorate General of Armaments (DGA) aims at fighting against trade-based money laundering. Money laundering is the process of concealing the existence, illegal source, or application of income derived from a criminal activity, and the subsequent disguising of the source of that income to make it appear legitimate [Zda09]. Trade-based money laundering occurs primarily through abnormal pricing, that is, over- and under-invoicing. Over-(respectively under-)invoicing happens when the price reported for the transaction is higher (respectively lower) than the actual price. In addition to allowing its perpetrators and their accomplices to evade income taxes or import duties, this illegal procedure contributes to the perpetuation of terrorist activities. For example, it was reported that between 2006 and the first half of 2016, 821 millions of dollars were under-invoiced during copper exportations from Chile

to Japan [HP19]. As a result, governments, with the help of banks, have deployed means to identify false international trade invoicing. The methods employed include [Zda09]:

— a comparison between the average import/export price of the product in the country involved and the average world price,

— the identification of invoice prices which are 50% below or above the average import/export price,

— an inter-quartile range price analysis.

Among these methodologies, the inter-quartile range price analysis, which is the most recent one and a standard since, is the most realistic. The first methodology was criticised because it did not take into consideration the country/product heterogeneity. The second one was criticized because the 50% filter was arbitrary. However, the inter-quartile range price analysis still performs a coarse analysis of the prices. Let us assume that the product imported/exported is a smartphone. Among smartphones, there are entry-level smartphones costing around 150€, mid-range phones costing in average 400€, and high-end phones whose prices can exceed 1200€. Analyzing the prices of smartphones as a whole, and comparing them to the one specified on the invoice does not take into account these disparities in characteristics. If the price marked on the invoice is 1400€, and the smartphone has the characteristics of a high-end phone, this is certainly not a case of under-invoicing. On the other hand, if the smartphone has the characteristics of an entry-level phone, but for the same aforementioned price, there is definitely something wrong.

Problem Statement

The scientific objective of the Sea Defender Project is to use more recent techniques to identify these cases of abnormal pricing, while taking into consideration the disparities mentioned previously. From an AI perspective, over- and under-invoicing are anomalies when considering as data the set of transactions. Automatically identifying the disparities across the regular transactions implies making homogeneous groups of data. As such, it is related to data clustering. Finally, as the final users of the system are humans and the domain is sensitive, the explanation component is crucial. To sum up, the objective is, given a data set (set of transactions in the case of the project, or a more general data set from a scientific perspective), to identify groups of regular data points, anomalies

deviating from each group(s) and provide explanations of these anomalies in relation to these identified groups of regular instances. This explanation is to say, for example, that a 400€ iPhone 15 is an anomaly for high-end phones because, although it shares some characteristics with the other high-end phones, its price is too low. This problem can be solved using a pipeline. The first component of the pipeline would be an anomaly detector separating regular instances from outliers. The second component would be a clustering algorithm aiming at dividing the regular instances into groups. The last component would make a contrast between the structure of the regular instances from the second component and the outliers identified by the first component to generate explanations.

Problem Statement

Can we replace that pipeline with a unified method?

Contributions

The research presented in this manuscript provides an answer to the question above. It therefore deals with anomaly detection and explanation, with an emphasis on the latter. As mentioned earlier, anomaly explanation has been less explored in the literature than anomaly detection. It also received less attention than the explanation of classifiers outputs. As a result, our first contribution is a categorization of anomaly explanation methods, followed by a review of existing works and motivated by the lack of state of the art in the field. The second and main contribution of this work is a unified method to identify and explain anomalies in relation to groups of regular data points, called CADI for Contextual Anomaly Detection using Isolation-Forest. This method is based on a revisited version of the Isolation Forest (IF) algorithm [LTZ12] which is a popular anomaly detector. This extension allows to isolate anomalies, identify clusters of regular instances, and generate contrastive explanations of each anomaly with regard to the clusters of regular instances, altogether based on the same data structure. While the few existing methods to extract contrastive explanations of anomalies depend on external anomaly detectors and external clustering algorithms, CADI does not.

The source code is available to the scientific community at: `https://gitlab.com/yveronne/cadi`.

Roadmap

The remainder of this dissertation is organized as follows:

— Chapter 1 first details the background on anomaly detection. It then presents our taxonomy of anomaly explanation methods and our state of the art. The last section of the chapter covers outlier-aware clustering.

— Chapter 2 details our dedicated approach to identify anomalies, groups of regular data points and generate contrastive explanations of anomalies in relation to these groups. Changes made to the Isolation Forest algorithm are specified, as well as the new properties resulting from them.

— Chapter 3 extensively presents the results of the experiments conducted on the proposed method. These experiments evaluate the anomaly detection component, the clustering component and the explanation generation component.

The dissertation ends with a conclusion including a summary of the work performed and the future directions.

Chapter 1

Background and Related Work

This chapter lays the foundations for this work. In Section 1.1, some notions about anomalies are recalled. Then, the existing works on the three scientific problems addressed in this work are reviewed. These are: anomaly detection (reviewed in Sec. 1.2), anomaly explanation (Sec. 1.3) and outlier-aware clustering (Sec. 1.4). In Section 1.2, there is an emphasis on the most popular methods.

Most of the content of this chapter has been published in the proceedings of the AIMLAI workshop of ECML/PKDD 2021 [TSP21] and in the journal Data & Knowledge Engineering (DKE) [YSP22].

Contents

1.1 Terminology and Background

In the introductive chapter, we stated that the data is created by one or more generating processes, and that outliers are deviating instances arousing suspicion they were generated by a different mechanism (Definition 1). In opposition to outliers are *inliers* which do not deviate from the generating processes.

Definition 2: Inlier

An observation generated by one of the processes creating the data.

Inliers are also called *regular* or *normal instances*. Among the deviating instances, *noise*, *anomalies* and *novelties* are found. An example of noise in a data set is a temperature specified in Fahrenheit instead of Celsius. Observations qualified as noise are often caused by an imperfection on the generating processes, measurement or reporting errors. As a result, they slightly deviate from the inlying observations and sometimes do not make sense at all. They are still generated by one of the processes, but perturbed enough to not be fully considered as inliers.

Definition 3: Noise

A perturbed inlier; still generated by one of the process creating the data, but deviating.

Anomalies on the other hand are those instances which deviate significantly enough from inliers, to the point where they are more likely to have been generated by a different process. While noise are incorrect and/or meaningless observations, anomalies are correct, which causes the analyst to be even more suspicious. A temperature specified in Celsius (like the other temperatures in the data set), but too high, is an anomaly.

Definition 4: Anomaly

A deviating observation potentially generated by a process different than the ones creating the data.

The definition above is similar to Definition 1 by Hawkins. For this reason, the definition by Hawkins is usually the one employed in the literature to refer to an anomaly. A more general definition of an outlier, to make a contrast with inlying observations, which is the seminal usage of the term, would be "a deviating instance". And then, according to the deviation degree, the outlier can be noise or an anomaly.

Novelties are similar to anomalies. However, it is certain that they have been generated by a different mechanism, previously unseen. A new star is a novelty given our assurance that it has not been witnessed before.

Definition 5: Novelty

A deviating observation generated by a newly identified stable process, different from the ones creating the data.

Figure 1.1 shows a data set in two dimensions.

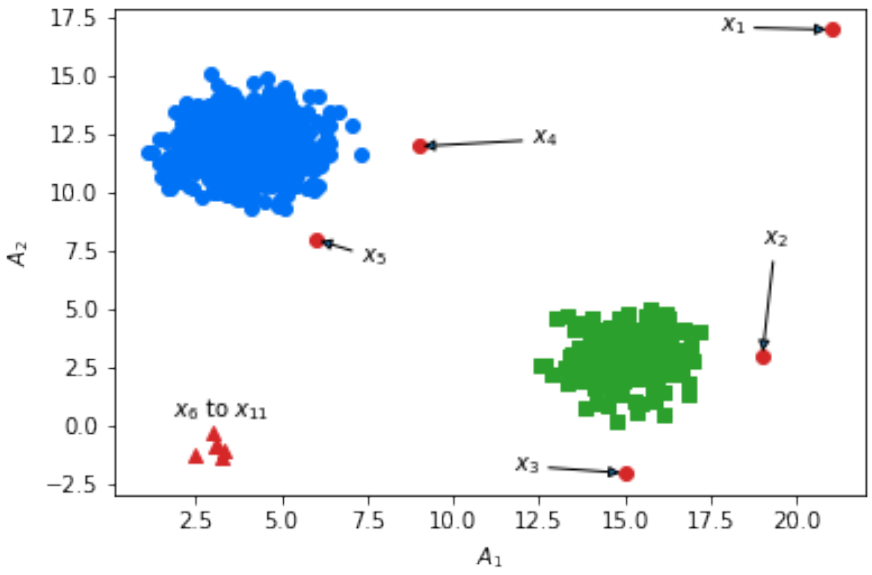


Figure 1.1 – A data set in two dimensions

The points from the two big clusters in Fig 1.1 are the inliers. A few points in the data set are suspicious: x_1 to x_{11}. Those data points are deviating from other observations: they are outliers. There are also two instances mildly deviating from the blue

(circles) cluster and which are not annotated. These two observations may be considered as noise as they are a bit detached from the cluster. In other data sets, there may be a randomly distributed noise. Yet, on Fig. 1.1, the instances x_1 to x_{11} are the ones capturing attention. Some authors [KN99] refer to noise as *weak outliers* to make a contrast with anomalies which are called *strong outliers*. In that case, outliers are all the deviating instances, but anomalies are the strongest deviating instances. From the user perspective, the latter are the most interesting ones as they make the user skeptical, even in the presence of noise in the data set. They are therefore the main focus of outlier detection algorithms, even if both weak and strong outliers can be identified. We will use the words anomalies and outliers interchangeably throughout this work to refer to those strong deviating instances. The expression *novelty detection* is also employed in the literature. Sometimes [MP03], it refers to the detection of anomalies, the concepts of anomaly and novelty being interchanged without distinction. However, it also refers in some works to the identification of new phenomena/generating processes, which is more in line with Definition 5. We will therefore avoid using that expression and remain consistent with the wording anomaly/outlier detection.

In general, anomalies can be divided into four types using two dimensions [LTZ10]. The first dimension considers the proximity to normal instances. A distinction is therefore made between *global* anomalies deviating from the other points in the data set, and *local* anomalies which deviate from a subset of the data set. In Fig. 1.1, x_1 is a global anomaly while x_2, x_3, x_4 and x_5 are local anomalies. The second dimension takes into account the distribution of anomalies: *scattered* anomalies are dispersed throughout the data set, while *clustered/collective* anomalies are close to each other, forming a small cluster. In Fig. 1.1, x_6 to x_{11} belong to the latter, while the other abnormal instances belong to the former category. The two dimensions are not mutually exclusive: the instances x_6 to x_{11} are collective global anomalies. In [Ruf+21] a distinction is made between *point anomalies*, *group anomalies*, *contextual anomalies*, *low-level sensory anomalies* and *high-level semantic anomalies*. Group anomalies are collective anomalies, while point anomalies are individual outliers that can be local or global. Contextual anomalies, also called *conditional* or *dependency-based* are a body of research on their own [Son+07; LP16; LL23]. A conditional anomaly is outlying given a specific context. This context can be time-related: a temperature of $25°C$ is abnormal in January in France, but perfectly normal during summer. To identify these dependency-based outliers, the set of features is divided into contextual features and behavioral features. Instances having the same context (e.g.:

time) are expected to have similar values on behavioral features (e.g.: temperature). Dependency-based anomaly detection flags as outliers the instances violating those constraints. Figure 1.2 depicts an example. Low-level and high-level anomalies occur when there is a hierarchy on the features. Low-level anomalies can be character typos in words, while semantic anomalies can be misposted news articles. These last two categories are not much explored in the literature.

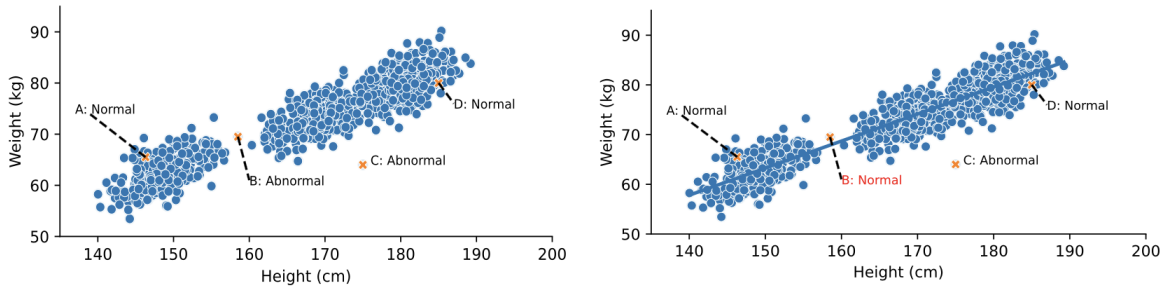


Figure 1.2 – The difference between point (left) and conditional (right) anomalies [LL23]. There is a linear relationship between height and weight. As a result, B is not a conditional anomaly, since it does not violate the dependence.

In regards to the second taxonomy, our work falls in the point anomaly detection category. Dependency-based anomaly detection assumes there is a partition of the features, often provided by a domain expert. This is not an hypothesis that holds in our particular scenario.

1.2 Anomaly detection

Let $\mathcal{D}$ be a data set containing n instances: $\mathcal{D} = \{x_1, x_2, ..., x_n\}$.
Let $\mathcal{A} = \{A_1, A_2, ..., A_d\}$ be the set of features/dimensions.

Definition 6: Anomaly detector

An anomaly detector is a function $f : \mathcal{D} \to \mathbb{F} \subset \mathbb{R}$.

$\mathbb{F}$ can be a 2-element set or a continuous set. When $\mathbb{F}$ is continuous, the anomaly detector returns for each instance x in the data set an *anomaly score* indicating how abnormal/deviating the instance is. In this situation, the contrast between weak and strong

outliers, mentioned in Sec. 1.1, is reflected in the score as illustrated on Fig. 1.3. When $\mathbb{F}$ is binary, the anomaly detector just indicates if the instance is regular or abnormal. The data set is then partitioned into two: the set of anomalies $\mathcal{D}_{\mathcal{O}}$ and the set of inliers $\mathcal{D}_{\mathcal{I}}$, with $|\mathcal{D}_{\mathcal{O}}| \ll |\mathcal{D}_{\mathcal{I}}|$. Even when the anomaly detection algorithm returns scores, a binary partition of the data set can be obtained by selecting an *anomaly score threshold*.

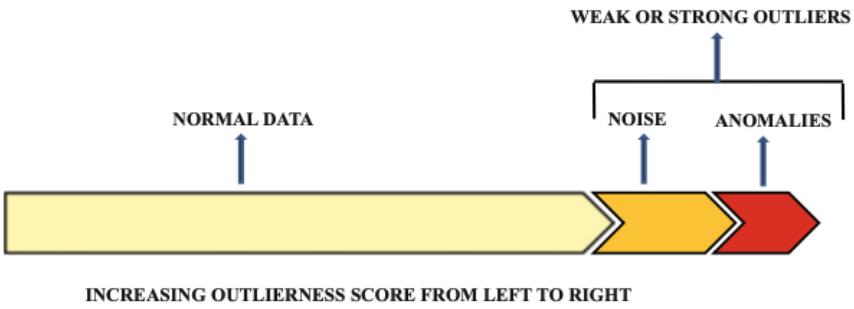


Figure 1.3 – The spectrum from inliers to outliers [Agg16].

The earliest works on outlier detection stem from the statistics community. Examples are extreme-value analysis and Gaussian Mixture Models (GMMs). These statistical-based methods are mathematically more precise. However, they make simplified assumptions about data representations and have poor algorithmic scalability and interpretability [Agg16]. ECOD (Empirical-Cumulative-distribution-based Outlier Detection) [Li+22] is a statistical anomaly detection method which is interpretable. The outlier score of an instance is the aggregate of its tail probabilities across all dimensions. ECOD therefore assumes that all the features are independent from each other. Information-theory-based outlier detection methods are also available. The hypothesis behind these methods is that outliers increase the minimum length of the data summary. In the remainder of this section, we focus on ML methods for outlier detection.

There is no unified taxonomy of anomaly detectors in the ML literature. According to the availability of labels, a distinction is made between *supervised* algorithms and *unsupervised* algorithms. Supervised anomaly detection methods use a labelled data set during training. Identifying anomalies therefore becomes a binary classification task in which there is high imbalance between the classes, since anomalies are far outnumbered by inliers. If no labels are present, anomaly detection is performed in an unsupervised manner: there is no training, the data are fed to the algorithm which identifies the outliers.

The last setting is more convenient since labelling a data set 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 may insert the *semi-supervised* setting where 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. In theory, all the methods, even the unsupervised ones, assume a model of normality. However, with semi-supervised approaches, this model is explicitly discovered through examples. If the training set is completely free of deviating instances (difficult to assume in practice), the semi-supervised setting is equivalent to novelty detection in the exact sense. Since outliers are few in the data set, and recently most of the semi-supervised methods are robust enough to provide good results even with the presence of outliers in the training set, we classify them into the unsupervised methods. Ultimately, what we 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. As mentioned before, the unsupervised setting is the most realistic one when dealing with anomaly detection, especially given the context of the Sea Defender project where labels are not available. Unsupervised methods will therefore be reviewed in the upcoming paragraphs.

There is no unified taxonomy for unsupervised anomaly detection either. In [GU16] for example, the authors make a distinction between nearest-neighbor-based, clustering-based, statistical, subspace-based and classifier-based methods. In [CBK09], the categories are: classification-based, clustering-based, nearest-neighbor-based, statistical, information theoretic and spectral. In [Ruf+21], a contrast is established between classification, probabilistic, reconstruction and distance-based techniques on one hand, and on shallow (viz. non-deep-learning) and deep methods on the other hand. In [LTZ12] the authors consider three sets of approaches: density-based, distance-based and model-based. From our perspective, nearest-neighbor-based, distance-based and density-based methods can be combined, since distances are computed to evaluate densities and they all necessitate distances computations. Clustering-based methods do not belong to the previous group, because in contrast to the previous techniques 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 categories the neural-network-based techniques containing all the deep learning anomaly detection algorithms.

To sum up, we propose to divide the anomaly detection methods into three groups: distance-based methods, model-based methods and neural-network-based methods. A particular focus will be directed toward the most popular techniques of each category. Distance-based methods are reviewed in §1.2.1. Model-based approaches are reviewed in §1.2.2. Neural-networks-based methods are reviewed in §1.2.3.

1.2.1 Distance-Based Methods

All the strategies relying on distance computations to identify anomalies lie in this category. Distances between data points 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) [Bre+00] compares the surrounding density of a data point to the surrounding densities of its k nearest neighbors. The underlying theory of the approach is that those quantities will be approximately the same for an inlier. The surrounding 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:

$$LOF_k(x) = \frac{\sum_{x' \in N_k(x)} \frac{l_k(x')}{l_k(x)}}{|N_k(x)|}, \tag{1.1}$$

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:

$$l_k(x) = \frac{|N_k(x)|}{\sum_{x' \in N_k(x)} max(d(x, x'), d_k(x'))}. \tag{1.2}$$

In Equation 1.2, $d_k(x)$ is the distance D such that there are at least k data points x' for which $d(x, x') \leq D$ and there is at most $k-1$ data points x'' such that $d(x, x'') < D$. 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 [Bre+00]. 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 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, it is stated 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 [Bre+00]. Consequently, the maximum is used in the experiments.

Because LOF uses the Euclidean 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 [GU16]. To solve that issue, Connectivity-based Outlier Factor (COF) [Tan+02] was introduced. COF uses the *chaining distance* instead of the Euclidean distance and computes the outlier scores in a way similar to LOF. The chaining distance of x is the minimum of the sum of all distances from the k neighbors and x. Other variants of LOF have been proposed in the literature, and they are presented extensively in [GU16].

Other distance-based techniques include $k-$nearest-neighbor (k-NN) [RRS00] where the anomaly score of an instance is equal to its distance to its k^{th} nearest neighbor. In [AP02], the anomaly score is the average of the distances from the instance to its k nearest neighbors. With both methods, it is not easy to select an appropriate threshold.

1.2.2 Model-Based Methods

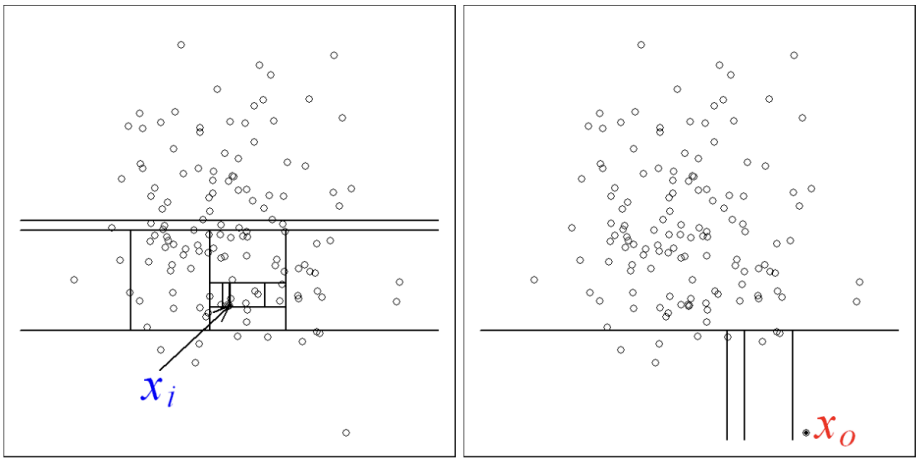
The idea behind clustering methods for anomaly detection is to cluster the data set and then flag as anomalies data points which do not belong to any cluster. To this end, the clustering method must be robust enough to not be sensitive to the presence of outliers in the data set. Outlier-sensitive methods try as much as possible to insert the abnormal instances into clusters, which can lead those to be flagged as normal instances. Sensitive methods can also simply throw away these outliers. Robust methods, like *Find-*

Out [YSZ02], do not force the outliers into clusters. An evident drawback of this model 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 [MLC07] 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 closest to. Moreover, if the instance is located at a distance greater than a predefined threshold from a normal cluster, it is classified as anomalous. To identify anomalies, CBLOF (Cluster-Based Local Outlier Factor) [HXD03] first clusters the data set using a clustering algorithm. Each cluster is either large or small, depending on the fraction (controlled by two parameters) of points of the data set it contains. Then, each instance in the data set is assigned an anomaly score. If the data point belongs to a small cluster, the score is measured by the size of its cluster and the distance between the point and its closest large cluster. If the data point belongs to a large cluster, its score is measured by the size of its cluster and the distance between the point and the cluster it belongs to. More outlier-aware clustering techniques are reviewed in Sec. 2.3.

After projecting the data in a higher dimensional space using a kernel, One-Class Support Vector Machines (One-Class SVMs) [Sch+99] 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 is $+1$ 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. However, because One-Class SVMs, as described in [AGA13], are robust enough to deal with the presence of anomalies in the training data, they are considered unsupervised and outlined here. In [AGA13], 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 classic One-Class SVMs. However, there is an explicit assumption that outliers exist in the data. For Robust one-class SVMs, slack variables already present in the classical One-Class SVMs optimization objective are modified to consider outliers. In η One-Class SVMs, there is an outlier suppression mechanism through the 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) [LTZ12] is based on the idea that outliers are isolated in the feature space. This principle is illustrated on Figure 1.4. Starting from a random sample of the data set, the method randomly selects one attribute $A \in \mathcal{A}$, then randomly selects 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. As a result, each node has two children representing the two subsets obtained after the split. The tree building process is stopped when no partition can be made anymore (when the size of the sample in the node is 1) or when a tree depth limit is reached. A set of trees is constructed in this manner to obtain a forest. After building the forest, every data point passes through every tree in the forest until it reaches a terminal node. Then, the anomaly score of the data point is computed using its average depth in the trees of the forest:



(a) Isolating x_i (b) Isolating x_o

Figure 1.4 – The isolation principle [LTZ12].

Some limits of the Isolation Forest have been highlighted, which has lead to some improvements of the method. One limit, displayed in [HKB21], is the inconsistency of the anomaly scores towards the distribution of the data points in some situations illustrated in the paper. To solve this issue, the authors of [HKB21] propose a variant of the Isolation Forest called Extended Isolation Forest (EIF) which uses hyper-planes with random

slopes instead of axis-parallel separations during the construction of the trees. Similar to the classic 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 hyper-planes had already been explored by SCI-Forest [LTZ10], with a deterministic selection of the split points targeting local clustered anomalies. In [MB21], the tree building process stays consistent with the traditional IF approach, but five new functions to compute anomaly scores are suggested. Similarly, in [Cha+22], the anomaly scores computation of IF is modified. In the latter, the score of an instance is computed in each tree, then compared to the anomaly score threshold and the instance is flagged as outlier or inlier at the tree level. Then, the method called MVIForest (Majority Voting Isolation Forest) flags the instance as outlier if it is outlying in more than half ($half + 1$) of the trees, avoiding the computation of the anomaly score in the remaining trees. In [Cor21], axis-parallel separations are used. The split attribute is selected uniformly at random, while the split value is chosen in a deterministic way by maximizing a pooled information gain metric. The aim of the previous technique is to better identify clustered anomalies, like SCIForest. The goal of Deep Isolation Forest (DIF) [Xu+23] is to identify anomalies which are harder to isolate with linear separations. It first maps the data into new created spaces using casually initialized neural networks. Then, classic IF algorithm is applied in these new spaces. Since the new spaces are non-linear combinations of the original spaces, applying axis-parallel separations in these new spaces is equivalent to using non-linear separations in the original data space. The scores computations are also revisited by adding to each edge a weight representing the deviation between the feature value and the split value in the new space. Other tree-based approaches less similar to IF are available [Guh+16; GSW19]. A more exhaustive review of these techniques can be found in [Bar+22].

1.2.3 Neural-Networks-Based Methods

One of the earliest works on outlier detection with deep learning is [Haw+02] where 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 that have a structure similar to RNNs, are also utilized for outlier detection. Similar to a RNN, an autoencoder takes a data point as input and endeavors to reconstruct it. A first set of layers called the *encoder* transforms the input into another instance with less

features in the space known as *latent space*. Then, another set of layers called the *decoder* tries to transform the lower dimensional data into the original input. During the training, the neural network strives 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 ($x' = x$), with a reconstruction error of 0. The lower dimensional data points are the latent representations of the original ones. Figure 1.5 below shows an example of autoencoder:

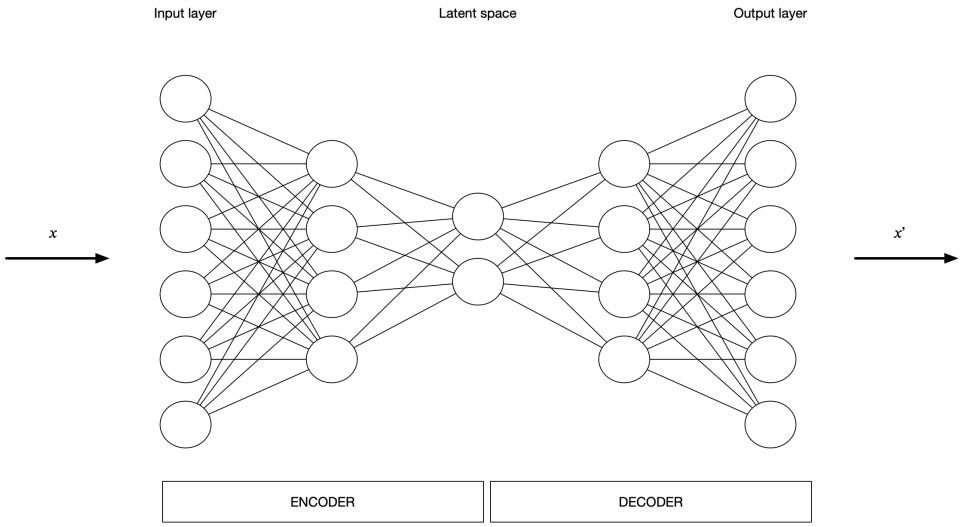


Figure 1.5 – Example of autoencoder: The input space has 6 dimensions ($d = 6$) and the latent space has 2 dimensions.

The usage of autoencoders for outlier detection assumes that outliers will always have a higher reconstruction error than normal data points. This reconstruction error therefore represents a measure of outlierness. This assumption is justified by the fact that the AE learns a model of normality: it should be able to reconstruct perfectly the regular instances, and outliers which deviate from regular data points should be reconstructed poorly. In theory, an AE is a semi-supervised anomaly detection method as it should be trained only with regular instances, to ensure that outliers are easily detected because of their high reconstruction error. In [Alf+20] for example, the authors use an AE for anomaly detection in a semi-supervised way. However, 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 [Che+], an ensemble of AEs, 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 data

set. Finally, the median score over the ensemble is employed as the final anomaly score for an instance. An ensemble of AEs is also used in [CGR20]: 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. Principal Component Analysis (PCA) sometimes also serves that purpose [Qi+18].

Using an ensemble of autoencoders is not the only way to make the architecture robust enough to perform unsupervised anomaly detection. Autoencoder variants like Variational Autoencoders (VAEs) are also employed. In [Ngu+19], the authors use 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, the VAE aims at discovering 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. Then, 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 opposed to each other: a generator and a discriminator. The generator attempts to generate instances which are close to real instances (trying to learn the distribution of the data points). The discriminator tries to distinguish between real instances and fake instances produced by the generator. A GAN is used in [Sch+19] in combination with an AE to detect anomalies in medical images in a semi-supervised way. Another AE variant, an Adversarial Autoencoder (AAE) is used in [Raj+19] to detect anomalies in wireless spectra. An AAE 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 AE 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 section. More detailed surveys can be found in [Pan+21] and [Ruf+21].

1.2.4 Discussion

Neural-network-based methods are suitable for the identification of anomalies in complex data types like images, graphs and sequence data (time series, videos, audios, text). Even though more classic approaches have been utilized, neural networks remain appealing as they are able to automatically capture complex relationships in the data. Graph anomaly detection, which is now mainly dealt with Graph Neural Networks, is particularly challenging, especially because of the abundance of anomaly definitions for graphs [Ako21]. For more classical data like tabular data, the time and effort put in the training of a neural network can be disheartening, in particular in the absence of labels. Neural networks often require a large training data set and possess an important list of (hyper)parameters to set in comparison to other methods. Furthermore, the training time is not insignificant, although there are specialized ecosystems for neural networks training nowadays. Distance-based methods require distance computations which are time consuming. Even when the distances are only computed between neighboring data points and some pruning strategies are employed, the neighbors have to be identified and the size of the neighborhood is generally a crucial parameter of the method. Plus, if we do not have tabular data, the classical Euclidean distance may not be suitable anymore, and an appropriate distance metric must be selected or designed, 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. The choice of the number of clusters in clustering-based methods is of paramount importance. However, clustering methods are able to discover different types of anomalies in the data. IF is one of the most effective methods, while being simple to use and having few hyper-parameters to set.

Most of the approaches listed in this section are only devoted to anomaly detection. 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.

1.3 Anomaly Explanation

An anomaly detection algorithm just tells for each data point if it is abnormal or not, sometimes with a score indicating a deviation degree. 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.

In this section, we first propose a taxonomy of anomaly explanations according to the information conveyed by the explanation (§1.3.1). We then position our taxonomy in relation to the state of the art (§1.3.1), before reviewing the existing anomaly explanation approaches (§1.3.2 to §1.3.5).

1.3.1 Taxonomy of Anomaly Explanations

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 establish difference. This group of pixels can represent the whiskers of the cat on each image for example. The user will then 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 that have contributed most to the decision of an algorithm is a good start and a classic method to provide explanations. Anomaly detection is no exception to the rule. In Figure 1.6 below, to mark the square data point as anomalous, one can look only at the feature A_1 for all the instances: in comparison to the regular data points in blue for which the values on the attribute A_1 vary between 3 and 5, it takes the value 6 on A_1.

The same cannot be told for the feature A_2 since the square instance has a value of 2 on that attribute, which is normal when comparing it with the values on A_2 of the regular instances. As a result, to explain that anomaly to the user, we can just say that attribute A_1 contributed to the abnormality of the square data point. This first category of anomaly explanation is **feature importance**.

Stating which features are important is sometimes insufficient. In Figure 1.7, when trying to explain the abnormality of the square instance using feature importance, we note that both features have equal importance, because no feature helps identifying the anomaly better than the other. The anomalous instance has a regular value for each of the features taken independently. It is the combination of the values for both attributes

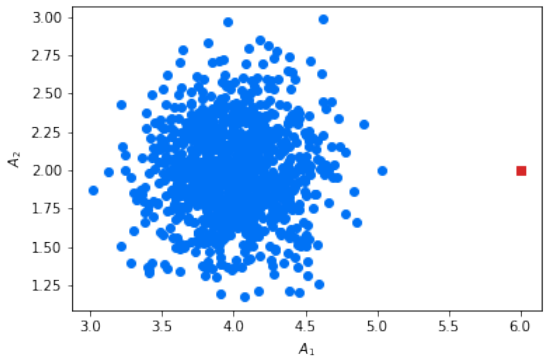


Figure 1.6 – Anomaly explanation by feature importance: attribute A_1 helps us to tag the square data point as anomalous

which makes the data point irregular. In this setting, explanation by feature importance returns the two attributes, which provides no information at all. In two dimensions, like in our examples, it is easy for the user to plot the data. However, if we are in higher dimension, which is generally the case, displaying a list of features with more than two having the same importance is not really helping the user.

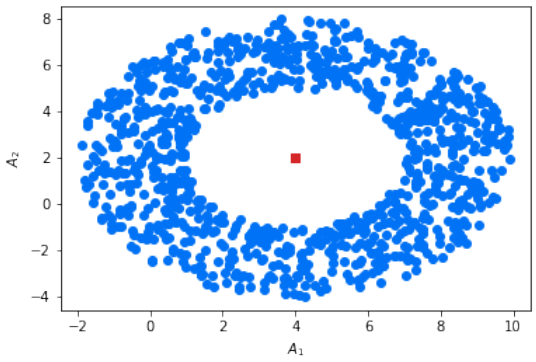


Figure 1.7 – Anomaly explanation by feature values: the square data point is anomalous because $A_1 = 4$ and $A_2 = 2$, and that combination of values is abnormal.

It would have been clearer to the user to say, for instance, that the data point in Figure 1.7 is anomalous because it has a value on the attribute A_1 around 4 and a value on the attribute A_2 around 2, or more generally because $1 \leq A_1 \leq 6$ and $-1 \leq A_2 \leq 5$. 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 there are several conditions on the features.

In addition to that, with the two previous categories of explanations, only information regarding the anomaly is provided. We do not know what is the difference between anomalies and regular data points. With the example in Figure 1.7, after discovering that the instance is anomalous because $1 \leq A_1 \leq 6$ and $-1 \leq A_2 \leq 5$, the user can wonder if a data point with $A_1 = 8$ and $A_2 = 8$ is an outlier (without plotting the data set of course). Explanations by feature importance and by features values do not provide an answer to this question. A response would be delivered if the anomaly was explained by directly comparing it to regular data points. Even if explanation by feature values returns regions containing anomalies, it could be more interesting to have a link with the regular instances, since it has been shown that people often prefer *contrastive* explanations [Mil19]. 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 regular instances. This third category of explanations will be called **anomaly explanation by data point comparisons**.

With a data set like the one in Figure 1.1, where there are multiple clusters and local anomalies, the previous types of explanations show their limits. With feature importance for example, the data points x_2 and x_4 receive the same explanation: attribute A_1 is responsible for their outlierness. When employing feature values, an explanation for x_4 can be $A_1 \geq 7.5$ and $A_2 \geq 6$. With data points comparisons, an explanation for x_4 is the closest instance located at $(7.6, 11)$. However, an important information is missing with all the aforementioned categories: the locality of the anomalies. x_2 and x_4 are local anomalies to two different clusters of regular instances. Explanations by data point comparisons return the regular data points the closest to x_2 and x_4. These data points belong to two distinct clusters and that information is lost. The most complete explanation for x_2 would be that it is an anomaly for the green (squares) cluster because the value of A_1 is too high. x_4 is an outlier, this time for the blue (circles) cluster for the same reason.

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 is called **explanation by structure analysis**. It starts at the anomaly detection level by identifying anomalies local to each group of regular data points. This is the type of explanations needed for the Sea Defender project, because there are categories of products that need to be identified through clustering.

To sum up, we found out that existing anomaly explanation approaches may be organised into the following four categories:

— explanation by feature importance,

— explanation by feature values,

— explanation by data point comparisons,

— explanation by structure analysis.

In the literature, the most recurrent categories of anomaly explanation methods are *model-agnostic* vs *model-specific*, and *local* vs *global*. 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 as a whole, or why a group of outliers is abnormal. In [Pan+22], the proposed categories are: methods that rank anomalies, methods that reveal causal relationships between anomalies, and methods that identify the attributes responsible for the abnormality of points or groups of points. Method that rank anomalies cannot be considered as an explanation in our opinion, because the anomaly score is just an indication of how much the instance is deviating, not *why* the instance is deviating. Methods that reveal causal relationships between anomalies focus on time-series anomaly detection and can be categorized as data point comparisons. The last type of approaches is equivalent to explanation by feature importance. In [LZV23], the taxonomy is based on six criteria. The first two distinctions are related to the locality (local vs global) and the specificity (model-agnostic vs model-specific) of the method. The third distinction is related to the data type, while the fourth is related to the process time of the explanation (pre-model, in-model and post-model). Pre-model techniques are constructed and implemented before the anomaly detection process [LZV23]. It is therefore hard to consider them as explanation methods. The last two criteria are the data perspective (feature-based, sample-based, feature&sample-based) and the techniques (approximation-based, perturbation-based, gradient-based, visualisation-based, etc). The distinction between techniques is probably too finely tuned, as it is not exhaustive. The feature-based category brings together feature importance and feature values methods. Sample-based approaches compare data points. Explanation by structure analysis is closely related to feature&sample-based methods which combine the two former aspects. While the existing taxonomies are centered around the method generating the explanations, our taxonomy is focused on the nature of the generated explanations.

To illustrate the remainder of this section, we will consider the following example: in Table 1.1, 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 1.2.

The latter corresponds to the real properties of the products. This a simpler version of the kind of data to analyze in the Sea Defender project.

Table 1.1 – Running example: list of products

ID	**Brand**	**Model**	**Unit weight(g)**	**Unit price(€)**
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 1.2 – Running example: true characteristics of the products

Brand	**Model**	**Unit weight(g)**	**Unit price range(€)**
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, the anomalies are:

— the product *2* because of its low price,

— the product *3* because of its high weight and low price,

— the products *6* and *9* because of their high weight and high price,

— and the product *7* because of its low weight.

We now show that existing anomaly explanation approaches can be inserted in one of the categories of our taxonomy.

1.3.2 Anomaly Explanation By Feature Importance

A distinction is made between the methods which identify the important features without further details, and the methods weighting the features or providing an ordering of the features according to their importance.

Non-weighted Feature Importance

Definition 7: Explanation by non-weighted feature importance

An explanation by non-weighted feature importance is a subset of features $\mathcal{E} \subset \mathcal{A}$ containing the attributes that contributed to the identification of the instance as an anomaly.

The earliest work on anomaly explanation is a non-weighted feature importance approach. In [KN99], the authors identify outliers in subspaces of the attribute 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 as mentioned in Sec. 1.1. 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 [KN99], some methods also explain anomalies by finding the set of features that isolates them. In [Mic+13], the authors explain a given anomaly by identifying the subspace of features that best separates that outlier from the rest of the data set. More generally, anomaly detection methods which identify outliers in subspaces of the original feature space like Subspace Outlier Degree (SOD) [Kri+09], or in subspaces of a transformation of the original feature space like Correlation Outlier Probability (COP) [Kri+12] 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 [Gup+19] 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 [Liu+20], 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 [Liu+20] propose a method to explain clusters of outliers, clusters based on the behavior 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 data set.

More generally, there is a set of data mining methods called Group Outlying Aspects Mining (GOAM) which try to identify the features making 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. An extensive exploration of GOAM is provided in [Wan+18].

In [Qi+18], the authors explain anomalies in images using metadata. Anomaly detection is first performed using PCA. 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 identifi-

cation 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 [PW10]. 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) [Sid+19], 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, the two first features are used and so on, until the data point is found outlying using a sequence of features. The explanation for the outlier is 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.

Weighted Feature Importance

Definition 8: Explanation by weighted feature importance

An explanation by weighted feature importance is a couple $(\mathcal{E}, w_{\mathcal{E}}) \in (\mathcal{A} \times \mathbb{R})$. $w_{\mathcal{E}}$ quantifies the importance of feature $\mathcal{E}$ in the identification of the anomaly.

Local Outlier Detection with Interpretation (LODI) [Dan+13] and Local Outliers with Graph Projection (LOGP) [Dan+14] 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 introduced before. However, LODI and LOGP provide weights quantifying the importance of each identified feature.

SHAP (SHapley Additive exPlanations) [LL17] 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* (Tables 1.1 and 1.2). 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 [ASR19], 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 [GS19] produce similar explanations to time series anomalies using an extension of Kernel SHAP, the anomalies having been identified by a GRU-Autoencoder (Gated Recurrent Unit). 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 are also exploited for anomaly explanation by feature importance in [Tak19], but using PCA as anomaly detector. In [TK20], the computation of the Shapley values is generalized to provide explanations to any semi-supervised anomaly detector.

DIFFI (Depth-based Feature Importance for the Isolation Forest) [CTS23] 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. It then aggregates the scores to compute the feature importance scores for the whole forest. In addition to these global 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 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 [Ngu+19], 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 [KMM20], the authors convert One-Class SVMs models into a neural network 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. It is a local, model-specific explanation method. Layer-wise Relevance Propagation is also used in [AKM18], although anomaly detection is performed in a supervised way using a neural network;

— in [Bro+18], attention mechanism is used with a Long Short-Term Memory (LSTM) neural network 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) [Zha+19] is a model-agnostic method close to LIME [RSG16] 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.

1.3.3 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 $A\delta v$ where A is a feature, δ is one of the signs $<, \leq, =, >, \geq, \in, \subset, \notin ...$ and v is a feature value. As an illustration, an explanation by feature values of outlier 9 can be: $unitweight \geq 210$ and $unitprice \geq 300$.

Definition 9: Explanation by feature values

An explanation by feature values is a logical formula specifying the bounds of the regions in which the anomaly/anomalies is/are found: $\mathcal{E} = \wedge_p A_p \delta_p v_p$.

In [Bas+16] 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. The human analyst can then throw out the least interesting ones in order to prune the decision trees, so that only relevant anomalies could be identified later.

In [BCB22], 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. Noteworthy is 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. With x-PACKS [MA18], a subspace clustering is first performed on the data set containing anomalies and normal data points. Subsequent to that phase, hyper-rectangles containing the maximum number of anomalous data points and the minimum number of regular instances are obtained. Then, each hyper-rectangle 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.

In [Son+18] the authors perform anomaly detection using an 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-understandable.

The Explainer [KPH20] 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.

1.3.4 Anomaly Explanation By Data Point Comparisons

Angle-Based Outlier Detection (ABOD) [KSZ08] is an unsupervised anomaly detection method providing explanations. To detect outliers, the algorithm will compare the variance of the angles between data points. The hypothesis is 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 [KSZ08]. To give explanations on why an instance is outlying, ABOD finds its closest neighbor in the nearest cluster, then computes and returns the difference vector between the two data points. 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 [Mok19], only the closest neighbor is used for the explanation. The other instances in the data set could contain more insights on why a given instance is anomalous.

In [RL09], 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.

Kernel-based Supervised Hashing (KSH) [Li+18] constructs a group of hash functions which map the original data points to lower dimensional expressions in a hash code space. To build the hash functions, KSH uses a labelled training set. Data points having the same label are similar/neighbors in the hashing space. To find out if a given data point is anomalous or not, KSH searches 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 [Mia+19], 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/she 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 [SRC19], anomaly detection is performed in a semi-supervised way using *GANomaly* [AAB19] which consists of a GAN whose generator is an AE 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.

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* from our running example will indicate that the

unit price must be increased by 200 to obtain a regular instance. Counterfactual explanations in the context of anomaly detection are explored in [HJS21]. The authors generate counterfactual explanations with an AE-based anomaly detection.

Definition 10: Explanation by data point comparisons

An explanation by data point comparisons is a set of representative instances defined in the same space as instances from $\mathcal{D}$:

$$\mathcal{E} = \{x \in dom(A_1) \times dom(A_2) \times ... \times dom(A_d)\}$$

$x \in \mathcal{E}$ can be a neighbor of the anomaly to explain, the difference vector between the anomaly to explain and a regular instance, the average of the regular instances in the data set...

1.3.5 Anomaly Explanation By Structure Analysis

This last category of explanations takes into account the structure of the data set. An explanation is the set of properties shared with a regular pattern, and the set of properties deviating from this pattern. Analyzing the structure means discovering in the data set clusters of regular data points and instances which deviate from each group. In the example from table 1.1, 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 1.1, high-end products can be those with a true price range in the interval $[800 - 1500]$, entry-level products are those whose true prices range from 100 to 400 and, mid-range products are those for which $unitprice \in [450-700]$. With this breakdown, an explanation by structure analysis for the outlier *2* is that according to its *unitprice* it is a mid-range product, but it is not an inlier because products of this model are supposed to be high-end products. This kind of explanations can be provided by scrutinizing thoroughly (possibly manually) the detected anomalies, but the goal is to simplify the process as much as possible, for users 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.

Definition 11: Explanation by structure analysis

Given a partition of $\mathcal{D}$ into k clusters ($\mathcal{D} = \mathcal{C}_1 \cup \mathcal{C}_2 \cup ... \cup \mathcal{C}_k$), an explanation by structure analysis is a pair $(\mathcal{E}^{shared}, \mathcal{E}^{dev})$.
$\mathcal{E}^{shared}$ (resp. $\mathcal{E}^{dev}$) is the set of properties shared with each cluster (resp. deviating from each cluster): $\mathcal{E}^{shared} = \{\mathcal{E}_1^{shared}, \mathcal{E}_2^{shared}, ..., \mathcal{E}_k^{shared}\}$ and $\mathcal{E}^{dev} = \{\mathcal{E}_1^{dev}, \mathcal{E}_2^{dev}, ..., \mathcal{E}_k^{dev}\}$, where each $\mathcal{E}_i$ can take one of the forms in Definitions 7 to 10.
$\mathcal{E}_i^{shared}$ is the set of properties that the anomaly shares with the cluster $\mathcal{C}_i$. $\mathcal{E}_i^{dev}$ is the set of properties making the anomaly deviate from the pattern represented by the cluster $\mathcal{C}_i$.

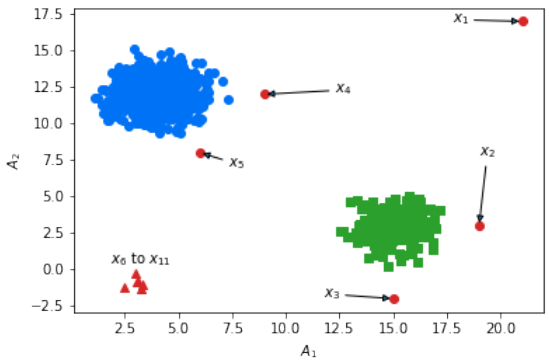


Figure 1.8 – A data set (same as in Sec. 1.1)

A partition of the data set on Fig. 1.8 is $\mathcal{D} = \mathcal{C}_1 \cup \mathcal{C}_2$, where $\mathcal{C}_1$ is the blue (circles) cluster and $\mathcal{C}_2$ is the green (squares) cluster. An explanation by structure analysis of the anomaly x_4 can be the pair $(\mathcal{E}^{shared}, \mathcal{E}^{dev})$, with $\mathcal{E}^{shared} = \{\{A_2\}, \emptyset\}$ and $\mathcal{E}^{dev} = \{\{A_1\}, \{A_1, A_2\}\}$. In this example, each $\mathcal{E}_i$ is a non-weighted feature importance (Definition 7). x_4 shares the attribute A_2 with the points of C_1, while A_1 makes x_4 deviate from C_1. x_4 does not share any attribute with C_2. In contrast, A_1 and A_2 make x_4 deviate from C_2.

In [Mej10], clustering is used to detect anomalies. After the clustering, the 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 produced. The most dissimilar pairs of clusters can also be returned with the percentages of differences between features. In [Shu+20], the authors derive a similarity measure from an IF. A clustering (or more precisely an Agglomerative Hierarchical Clustering) of the regularities and the outliers is then performed based on the similarity measure defined. After that, each abnormal cluster is compared to regular clusters based on their distinctive properties. Ultimately, linguistic summaries, describing not only the properties of each cluster but also the differences between clusters, are generated.

The works that most closely belong to this category are the model-agnostic methods COIN (Contextual Outlier INterpretation) [LSH18] and ATON (Attention-guided Triplet deviation network for Outlier interpretatioN) [Xu+21]. COIN first identifies the nearest regular neighbors of the outlier to explain. Then, these nearest neighbors are clustered. After that, synthetic sampling is employed to expand the outlier to an anomaly class. A set of classifiers are later trained to draw a linear boundary between the outliers and each cluster of regular instances. The weight of each attribute is finally computed by aggregating the features importance from each linear classifier. In addition to feature importance scores, COIN returns as explanation the set of nearest neighbors and an anomaly score. With ATON, a set of triplets is first generated. Each triple is composed of the outlier to explain and two random regular instances: one from the data set and one from the set of neighbors of the outlier to explain. Then, the original feature space is transformed into a new space. The feature mapping function is linear and attention is attached to each embedding dimension. The separability between the outlier and the normal instances in each triple is then learned. After that, the importance weights of each original features are derived from those of the computed features. ATON also proposes a threshold setting approach to convert weighted feature importance into non-weighted feature importance. In contrast to the previous approaches, ATON and COIN acknowledge the potential locality of anomalies when generating explanations, through the use of nearest neighbors. COIN even goes one step further by clustering these neighbors. Nevertheless, the explanations ultimately produced are not really cluster-specific. In addition to that, they do not tell anything to the user about the regular patterns in the data set.

1.3.6 Discussion

Explanation by feature importance is the most widely researched. 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 weights indicating the importance of each feature, a pair of features or a list of feature pairs, or a plot picturing 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 feature subspaces that best separate outliers and normal data points is relatively 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 can be leveraged. 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. However, 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 [Ngu+19] where the authors propose a clustering of the anomalies based on the feature gradients to identify the types of anomalies located in the data set. However, anomaly explanation by feature importance remains 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 data set or to avoid the leak of sensitive information.

The output of an anomaly explanation by feature values method is typically a set of rules on the features. It can also be a text in natural language, like in [Mun+19] 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. These parts are then checked against some predefined rules. The 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. 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 SVMs, 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.

The possible outputs of anomaly explanation by data point 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 suitable for distance-based anomaly detection methods. Since the latter already requires 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 necessitate distances computations. More generally, it can be employed with any anomaly detection algorithm. The difficulty of use comes from the choice of an appropriate distance/similarity metric, which may turn out to be complicated with complex data types. Even if the data type is not complex, explanation by data point comparisons requires finding 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 generate explanations. Ultimately, displaying similar instances and showing the differences between the anomalous instance and these similar instances allow the user to discern clearly why a data point is irregular. However, this type of explanations just provides a restricted overview of the data set.

Explanation by structure analysis provides insights about the abnormality of an instance in relation to the global structure of the data set. As it was not heavily investigated in the literature, the output can ideally be a set of discriminating features with each group of regular data points, or even a set of rules, in addition to the set shared features. This kind of explanations is suitable for model-based anomaly detection methods, especially

cluster-based methods which provide a partition of the regular instances. It is more difficult to extract explanations by structure analysis with neural-network-based anomaly detection methods since the structure of the data set is not really analyzed when using neural networks. Providing this kind of explanations starts at the anomaly detection level with the identification of clusters of regular instances and local anomalies. The approaches labeled as belonging to this category are either incomplete or a sequence of steps (anomaly detection → clustering → structure analysis of the clusters → explanations generation) often relying on external methods. No method in the literature has been able yet to provide a unified algorithm going directly from the detection to the detailed explanations. Because of the context, we will focus on this last category. However, it could be interesting to compare the different categories according to the needs of the users. The example from Tables 1.1 and 1.2 is simple. In the Sea Defender Project, there are more types of products, hence the need for an automated framework with the explanatory component to increase the trust of the customs officials.

1.4 Outlier-Aware Clustering

Most clustering algorithms suffer from the presence of outliers: the points that to do not conform with the global structure of the data most often hinder the identification of regular clusters. *Robust clustering* methods aim at addressing this issue, providing data partitions that are not perturbed by outliers. They aim at outputting the same results as would be obtained if the outliers had been removed from the data set, without requiring to perform a preliminary step of outlier detection and removal.

Robust clustering can be roughly categorized into two types of methods [Bor+15]. Some methods proceed by automatically down-weighting atypical points. This is the case in [Dot+18] where the initial proportion of outliers in the data set is controlled by a fixed trimming level. This initial trimming level is set to a high value (thus discarding a lot of data points), and clustering is applied. Then, the partition is refined by decreasing the trimming level through the inclusion of data points close to cluster centers. In [CG13], the farthest points are discarded during the update of the clusters centers in the $k-$means algorithm. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [Est+96] is a robust clustering method classifying data points into three categories: *core* points, *border* points and anomalies. Core points are points containing at least $minPts$ in their ϵ neighborhood. The border points contain at least one core point

in their neighborhood. The other data points are anomalies. The clusters are obtained by combining *density-reachable* points. Density-reachable points are points located in the neighborhood of core points.
While fuzzy clustering may assign to outliers equal membership to the different clusters, possibilistic clustering [KK96; Pal+05] makes a clearer contrast between abnormal and regular instances. In [MP98], normal distributions are replaced by multivariate t-distributions which have longer tails to make the clustering more robust. Noise clustering [Dav91] assumes that all the outliers can fit in a cluster. A characterization of the points belonging to the noise cluster is therefore required. Other methods, instead of adjusting weights, propose to replace the classic squared Euclidean distance, which is known to be highly sensitive to outliers, by other distances [Seh+89; Jaj91; GJ01; RB99; FK96].

Although robust clustering methods are also able to identify abnormal instances in the data set, they remain clustering methods. They are not the foremost thought when it comes to identifying deviating instances. This is due to the fact that anomaly detection is not their main focus but a secondary mission. Furthermore, robust clustering often necessitates distance computations between data points, while there are anomaly detection methods that do not.

1.5 Summary

In this chapter, we explored the background on anomaly detection, anomaly explanation and robust clustering. In Section 1.1, some concepts are recalled. In Section 1.2, we reviewed existing works on anomaly detection. In Section 1.3, we first introduced our taxonomy of anomaly explanation approaches, before reviewing existing works. Section 1.4 finally reviewed existing works on robust clustering. Our proposed taxonomy of anomaly explanation methods contains four categories of approaches: explanation by feature importance, explanation by feature values, explanation by data point comparisons and explanation by structure analysis. While the first two categories focus on the anomaly to explain, explanation by data point comparisons introduces a contrast between the anomalies and the regular instances. The last category provides even more contrast by analyzing the structure of the data set (in terms of clusters) and generating contrastive explanations with respect to these clusters. There are not many approaches in the literature falling in that category. However, it is the type of explanations needed in our context and is therefore the focus of the work presented in this dissertation.

The next chapter will present our approach to produce explanations by structure analysis.

Chapter 2

CADI: Contextual Anomaly Detection with Isolation-Forest

This chapter introduces our proposal to anomaly explanations by structure analysis. Although techniques like COIN [LSH18] and ATON [Xu+21] consider the local context of the outlier when generating explanations, this local context is either forgotten in the explanations produced (ATON) or not specific enough (COIN). In this latter method, if an outlier to explain is really close to one cluster, there is no information on the other clusters in the data set. Furthermore, COIN relies on distance computations for the nearest neighbors identification and on an external algorithm for clustering. Both ATON and COIN are model-agnostic. In contrast to these approaches, we propose a unified model-specific method to generate explanations by structure analysis. This method, called CADI, stands for Contextual Anomaly Detection with Isolation-Forest, performs anomaly detection, data clustering and finally generates data-structure-aware explanations of the anomalies.

To perform anomaly detection and data clustering using the same technique, there are two options. The first one is to use a robust clustering method. The second one is to extend an anomaly detector for data clustering. Employing a robust clustering method entails that, within the prioritization structure, the clustering task takes precedence, followed by the anomaly detection task. The main objective of robust clustering is to perform clustering. It is not the case here. Anomaly detection is still the primary focus, while clustering is a means to extract explanations by structure analysis. Even if there are no clusters in the data set and we are not able to extract explanations, the method should at least be able to identify anomalies. As a result, the second path, namely utilizing an anomaly detector for data clustering, is the one we chose.

> Which anomaly detector to choose as the backbone of our unified method?

Distance-based methods are good candidates. They already rely on distance computations between pairs of points. These distances could later be employed to identify clusters in the data set. Model-based methods are also good candidates. Neural-network-based methods are perhaps the least interesting candidates. They are often too complex and it would therefore be difficult to recover the structural information of the data set as this structural information is often lost during the anomaly identification. Furthermore, neural networks are black-boxes. Explanations are generally provided by a post-hoc process (e.g.: LRP or converting the neural network into a decision tree). There are more interpretable anomaly detectors in the literature. Why explaining a black-box when we can use an interpretable model, especially when the data can be handled relatively easily by a shallow (in contrast to deep as in DL) method? [Rud19].

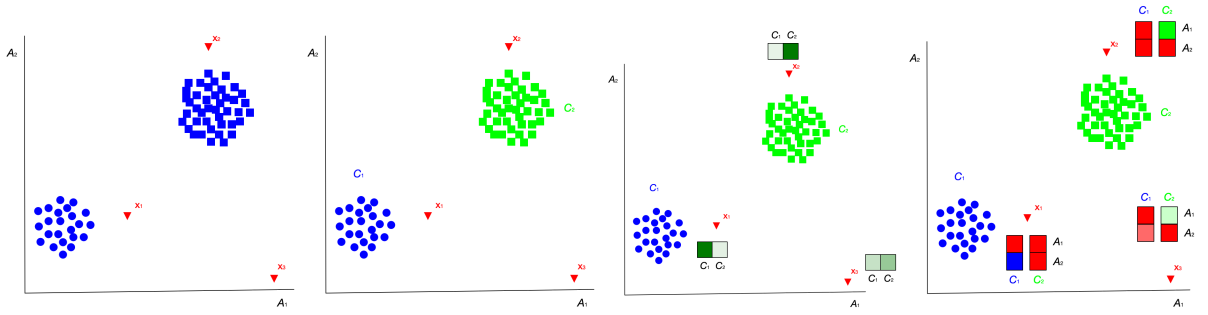
Among the model-based methods, the Isolation Forest algorithm is very appealing for anomaly detection. It is unsupervised, fast, has few hyper-parameters and is interpretable at the tree level. IF also makes no assumption regarding the distribution of the data set. In addition to that, the efficiency and the robustness of the approach against the choice of the hyper-parameters have been confirmed throughout the years by different benchmarks [Han+22; Cha+23]. Several extensions of IF have been proposed in the literature [LTZ10; HKB21; Cor21; Xu+23], but the performance of the seminal approach especially considering its low complexity remains astonishing. This motivates us to select IF as the backbone of CADI. The details of our approach are presented in this chapter.

The content of Sec. 2.2 has been published in the proceedings of EGC 2023 [Yep+23]. An extended version of the previous paper is under review for a special issue of the journal DKE (best papers of EGC 2023). The remainder of the chapter is published in the proceedings of SAC 2024 [Yep+24].

Contents

2.1 Overview

CADI performs anomaly detection, data clustering and generates anomaly explanations by structure analysis in a unified framework based on IF. The principle of IF is recalled in Sec. 2.2. In contrast to the original IF where the separations are completely random, CADI introduces a split selection criterion. The goal of this criterion is to avoid splitting dense regions which most likely contain portions of clusters. These portions of clusters are later combined to obtain a partition of the regular instances in the data set, after the anomalies have been identified. Once the reconstruction of the regular data inner structure done, anomalies are localized and explained by harnessing the trees from the same forest. The attributes shared by the anomaly and the points in each cluster, as well as the attributes making the instance deviating from each cluster, are identified. Figure 2.1 illustrates the CADI framework.



(a) Anomaly detection (b) Clusters identification (c) Anomaly localisation (d) Anomaly explanation

Figure 2.1 – The CADI framework

Section 2.2 details step a of Fig. 2.1. Sec. 2.3 details step b and Sec. 2.4 details steps c and d.

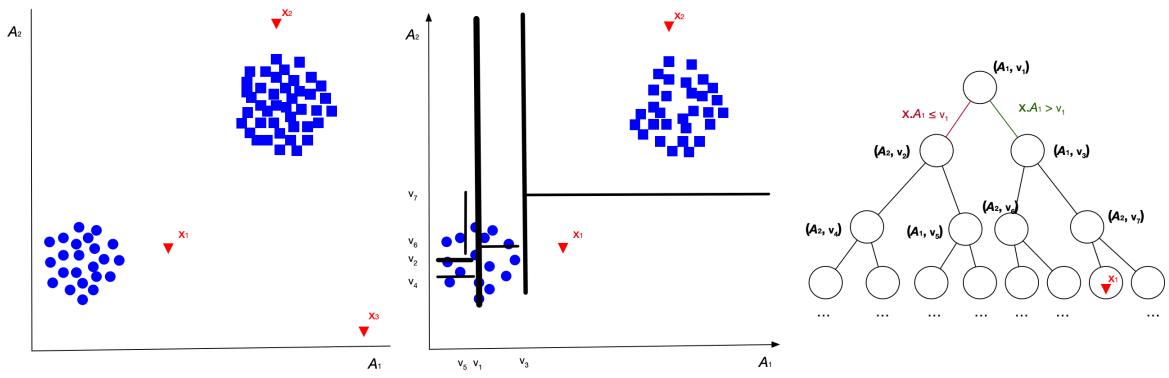
2.2 Density-Aware Isolation forest

In the initial IF approach, an isolation tree is built through recursive splits of a data set sample D. A split is a couple (A, v), where A is a randomly chosen attribute $A \in \mathcal{A}$ and v a value from its observed domain $v \in [\min_{x \in D} x.A, \max_{x \in D} x.A] \subseteq dom(A)$. Algorithm 1 recalls the tree construction process of IF.

Figure 2.2 shows the isolation process on a sample of a data set.

Algorithm 1 Isolation Forest : *build_tree* [LTZ12]

Inputs: a sample $D \subset \mathcal{D}$, the depth d of the current node; $d = 0$ when the method is first called
Output: a node in an isolation tree
if $|D| = 1$ or $d \geq h_{lim}$ **then**
    Return $node(null, null, D, d, null, null)$ ▷ Leaf (terminal node)
else
    $A \leftarrow random(\mathcal{A})$ ▷ Random attribute selection
    $v \leftarrow random(dom(A))$ ▷ Random value selection
    $D_l \leftarrow \{x \in D / x.A \leq v\}$
    $D_r \leftarrow \{x \in D / x.A > v\}$
    Return $node(build\_tree(D_l, d+1),$ ▷ Internal node
                $build\_tree(D_r, d+1), D, d, A, v)$
end if



(a) A data set (b) Random sub-sampling and tree construction

Figure 2.2 – The isolation process

An IF is a set of t trees built on different random samples of size Ψ of the data set. Algorithm 2 recalls the forest construction process of IF.

Algorithm 2 Isolation Forest : $build\_forest$ [LTZ12]

Inputs: a data set $\mathcal{D}$, a number of trees t, a sampling size Ψ, a depth limit h_{lim}
Output: an Isolation Forest $\mathcal{F}$
$\mathcal{F} = \emptyset$
for $i = 1$ to t **do**
    $D = sample(\mathcal{D}, \Psi)$ ▷ Sampling
    $\mathcal{F} = \mathcal{F} \cup build\_tree(D)$
end for
Return $\mathcal{F}$

An issue with IF in the prospect of reconstructing the structure of the regular data points is that the separations are completely random. Nothing prevents points from different clusters to be found in the same leaves in the trees. The only information that can be derived from a forest is the ease of isolation of an instance, measured by its average isolation depth in the trees. If an instance is not easily isolated, then it is a regular instance. However, in order to partition the data, information about the proximity of the instances is needed. An IF does not provide such information. Consequently, leveraging a classic IF for data clustering would not be straightforward. It would require, for example, to derive a similarity measure from an IF. This similarity could be computed between pairs of points like in [Shu+20], which would be unfortunate since the original method does not compute pairwise distances and it is one of its strengths. As a result, we propose to revisit the classic IF approach in order to have more information on the proximity of data points using a built forest, without computing pairwise similarity between instances. This information about the proximity of instances would be provided by the leaves of trees. More precisely, the goal is to have in the same leaf, data points which are close in the original data space. A partition of the regular data would then be obtained by combining leaves, and not data points. In order to have leaves containing close data points, the split selection is revisited.

2.2.1 Forest construction

CADI revisits the completely random process of IF to keep only the splits that fall in sparse regions. Anomalies being by definition detached from regular phenomena materialized by dense regions, the objective of this revisited isolation algorithm is to find

separation lines in sparse areas surrounding dense regions. To do so, a density-based constraint is added and determines if a split is actually performed or discarded. The hypothesis is the following: if a significant number of points are found in the neighborhood of the split, it is potentially separating a cluster.

Hypothesis 1

If a significant number of points are found in the neighborhood of a split, it is potentially separating a cluster. And, by opposition, an informative split should separate dense regions from isolated points.

In that case, the split is discarded and another one is generated. The goal is to surround the regular point clusters by the separations, so that dense regions of points remain unseparated during the tree construction. One hyper-parameter is introduced in addition to the IF hyper-parameters: the size of the margin around the separation which represents its neighborhood. This margin size is controlled by the hyper-parameter α which is a percentage of the attribute range of the points in the sample. Figure 2.3 depicts an example of split selection. The margin around the separation (A_1, v_1) contains many data points. This split is therefore discarded. In contrast, the area surrounding the split (A_2, v_2) is sparse, so the split is selected.

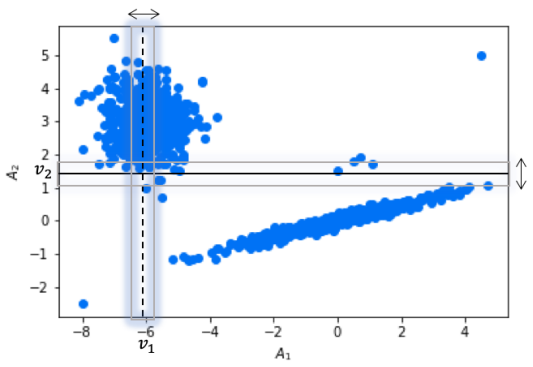
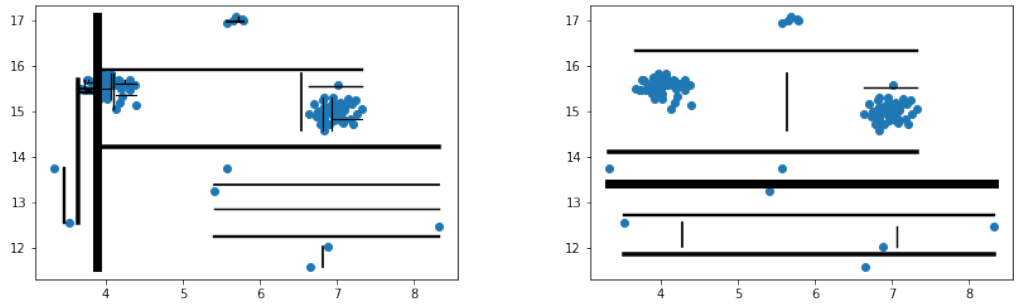


Figure 2.3 – Example of a discarded separation line (A_1, v_1) falling in a dense area (dashed line) and a validated separation (A_2, v_2) (plain line)

Figure 2.4 illustrates the impact of the split selection criterion of CADI on the isolation procedure. With the proposed criterion, the separations less often separate points belonging to the same cluster, and the anomalies remain isolated.



(a) Splits of a classic IF tree

(b) Splits of a CADI tree

Figure 2.4 – Examples of splits, shown by the black lines, of a tree: (left) IF, (right) CADI. The width of the line is inversely proportional to the depth of the split.

In the original IF method, sampling is performed. This sampling which is completely random allows to build a tree on a smaller, but still representative, portion of the data set. Building trees on different random samples and combining the information from the trees at the forest level provides the opportunity to partially scan the data set, while maintaining a good complexity and reducing memory cost. Sampling also lessens the impacts of *swamping* and *masking*. Swamping occurs when the number of normal instances increases or they become more scattered. Masking occurs when there are too many anomalies in the data set which therefore conceal their own presence [LTZ12]. CADI also employs sampling. Consequently, some splits may still separate points belonging to the same cluster, despite the density constraint. In this case, points forming a cluster may be found in different leaves of the trees. These leaves have to be combined to reconstruct the whole data inner structure.

Alg. 3 details how density-aware isolation trees are constructed.

To learn from the discarded splits and to avoid generating separations in intervals that have already been discarded -because they contain many data points-, the set of tested intervals I (I^A being the union of intervals on attribute A) is stored and passed as a parameter through the recursive calls to the *build_tree* function (line 20 in Alg. 3). If the method was not able to find a valid separation in the whole interval of values of an attribute (line 10), this attribute is discarded (line 11). The discarded attributes are therefore also stored (in the variable C). A separation is kept when the number of points in the margin is less than the number of points which would fall in the margin if the distribution was uniform (line 14). If the method is unable to find a valid separation on

Algorithm 3 CADI: *build_tree*

Inputs: data subset $D \subset \mathcal{D}$, depth d of the current node, margin width percentage α, sets of tested intervals $I = \{I^{A_1}, \ldots, I^{A_m}\}$, set of covered attributes C, margin widths $margs$; I and C are empty when the method is first called, $d = 0$, $|D| = \Psi$ (sample used to build the tree) and $margs[A] = \frac{1}{2}\alpha(\max_{x \in D} x.A - \min_{x \in D} x.A)\ \forall A \in \mathcal{A}$
Output: a node in an isolation tree
if $C = \mathcal{A}$ or $|D| = 1$ or $d \geq h_{lim}$ **then**
▷ returns a leaf
    Return $node(null, null, D, d, null, null)$
else
▷ random attribute selection
    $A \leftarrow random(\mathcal{A} \setminus C)$
▷ random value selection
    $v \leftarrow random([min_{x \in D} x.A, max_{x \in D} x.A] \setminus \cup_J \{J \in I^A\})$
    $marg \leftarrow margs[A]$
    $I^A \leftarrow I^A \cup [v - marg, v + marg]$
    if $[\min_{x \in D} x.A, \max_{x \in D} x.a] \subseteq I^A$ **then**
▷ A has been entirely scanned
        $C \leftarrow C \cup \{A\}$
    end if
▷ points contained in the margin
    $D_m \leftarrow \{x \in D / x.A \in [v - marg, v + marg]\}$
    if $|D_m| \leq \alpha \times |D|$ **then**
        $D_l \leftarrow \{x \in D / x.A \leq v\}$
        $D_r \leftarrow \{x \in D / x.A > v\}$
▷ returns an internal node
        Return $node(build\_tree(D_l, d + 1, \alpha, \emptyset, \emptyset, margs),$
                $build\_tree(D_r, d + 1, \alpha, \emptyset, \emptyset, margs), D, d, A, v)$
    end if
▷ selection of another split
    Return $build\_tree(D, d, \alpha, I, C, margs)$
end if

any attribute (line 3), then a terminal node is returned (line 4), the current set of points being considered as inseparable.

Hypothesis 2

If the method is unable to find a valid separation on any attribute, the current set of points is considered as inseparable.

In a tree generated by CADI a leaf may be of three different types depending on the termination condition that yields it:

— an **Isolation Node** (IN) stores a data point that has been isolated from the rest of the data set; it is generated when the node contains one point ($|D| = 1$);

— a **Dense Node** (DN) gathers a set of inseparable points, viz. if $|D| > 1$ and all the attributes have been discarded ($C = \mathcal{A}$ on line 3 in Alg. 3);

— a **Depth-Limit Node** (DLN) is a node which is terminal because the depth limit has been reached, viz. $d = h_{lim}$ and $C \neq \mathcal{A}$.

Definition 12: DN leaf

Given a margin of width $marg$ controlled by α, a leaf gathering a set D' of points is called a DN leaf iff., $\forall A \in \mathcal{A}$ and $\forall v \in [min_{x \in D'} x.A, max_{x \in D'} x.A]$:

$$|\{x \in D', x.A \in [v - marg, v + marg]\}| > \alpha \times |D'|,$$

where $marg = \frac{1}{2}\alpha(\max_{x \in D} x.A - \min_{x \in D} x.A)$

In Definition 12, D is the sample used to build the tree. $marg$ is therefore of fixed size given α. More details will be provided in the first section of Chapter 3.

Whereas the original IF algorithm yields only nodes of type IN and DLN, the nodes of type DN induced by the density constraint applied on the randomly generated splits are particularly informative in the prospect of reconstructing the data inner structure (Sec. 2.3).

2.2.2 Evaluation stage: anomaly scores

With classic IF, after the forest construction, the evaluation stage begins. Each instance traverses all the trees until it reaches a terminal node. The anomaly score of an instance x in the data set is computed using the following formula:

$$s(x) = 2^{-\frac{\bar{h}(x)}{c(\Psi)}}, \tag{2.1}$$

In Eq. 2.1, $\bar{h}(x)$ is the average path length of x. The path length of a data point in a tree is computed using the method in Algorithm 4 [1]. This path length is the depth of the terminal node containing x after it has traversed the tree.

Algorithm 4 Isolation Forest : *path_length* [LTZ12]

1: **Inputs:** an instance $x \in \mathcal{D}$, an isolation tree T, the depth limit h_{lim}, the current path length h; $h = 0$ when the method is first called
2: **Output:** the path length of x in the tree T
3: **if** T is a terminal node **then**
4: Return $h + c(T.size)$
5: **end if**
6: $A \leftarrow T.splitAtt$ ▷ The split attribute of the current node
7: $v \leftarrow T.splitVal$ ▷ The split value of the current node
8: **if** $x.A \leq v$ **then** ▷ The value of x on attribute A is less than the split value
9: Return $path\_length(x, T.left, h_{lim}, h + 1)$
10: **else**
11: Return $path\_length(x, T.right, h_{lim}, h + 1)$ ▷ x goes to the right child
12: **end if**

When the depth limit is reached, the returned value is h plus an adjustment size $c(T.size)$. This adjustment represents an estimate of an average path length of a random sub-tree which could be constructed using data of size $T.size$ beyond the tree height limit [LTZ12]. If the average depth is equal to $c(\Psi)$ in Eq. 2.1, meaning that in every tree the search of the data point was unsuccessful -*because the tree depth limit was reached*-, then the anomaly score is equal to 0.5.

Leveraging the property that it is faster to isolate anomalies than regularities using random splits, the anomaly score of a given point in the original IF relies solely on its

1. In the approach as described in [LTZ12], the depth limit only intervenes during the evaluation stage. The trees are built until all the instances are isolated. However, because of line 4 of Alg. 4, the outcome is the same (when stopping the tree construction at the depth limit VS when building the trees until isolation and stopping the evaluation at the depth limit.)

depth of isolation in the different trees of the forest. With the CADI approach, because of the density constraint imposed on the randomly generated splits, a dense region may be scanned completely without increasing the depth of the tree, resulting in a leaf of type DN which contains a high number of inseparable points located at low depth. To differentiate leaves of type IN from those of type DN, it therefore makes more sense to define an anomaly score based on the cardinality of the set of points isolated in the same terminal node, instead of its depth. Equation 2.2 is used to calculate an anomaly score for a given point x that depends on the cardinality of the node it is isolated in:

$$s_i(x) = 1 - \frac{|\eta_i(x)| - 1}{\Psi}, \tag{2.2}$$

where $\eta_i(x)$ is the node containing x in the tree T_i. The score $s_i(x)$ varies in $]0, 1]$ taking its maximum value when x is isolated alone in an IN leaf and is close to 0 when the whole data sample ends in the same leaf. This last situation occurs when no separation line can be validated on the whole universe: the data set consists of a single indivisible cluster.

The global anomaly score is the average over the whole forest containing t trees:

$$s(x) = \frac{1}{t} \sum_{i=1}^{t} s_i(x). \tag{2.3}$$

Algorithm 5 details the evaluation stage of the CADI approach.

Algorithm 5 CADI: *compute_score*

Inputs: an instance $x \in \mathcal{D}$, an isolation tree T
Output: the score of x in the tree T
if T is a terminal node **then**
    Return $|T|$
end if
$A \leftarrow T.splitAtt$
$v \leftarrow T.splitVal$
if $x.A \leq v$ **then**
    Return $compute\_score(x, T.left)$
else
    Return $compute\_score(x, T.right)$
end if

2.2.3 Complexity analysis

During the forest construction, a classic IF has a constant time complexity of $O(2^{h_{lim}}t\Psi)$ in the worst case. At each node, the value of every data point on the randomly selected split attribute is compared against the split value, and there are at most $2^{h_{lim}}t$ nodes in the forest. The space complexity is $O(2^{h_{lim}}t)$ in the worst case, for the storage of the nodes. At the evaluation stage, the time complexity is in the worst case $O(2^{h_{lim}}tn)$ (linear) for the evaluation of the whole data set.

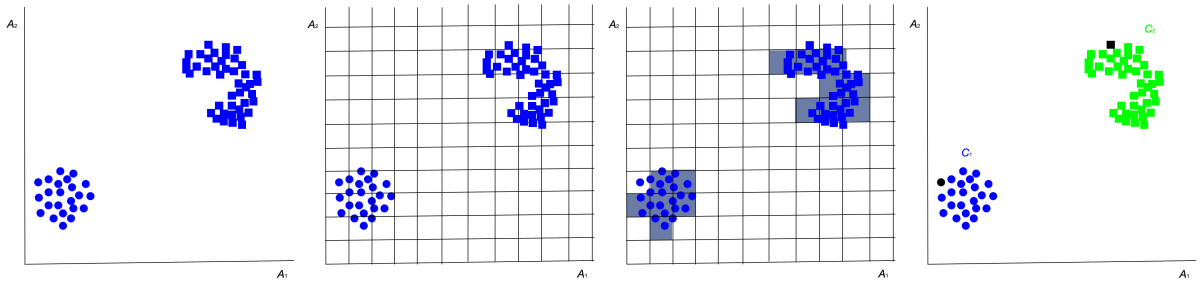
In addition to the classic IF cost, the forest construction stage of CADI induces an overhead due to the split selection. This overhead is in the worst case linear in the number of attributes. The time complexity of the split attribute selection is $O(2^{h_{lim}}td)$ in the worst case, while the time complexity of the split value selection is $O(2^{h_{lim}}t|I^A|)$ (constant) in the worst case. There are at most $O(2^{h_{lim}}t\Psi)$ comparisons for the points in the margin and the other data points in the node. The overhead space complexity of CADI during the forest construction is caused by the storage of scanned attributes and values, and the storage of the margins. This overhead is linear in the number of attributes. During the evaluation stage, the complexity of CADI is the same as IF.

2.3 Clustering from an Isolation Forest

A CADI forest has three types of terminal nodes: IN, DN, and DLN. Each IN leaf contains a potential anomaly, as the data point was isolated. DLN leaves are those containing points which have not been separated after a certain number of splits, just like in IF. DN leaves contain points that could not be separated, no matter the attribute. They therefore gather dense groups of points, corresponding to clusters or portions of clusters. As sampling is performed and trees are built on different parts of the data set, they most likely contain portions of clusters. Consequently, in order to obtain a complete partition of the data set, these leaves may need to be combined.

The combination strategy of CADI's DN leaves is inspired by grid-based clustering [Agr+98]. Grid-based clustering algorithms first partition the data space into a finite number of cells/blocks/units to form a grid structure. Each cell of the grid is a combination of r intervals on every attribute in $\mathcal{A}$. As such, each cell contains the data points belonging to the small portion of the data space delimited by the bounds of the intervals, based on their values on the attributes. As clusters correspond to regions that are more dense in data points than their surroundings [AR14], the most dense contiguous cells are

combined to form clusters. In GRIDCLUS [Sch96] for example, the cells are sorted in decreasing order of density. The most dense cells are chosen as cluster centers. A neighbor search is then applied on the cells to construct the rest of clusters. The neighbor search is done recursively starting at each cluster center and adding contiguous dense units to the cluster. Figure 2.5 illustrates the generic grid-based clustering process. The cells combination in grid-based clustering allows to discover non-elliptic clusters.

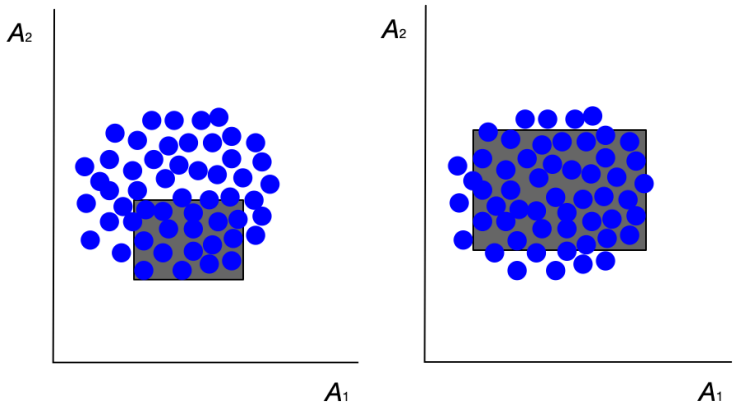


(a) Data set (b) Grid construction (c) Dense cells identification (d) Final clustering

Figure 2.5 – Grid-based clustering

Like cells in grid-based clustering, DN leaves in CADI contain data points and delimit areas within the data space. However, there are two major differences between cells and DN leaves. First, the regions limited by DN leaves may overlap. At the tree level, there is no overlap. The set of terminal nodes (including DN leaves) is a partition of the space containing the sample. Each DN leaf delimits a region separated from the regions enclosed by the other DN leaves of the same tree. However, each tree only paints an incomplete picture of the data set and the potential clusters. The knowledge imparted by each tree in the forest needs to be consolidated like during the anomaly detection phase. The regions delimited by DN leaves coming from different trees may overlap, unlike grid cells. The need for a combination of the information coming from different trees also engenders the second major difference between DN leaves and cells: DN leaves coming from different trees may have some points in common, whereas grid cells are disjoint in terms of points. At this stage, two options are opened for consideration in order to obtain a partition of the data set: combining the regions delimited by the DN leaves, or combining the points contained in the DN leaves. In grid-based clustering, the first option is employed because the contiguity of cells is relatively easy to verify, and the merging condition only involves contiguity and density. With CADI DN leaves, the contiguity (or inclusion) is much more complex to verify. Because of the random selection of the separations and

the ensemble technique, the regions delimited by two DN leaves can be disjoint, one can be fully included in the other or the intersection between the two can be non-empty like in Fig. 2.6. In Fig. 2.6, l_1 and l_2 need to be combined to obtain a bigger part of the cluster. However, if we have to combine the delimited regions (in grey), we may have to check the intersection attribute by attribute, which is not optimal. There is no hypothesis that could speed up the process. Even by restricting each interval of the region to the minimum and maximum values of the data points in each leaf, there may still be some overlap between the DN leaves that may be checked. The least complex solution is to combine the leaves based on the common points. On Fig. 2.6, as l_1 and $l2$ have many points in common, they are probably parts of the same cluster.



(a) DN l_1 (b) DN l_2

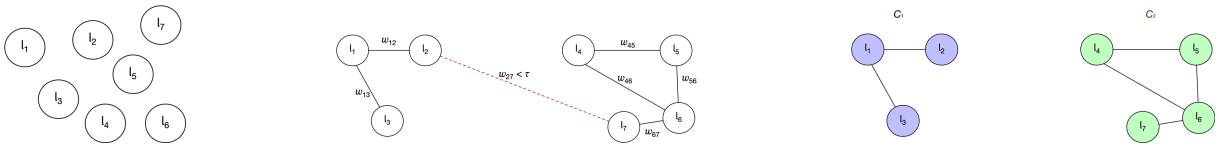
Figure 2.6 – Two DN coming from different trees

In grid-based clustering, the merging condition is for both cells to be dense and neighbors (contiguous). With CADI, the merging condition is for both leaves to be sufficiently similar in terms of points. In the grid-based clustering algorithm CLIQUE [Agr+98], after identifying dense cells in subspaces of the original feature space, a graph $\mathcal{G} = (\mathcal{V}, \mathfrak{E})$ is built. Each vertex of the graph is a cell and there is an edge E between two vertices $V_1, V_2 \in \mathcal{V}$ if the corresponding cells are contiguous. The connected components of the graph are later extracted, and each connected component is a cluster. In CADI, instead of creating an edge between two vertices V_1 and V_2 if the corresponding leaves l_1 qnd l_2 are contiguous, an edge E is created if the two leaves are somehow similar in terms of points. This similarity is measured by the *Jaccard index* between the two leaves and is

the weight of E:

$$w_E = \frac{|l_1 \cap l_2|}{|l_1 \cup l_2|} \tag{2.4}$$

with $E = (V_1, V_2) \Leftrightarrow (l_1, l_2)$. Like in CLIQUE, each connected component is a cluster. This strategy first allows to automatically discover the number of clusters in the data set. Second, it helps obtaining a partition of the data set without computing similarity between pairs of points, but instead between pairs of DN leaves. There are at most $2^{h_{lim}}t \ll n$ DN leaves in the forest. The most similar leaves are merged first, and the weakest edges are deleted. These weakest edges may correspond to edges whose weights are less than a specified percentile of the weights, which is another hyper-parameter τ. These steps are illustrated on Figure 2.7.



(a) DN leaves

(b) Computing the similarity between leaves and building the graph

(c) Extracting the connected components

Figure 2.7 – CADI clustering

Definition 13: Cluster

A set of DN leaves $C_m = \{l_1, l_2, \ldots, l_p\}$ is a cluster iff.:

$$\forall l_i \in C_m, \exists l_j \in C_m \text{ s.t. } l_i \textit{ and } l_j \text{ are sufficiently similar.}$$

The data points not affected to a cluster, because they were not part of any sample used to build the forest, traverse each tree until they reach a DN leaf. A majority vote is then performed among the corresponding DN leaves to assign these points to a cluster. This is illustrated on Figure 2.8.

Before the extraction of the connected components, a pre-processing step is applied: the leaves included in other leaves are deleted from $\mathcal{V}$ to remove redundancies. Algorithm 6 details the clustering stage of CADI.

The space complexity of the clustering phase is quadratic in the number of DN leaves, which is a constant for h_{lim} and t fixed: $(2^{h_{lim}}t)^2$. The time complexity is also quadratic

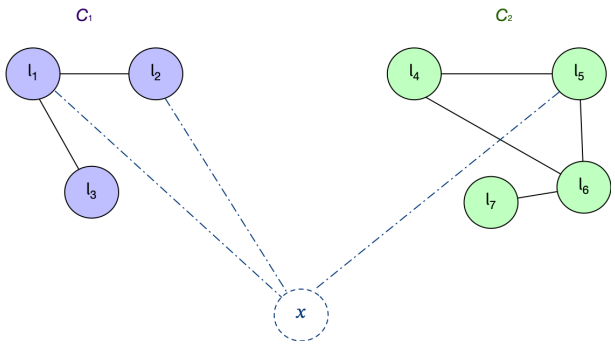


Figure 2.8 – Out-of-samples cluster assignation: x is found in leaves l_1, l_2 and l_5. It is therefore assigned to cluster C_1 after a majority vote.

Algorithm 6 CADI: *clustering*

Input: a CADI forest $\mathcal{F}$,
Output: a partition $\mathcal{C} = C_1 \cup \dots C_k$ of the data set $\mathcal{D}$
Compute anomaly scores of points in $\mathcal{D}$ using Alg. 5
Remove anomalies
$\mathcal{L} \leftarrow$ DN leaves of $\mathcal{F}$
Pre-processing: remove from $\mathcal{L}$ the leaves included in others
Compute pairwise similarities between elements of $\mathcal{L}$ using Eq. 2.4
Build graph $\mathcal{G}$
Remove the weakest edges
$\mathcal{C} \leftarrow connected\_components(\mathcal{G})$
Assign clusters to out-of-samples instances
Return $\mathcal{C}$

in the number of DN leaves for the pre-processing, the computation of weights and the removal of the weakest edges. The extraction of the connected components is linear in the number of DN leaves. The worst time complexity of the out-of-samples cluster assignation is linear in the data set size.

2.4 Anomaly Explanation

At this stage of the CADI framework, we have on the one hand the anomalies and on the other hand the regular data structure (viz. the clusters). The following task involves explaining the anomalies in relation to this structure.

In the COIN approach [LSH18], an outlier explanation has three parts. The first one is a quantification of the point abnormality. The second one is a local positioning of the anomaly to explain in relation to the surrounding regular instances. This is equivalent, for the method, to a set of clustered neighbors of the point. The last part of an anomaly explanation in the COIN approach is the set of attributes weighted by their relative contribution to the abnormality of the suspicious instance. These weights are obtained by training local linear classifiers to distinguish between the outlying class (obtained after employing synthetic over-sampling on the anomaly to explain) and the regular class (represented by a cluster of neighbors). The ATON [Xu+21] approach returns as explanation of an anomaly the set of weighted features, with the local context (regular neighbors of the instance) having been harnessed for the construction of the embedding space.

As mentioned in Section 1.3 when reviewing the existing taxonomies of explanations, an anomaly score does not explain why the instance is abnormal. It instead provides the extent to which the instance is deviating, which is the role of the anomaly detector. Furthermore, although the neighboring instances of the anomaly to explain are utilized to build the embedding space and later compute weights in ATON, there is no reference to them in the final explanation. COIN does include these neighboring instances in the final explanation and even clusters them. Yet, there is a dearth of information concerning clusters farther away. If the anomaly to explain is really close to one cluster, all the neighbors will be drawn from this cluster, and the information regarding the other clusters will be lost. In addition to that, there is no clue as to whether the anomaly is closest to a particular cluster. CADI enriches the explanations with these two information. An outlier explanation with CADI therefore consists of three components:

1. A quantification of its proximity to the different clusters.

2. The set of weighted common features with each cluster.

3. The set of weighted discriminating features with each cluster.

These three parts of a CADI explanation provide an answer to the following questions: 1) Which cluster is the anomaly to explain closer to? 2) How much does each attribute contribute to making the anomaly an element of each cluster? 3) How much does each attribute contribute to making the point an anomaly for each cluster, viz. which attributes make the instance an anomaly of each cluster?

Where COIN combines the weights coming from the different local cluster-specific classifiers, CADI does not average the weights and instead provides a global view relative to each identified cluster.

2.4.1 Local Structure-Aware Anomalies

In its original version, an IF detects points that may be easily separated from the rest of the data set, the so-called global anomalies. Local anomalies are also identified by an IF, but no distinction between local and global anomalies is made in the output of the method. The first part of the explanatory component of CADI makes such distinction, by specifying the proximity of an anomaly to the different clusters. Let us now show that a forest generated by CADI embeds all the necessary structural knowledge to identify possible links between anomalies and clusters.

Let x be a point whose anomaly score $s(x)$ is sufficiently high to consider that it is an anomaly. The first component of $\mathcal{E}(x)$, the explanation of x, determines for each cluster $C \in \mathcal{C}$ if x can be considered as an abnormal deviation of the regular phenomenon modelled by C. Let $\{l_1, \ldots, l_p\}$ be the set of DNs making up C: $C = l_1 \cup \ldots \cup l_p$. Let $T_{l_i}, i = 1...p$ be the tree in which l_i is found.

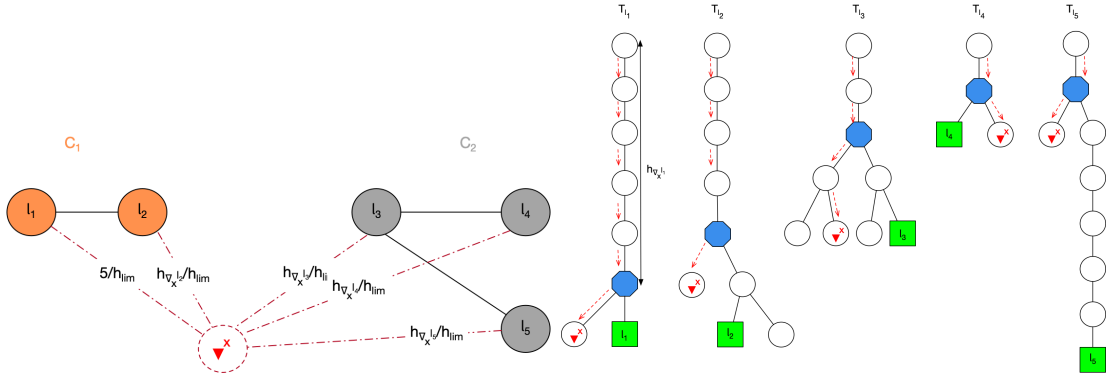
Using the structural knowledge embedded in T_{l_i} only, viz. without having to choose an appropriate distance measure, a contextual score denoted by $c(x, C)$ is computed as an aggregation of the comparisons between x and the l_is forming C. In a tree T_{l_i}, the path from the root to the leaf l_i consists of different separations each narrowing the region originally enclosed by the root. As a result, if x and the points in l_i are found in the same node *deep in the tree*, they are more likely to be close to each other in the feature space. By applying this principle to all the l_is in a cluster C, a score corresponding to the local deviation of x from C is computed. This contextual score depends on the depth of the deepest common ancestor between x and each l_i in the corresponding T_{l_i} (Fig. 2.9):

$$c(x, C) = \frac{1}{p} \sum_{i=1}^{p} \frac{h_{\nabla_x^{l_i}}}{h_{lim}}, \tag{2.5}$$

where $\nabla_x^{l_i}$ is the deepest common ancestor of x and the leaf node l_i, and $h_{\nabla_x^{f_i}}$ is its depth.

The higher the contextual score, the closer to the cluster the instance. Global anomalies have similar scores with all the clusters, indicating that they are not close to a particular cluster.

The process described above is illustrated on Figure 2.9. On Fig. 2.9a, two clusters C_1 and C_2 have been identified: $C_1 = \{l_1, l_2\}$ and $C_2 = \{l_3, l_4, l_5\}$. The contextual score of x with each cluster is computed using the depth of the deepest common ancestor between x and each l_i (Fig. 2.9b). These contextual scores are therefore given by: $c(x, C_1) = (5/h_{lim} + 4/h_{lim})/2 = 9/2h_{lim}$ and $c(x, C_2) = (2/h_{lim} + 1/h_{lim} + 1/h_{lim})/3 = 4/3h_{lim}$. In conclusion, x is a local anomaly of C_1.



(a) Clusters C_1 and C_2

(b) Deepest common ancestors (blue octogons) between x and the l_is (green squares). Partial trees are displayed.

Figure 2.9 – Contextual/Local anomaly detection: leveraging CADI trees and DN leaves

2.4.2 Common Attributes

In addition to the contextual scores computed between an anomaly x and each cluster $C \in \mathcal{C}$, CADI determines which attributes make x a local anomaly of a given cluster C. Each attribute is associated with a weight denoted by $e_{com}(x, C, A)$ that indicates if the value taken by x on A ($x.A$) is shared with other members of C. To quantify this weight, the paths in each T_{l_i} from its root to $\eta_i(x)$ and l_i respectively are analyzed. In a tree T_{l_i}, the path from the root to the deepest common ancestor $\nabla_x^{l_i}$ of $\eta_i(x)$ and l_i contains splits

that were not able to separate x from the points in l_i. Each split, therefore reinforces the closeness of x and l_i. Consequently, the weight of the attribute associated to the split must increase to take into consideration this reinforcement. The degree $e_{l_i}(x, A)$ quantifies the contribution of attribute A to explain x as a local anomaly of the dense subset of points gathered in the leaf l_i. It is simply the number of times attribute A is used as split attribute in the path from the root of T_{l_i} to $\nabla_x^{l_i}$ (excluded, since x and the points in l_i are separated at $\nabla_x^{l_i}$). The weight of A is therefore the average, on the different DN leaves l_i composing C, of $e_{l_i}(x, A)$.

$$e_{com}(x, C, A) = \frac{1}{p} \sum_{i=1}^{p} \omega_i(A), \tag{2.6}$$

where $\omega_i(A)$ is the number of times attribute A is used as a separation attribute in the path from the root of T_{l_i} to $\nabla_x^{l_i}$ (excluded).

Following the example from Fig. 2.9, Figure 2.10 shows the attributes of each separation from the roots of the trees to the deepest common ancestors.

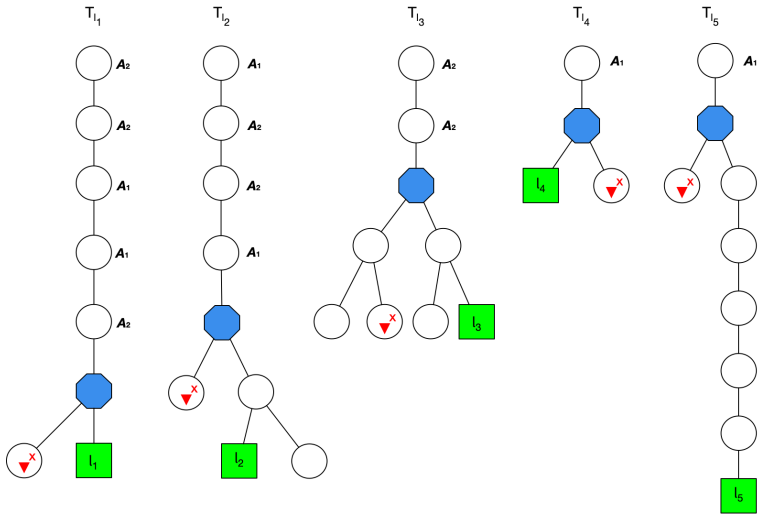


Figure 2.10 – Example from Fig. 2.9: common attributes

The weights of each attribute are: $\omega_1(A_1) = 2$, $\omega_1(A_2) = 3$, $\omega_2(A_1) = 2$, $\omega_2(A_2) = 2$, $\omega_3(A_1) = 0$, $\omega_3(A_2) = 2$, $\omega_4(A_1) = 1$, $\omega_4(A_2) = 0$, $\omega_5(A_1) = 1$ and $\omega_5(A_2) = 0$. As a result, $e_{com}(x, C_1, A_1) = 2$, $e_{com}(x, C_1, A_2) = 2.5$, $e_{com}(x, C_2, A_1) = 2/3$ and $e_{com}(x, C_2, A_2) = 2/3$.

Noteworthy is that e_{com} can be computed only for the closest clusters (greatest $c(x, C)$ from Eq. 2.5).

2.4.3 Discriminating Attributes

The discriminating attributes are the ones making the instance x abnormal for the cluster C. Each attribute is associated with a weight denoted by $e_{disc}(x, C, A)$ indicating how much the attribute A contributes to the abnormality of x with regard to C. A contributes to the abnormality of x with regard to C if A is frequently separating x from the instances in C. This separation occurs right at the deepest common ancestor $\nabla_x^{l_i}$ for a given l_i. In other words, the discriminating weight of A is the average over the l_is of the number of times A was the split attribute of the deepest common ancestor of x and l_i.

$$e_{disc}(x, C, A) = \frac{1}{p} \sum_{i=1}^{p} \delta(A, \nabla_x^{l_i}), \tag{2.7}$$

where $\delta(A, \nabla_x^{l_i}) = 1$ if A is the split attribute of $\nabla_x^{l_i}$ and 0 otherwise.

Following the example from Fig. 2.9, Figure 2.11 shows the split attribute at the deepest common ancestor. $e_{disc}(x, C_1, A_1) = 1$, $e_{disc}(x, C_1, A_2) = 0$, $e_{disc}(x, C_2, A_1) = 2/3$ and $e_{disc}(x, C_2, A_2) = 1/3$.

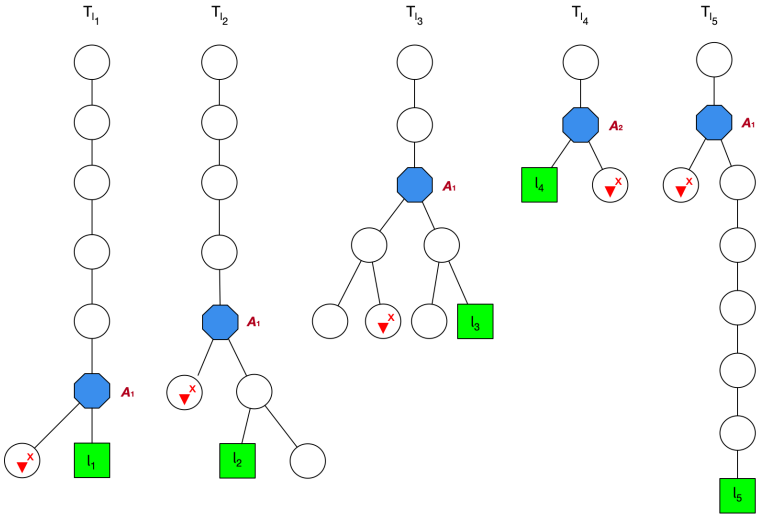


Figure 2.11 – Example from Fig. 2.9: discriminating attributes

Noteworthy is that e_{disc} can be computed only for the closest clusters (greatest $c(x, C)$ from Eq. 2.5).

The final explanation returned by CADI for an anomaly is illustrated on Figure 2.12.

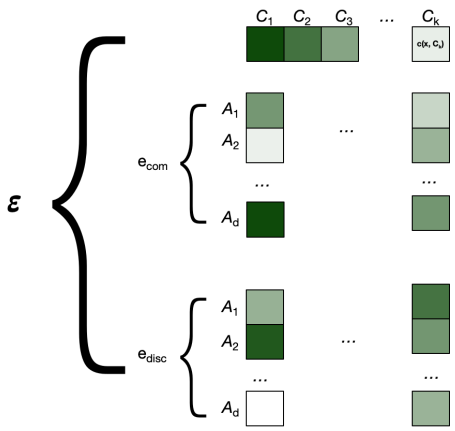


Figure 2.12 – Final explanation returned by CADI: $c(x, C_i)$, e_{com} and e_{disc}.

2.5 Summary

This chapter offered a comprehensive presentation of CADI, our unified approach to extract anomaly explanations by structure analysis. The modifications performed on the original IF approach were detailed in Section 2.2. The main revision is the introduction of a split selection criterion, or rather a split preservation criterion. The splits are still generated uniformly at random, but the separations falling into dense areas are discarded. The introduction of this criterion caused an update of the anomaly scores computation, as explained in §2.2.2. It also refrained from adding too much complexity to the initial approach. The new trees have three types of leaves, among which the leaves containing groups of inseparable points are combined to obtain a partition of the data set. The combination strategy is inspired by grid-based clustering. However, instead of combining the regions delimited by the leaves, the leaves somehow similar in terms of points are merged. That similarity is measured by the Jaccard index between leaves (Sec. 2.3). In Section 1.3, the explanation generation process is extensively described. The separations between the instance to explain and the leaves composing each cluster are utilized.

In the next chapter, the performance of CADI will be assessed.

Chapter 3

Experiments

This chapter details the experiments conducted to assess the performance of CADI. Answers to the following questions are sought:

Q.1) Is CADI able to accurately detect anomalies in a data set?

Q.2) Do CADI's DN leaves gather compact and homogeneous sets of points that can be combined to retrieve a partition of the regular instances a the data set?

Q.3) Is CADI able to combine those DN leaves to retrieve an accurate partition?

Q.4) Is CADI able to identify contextual anomalies in relation to clusters of regular instances?

Q.5) Is CADI able to provide accurate explanations by structure analysis of the anomalies?

Q.1 is addressed in Section 3.2. Q.2 and Q.3 are explored in Sec. 3.3. Q.4 is addressed in Sec. 3.4 and Q.5 is addressed in Sec. 3.5. The chapter ends with a summary of the experiments and results obtained in Sec. 3.6.

Contents

3.1 Experimental Setting

As a reminder, CADI possesses the following hyper-parameters: the number of trees in the forest t, the sample size Ψ, the depth limit h_{lim}, α which controls the size of the margin around each separation and τ, the threshold on edge weights during clustering. These hyper-parameters are set to these default values: $t = 100$, $\Psi = 256$, $h_{lim} = 8$ and $\alpha = 5\%$. The margin size is $\alpha\times$ the attribute's initial range. The threshold τ for the removal of the weakest edges is set to 0 unless specified otherwise.

The default values of t, Ψ and h_{lim} are the same as those of IF. The intuition behind a fixed value of the margin size is the following: if two points are separated by less than $\alpha \times dom(A)$ on an attribute A, they should remain together during the tree building process. However, the value of that parameter can be adjusted with some knowledge about the data. For example, if the user wants to keep together data points having a difference in values on a specific attribute A lower than a quantity β, then the value of α for this attribute can be set to $\beta/dom(A)$. This fixed value of α causes an update on line 14 in Alg. 3. If the data distribution is uniform, then $marg/(\max_{x\in D} x.A - \min_{x\in D} x.A) \times |D|$ points should be expected in the margin of fixed size $marg$. Nevertheless, α is a lower bound of the quantity $marg/(\max_{x\in D} x.A - \min_{x\in D} x.A)$, as the quantity $(\max_{x\in D} x.A - \min_{x\in D} x.A)$ decreases with separation and $\alpha = marg/(\max_{x\in sample} x.A - \min_{x\in sample} x.A)$. Consequently, a value lower than $\alpha \times |D|$ is also lower than $marg/(\max_{x\in D} x.A - \min_{x\in D} x.A) \times |D|$. Line 14 is therefore not modified in the implementation and the experiments.

3.2 Anomaly Detection

This section of the experiments aims at evaluating the anomaly detection performance of CADI.

3.2.1 Data sets

Thirteen data sets, including 2 synthetic ones, are used. These data sets are the same as those used in [LTZ12] to evaluate IF. The dimension, number of instances and number of anomalies of each data set are presented in Table 3.1 and available in [Ray16]. Each data set contains inliers and outliers.

Table 3.1 – Considered anomaly detection data sets: dimension, number of instances and number of anomalies.

Name	d	n	**# of anomalies**
Annthyroid	6	7200	534
Arrhythmia	271	420	57
Breast	9	683	239
Cover	10	286048	2747
Hbk (*synthetic*)	4	75	14
Http	3	567498	2213
Ionosphere	32	351	126
Mammography	6	11183	260
Pima	8	768	268
Satellite	36	6435	2036
Shuttle	9	58000	3511
Smtp	3	95156	30
Wood (*synthetic*)	6	20	4

3.2.2 General Assessment against IF

Evaluating the effectiveness of an anomaly detection algorithm is a difficult task in the unsupervised setting without labels. Fortunately, anomaly detection data sets are generally adapted from imbalanced classification, and the rare labels are used as surrogates for the ground-truth outliers [Agg16; Cor21]. It is the case for the real-world data sets from Table 3.1. Unsupervised algorithms can therefore be evaluated using those data sets.

The Area Under the Precision-Recall Curve (AUPRC) and the Area Under the Receiver Operating Characteristic Curve (AUROC) can be both employed as metrics to evaluate an anomaly detector, because they are independent of the anomaly score threshold. For an anomaly detector f outputting an anomaly score for each instance, a score threshold γ has to be selected in order to obtain a binary partition of $\mathcal{D}$ composed of the set of inliers and the set of outliers. Let $\mathcal{P}(\gamma)$ be the set of predicted outliers given the threshold γ, and $\mathcal{G}$ the set of ground-truth outliers. The size of $\mathcal{G}$ is fixed, while the size of $\mathcal{P}(\gamma)$ depends on γ. As γ decreases, more instances are reported as outliers. The precision of f is the proportion of true anomalies among those identified as such:

$$Precision(\gamma) = \frac{|\mathcal{P}(\gamma) \cap \mathcal{G}|}{|\mathcal{P}(\gamma)|}.$$

The recall of f is the proportion of true anomalies correctly identified:

$$Recall(\gamma) = \frac{|\mathcal{P}(\gamma) \cap \mathcal{G}|}{|\mathcal{G}|}.$$

The Precision-Recall (PR) curve displays the precision against the recall for different values of γ. The Receiver Operating Characteristic (ROC) curve on the other hand plots the True Positive Rate (TPR) against the False Positive Rate (FPR) for different values of γ. The TPR is the recall. The FPR is the proportion of inliers wrongly reported as outliers. It is given by:

$$FPR(\gamma) = \frac{|\mathcal{P}(\gamma) \cap \mathcal{G}|}{|\mathcal{D} - \mathcal{G}|}.$$

For two given detectors f_1 and f_2, the dominance of the PR curve of f_1 over f_2 is equivalent to the dominance of the ROC curve of f_1 over f_2. There is therefore no need to plot both curves to have a general assessment. However, the ROC curve is monotonic and, while the areas under both curves are independent of γ, the AUROC has a simple probabilistic interpretation [HM82]:

> Given a ranking or scoring of a set of points in order of their propensity to be outliers (with higher ranks/scores indicating greater outlierness), the ROC AUC is equal to the probability that a randomly selected outlier-inlier pair is ranked correctly (or scored in the correct order).

The AUROC is therefore the most used metric in the literature, and the one that will be employed here.

CADI is first compared to the classic IF approach in terms of AUC. IF is still one of the best unsupervised anomaly detection methods, even better than some DL approaches [Han+22]. The means and standard deviations of the AUC after ten runs of each method are reported in Table 3.2. To provide a fair comparison, the default hyperparameters are used for each approach.

In general, the AUCs of CADI and IF are close. CADI performs better than IF on 6 data sets, and IF also performs better than CADI on 6 data sets. Both approaches obtain the same perfect results on the artificial data set *hbk*. In most data sets, there is no significant difference between CADI and IF. However, on *mammography*, CADI performs much better than IF with a gain of +0.196 in mean AUC. In average, CADI performs better than IF, with an average gain of +0.015 in mean AUC. Another observation is that

Table 3.2 – CADI vs IF: AUC

Data set	CADI	IF
Annthyroid	0.762 ± 0.013	**0.810 ± 0.014**
Arrhythmia	**0.812 ± 0.016**	0.794 ± 0.035
Breast	**0.994 ± 0.001**	0.981 ± 0.003
Cover	0.816 ± 0.047	**0.873 ± 0.028**
HBK	1.0 ± 0.0	1.0 ± 0.0
HTTP	0.998 ± 0.002	**0.999 ± 0.001**
Ionosphere	0.829 ± 0.006	**0.848 ± 0.006**
Mammography	**0.839 ± 0.011**	0.643 ± 0.038
Pima	**0.701 ± 0.011**	0.683 ± 0.009
Satellite	**0.700 ± 0.014**	0.699 ± 0.016
Shuttle	0.992 ± 0.002	**0.995 ± 0.001**
SMTP	0.880 ± 0.011	**0.886 ± 0.008**
Wood	**0.967 ± 0.029**	0.885 ± 0.057
Mean AUC	**0.868**	0.853

the standard deviations of the AUCs with CADI are generally slightly lower, implying that the results obtained are more stable. This stability is caused by the fact that some properties of CADI are controlled in a deterministic manner. Consequently, though CADI remains a random method, it is less random than classical IF.

3.2.3 Identified Anomalies

Why does CADI perform better than IF on some data sets? There are two aspects to the answer to this question. The first aspect can be observed on the statistical data set *wood*. On this data set, CADI frequently assigns a higher score to the real anomalies in comparison to IF, having a perfect AUC several times. It is not the case for IF. Table 3.3 shows the 10 data points that receive the highest anomaly scores with both methods during one execution. Using the same representation as in [LTZ12], Figure 3.1 shows the first two principal components of the data set.

The four highest-ranked instances by CADI are the actual anomalies of the data set (instances 4, 6, 8 and 19), whereas IF scores the instance 10 first. Observing the two principal components on Figure 3.1 shows that instance 10, although regular, lies in a low density region. Since with IF the anomaly score only depends on the average isolation depth of a data point (Eq. 2.1), it is more difficult for the method to make a distinction between real anomalies and regular data points located in low density regions. On the

Table 3.3 – Anomaly ranking of the *wood* data set: bold-faced indexes are actual anomalies.

Rank	CADI	IF
1	**19**	10
2	**4**	**19**
3	**6**	**4**
4	**8**	**8**
5	7	20
6	12	7
7	11	1
8	9	12
9	1	11
10	20	17

other hand, CADI takes the local density surrounding the data point into consideration while generating the splits and during the score computation (Eq. 2.3). As a result, CADI offers a better contrast between regular data points located in low density zones (but still surrounded by data points when the local density is considered) and anomalies (isolated). LOF also correctly identifies the true anomalies, and ranks them first [LTZ12]. With LOF, there is also no ambiguity because instance 10 has a surrounding density similar to the one of its neighbors. CADI combines separability -*anomalies are far from the other data points*- and local density information -*anomalies have a lower local density in comparison to their neighbors*- during the identification of anomalies.

The second difference between IF and CADI lies in the identification of local anomalies. This phenomenon can be observed on the data set on Figure 3.2. On this Figure, the opacity of each data point is proportional to its anomaly score. The scores are min-max scaled. It appears that CADI gives higher scores than IF to local anomalies. As a refresher, local anomalies are instances deviating from a portion of the data set.

3.2.4 Hyper-parameters Sensitivity

How much does the choice of the hyper-parameters influence the anomaly detection performance of CADI?

As a reminder, CADI introduces one additional hyper-parameter for anomaly detection: α, which controls the width of the margin surrounding the split. The default margin size is $\alpha\times$ the attribute's initial range, with $\alpha = 5\%$ (Sec. 3.1). When $\alpha = 0$, a classic IF

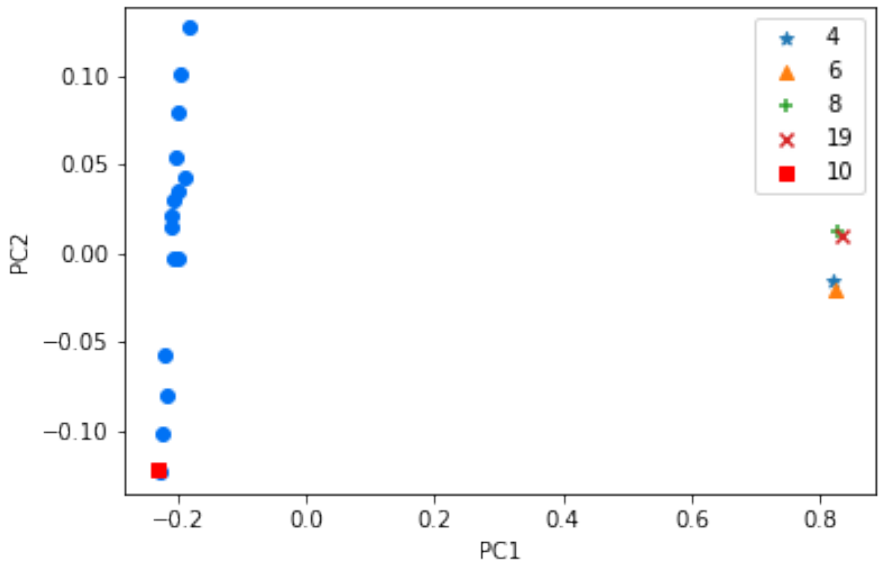
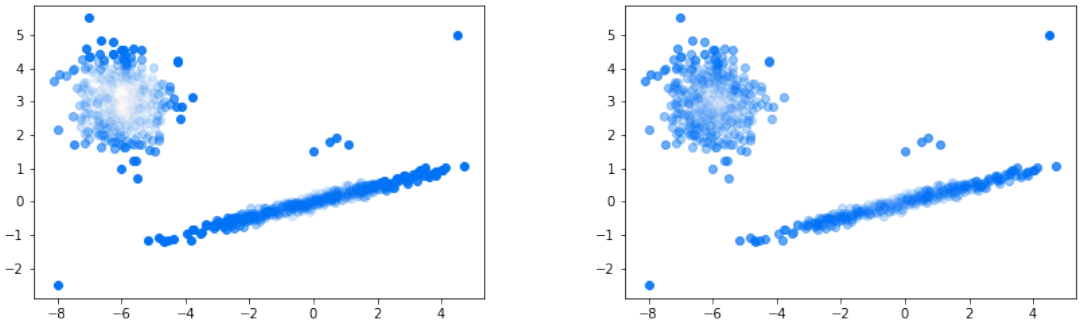


Figure 3.1 – First two principal components of the *wood* data set. Instances 4, 6, 8, 10 and 19 are displayed.



(a) CADI (b) IF

Figure 3.2 – Scores distribution: CADI vs IF

is built.

For different values of $\alpha \in \{1.25\%, 2.5\%, 7.5\%, 10\%\}$ and for each data set, ten CADI forests are built. The means and standard deviations for each value of α and for each data set are displayed on Fig. 3.3 and detailed in Table 3.4. The mean and standard deviation using default parameters are also added to Table 3.4. They had already been shown earlier in Table 3.2.

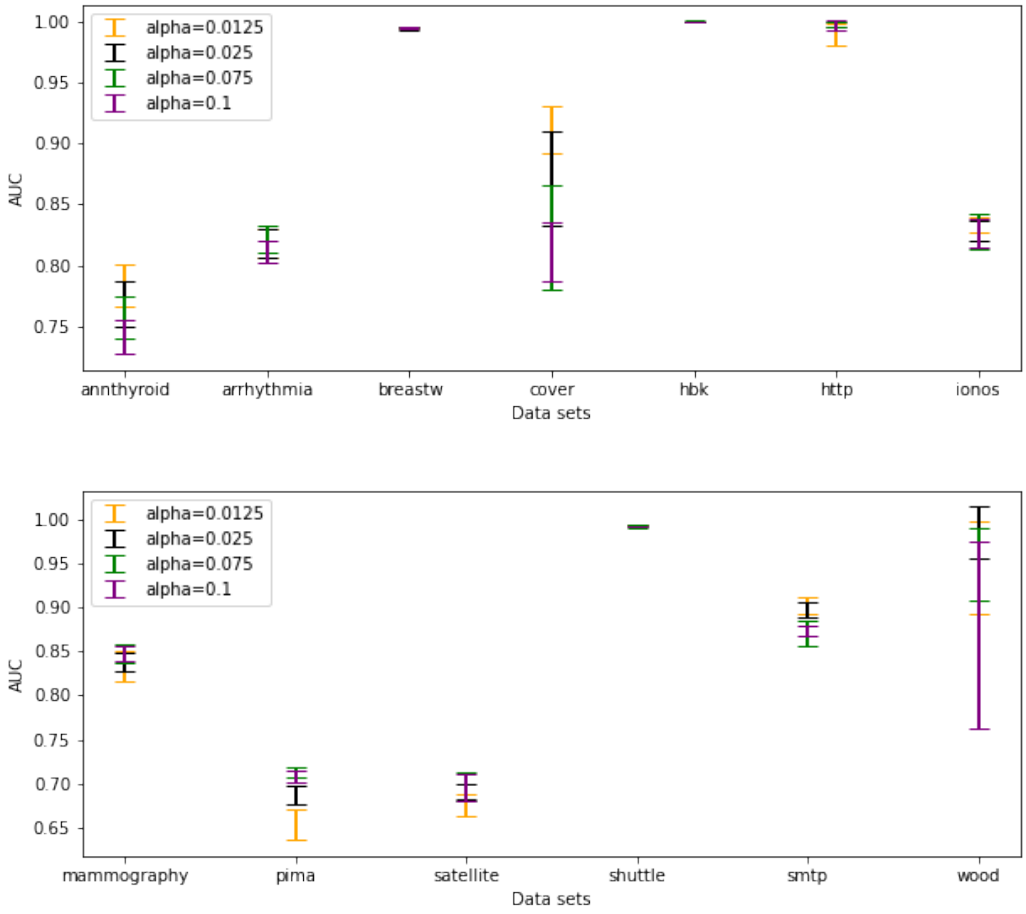


Figure 3.3 – Influence of α on the AUC

It appears that the AUCs do not vary much with the different values of α. Furthermore, no value of α is systematically better than the others across all the data sets. However, lower values of α tend to produce better results. $\alpha = 2.5\%$ provides the highest average AUC and the lowest average standard deviation, while $\alpha = 10\%$ delivers the worst average AUC and also the highest average standard deviation.

Table 3.4 – Means and standard deviations after 10 runs for different values of α

Data set	$\alpha = 1.25\%$	$\alpha = 2.5\%$	$\alpha = 5\%$	$\alpha = 7.5\%$	$\alpha = 10\%$
Annthyroid	0.783 ± 0.017	0.768 ± 0.019	0.762 ± 0.013	0.758 ± 0.017	0.742 ± 0.014
Arrhythmia	0.818 ± 0.012	0.818 ± 0.012	0.812 ± 0.016	0.821 ± 0.011	0.810 ± 0.009
Breast	0.994 ± 0.001	0.993 ± 0.001	0.994 ± 0.001	0.994 ± 0.001	0.994 ± 0.001
Cover	0.911 ± 0.019	0.871 ± 0.038	0.816 ± 0.047	0.823 ± 0.043	0.811 ± 0.024
Hbk	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
Http	0.989 ± 0.009	0.997 ± 0.002	0.998 ± 0.002	0.997 ± 0.002	0.997 ± 0.003
Ionosphere	0.833 ± 0.006	0.829 ± 0.008	0.829 ± 0.006	0.828 ± 0.014	0.826 ± 0.012
Mammography	0.834 ± 0.017	0.837 ± 0.010	0.839 ± 0.011	0.848 ± 0.010	0.848 ± 0.008
Pima	0.654 ± 0.017	0.687 ± 0.010	0.701 ± 0.011	0.713 ± 0.005	0.708 ± 0.007
Satellite	0.675 ± 0.012	0.691 ± 0.009	0.700 ± 0.014	0.697 ± 0.017	0.696 ± 0.014
Shuttle	0.993 ± 0.001	0.993 ± 0.001	0.992 ± 0.002	0.991 ± 0.002	0.992 ± 0.001
Smtp	0.902 ± 0.009	0.897 ± 0.009	0.880 ± 0.011	0.870 ± 0.014	0.873 ± 0.006
Wood	0.944 ± 0.053	0.984 ± 0.029	0.967 ± 0.029	0.949 ± 0.041	0.868 ± 0.106
Mean AUC	0.871	0.874	0.868	0.868	0.856

Can CADI yield better results with less trees in the forest? As the separations are not always completely random, we might wonder whether CADI's trees are so much more informative that less trees are required. The impact of the number of trees on the AUC is therefore also analyzed. Figure 3.4 illustrates, for each data set, the mean AUC obtained with forests containing 30 to 100 trees.

From Fig. 3.4, we cannot conclude that smaller forest in terms of number of trees yield better results, since there is a convergence around 100 trees on almost all the data sets. For the data set *satellite* where better AUCs are obtained with 50 and 60 trees, the performance cannot be attributed solely to the number of trees in the forest, as the samples also have an impact.

3.2.5 Assessment against Unsupervised Algorithms

In this last batch of experiments related to anomalies identification, CADI is compared against other unsupervised anomaly detection algorithms. The selected baselines are EIF [HKB21], SCIForest [LTZ10], LOF [Bre+00], COF [Tan+02], CBLOF [HXD03] and the statistical anomaly detection algorithm ECOD [Li+22]. The default values of the hyper-parameters are used for each algorithm to provide a fair comparison. In [Bre+00], the default value of the number of neighbors is not specified. In the original IF paper, this number is set to 10. Here we use the default value of $k = 20$. The implementations of EIF and SCIForest are provided in the package isotree [1]. The implementations of LOF, COF, CBLOF and ECOD are parts of the PyOD library [ZNL19]. Table 3.5 shows the results obtained by each method. For non deterministic methods, the average over 10 runs is displayed.

SCIForest and ECOD rank first the most (on four data sets). CADI has the best average AUC and the best average rank. COF struggles on data sets containing many instances, being unable to produce results in reasonable time. This is due to the computation of chaining distances which involves calculating the sum of all distances connecting k neighbors to the data point.

1. https://github.com/david-cortes/isotree/blob/master/README.md

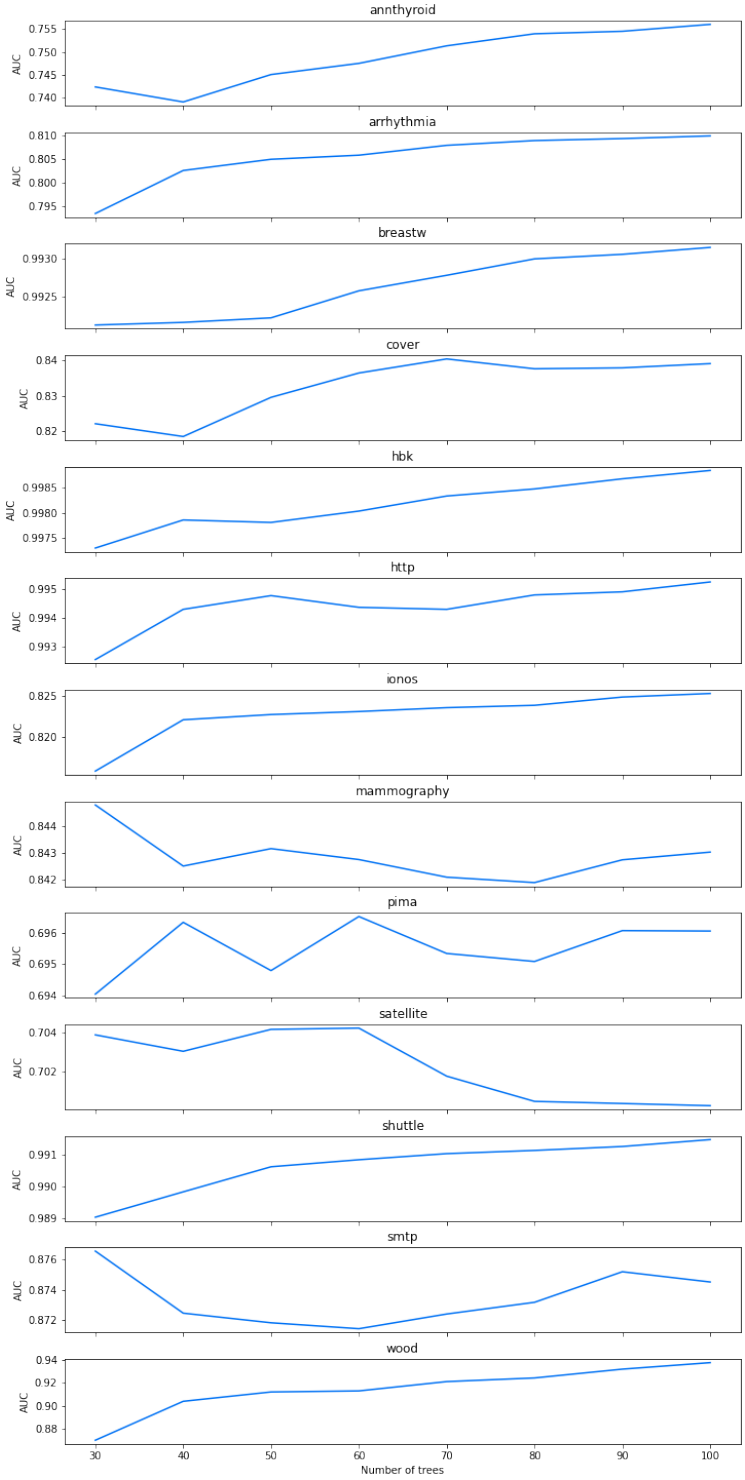


Figure 3.4 – Evolution of the AUC with the forest size

Table 3.5 – AUCs obtained by unsupervised methods. *** indicates that the method was not able to produce results in reasonable time.

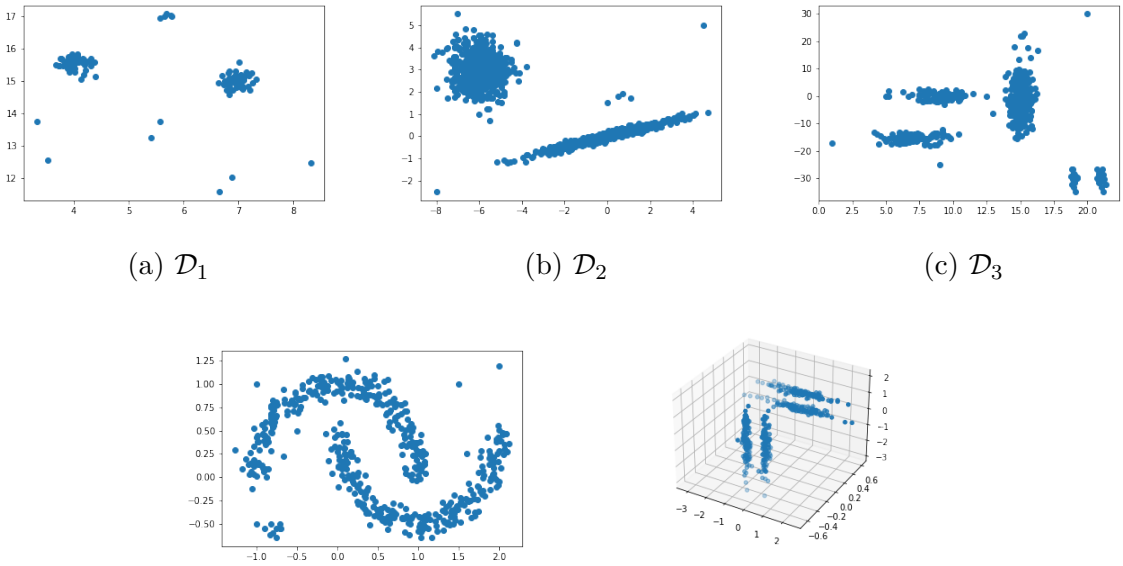
Data set	CADI	IF	EIF	SCIForest	LOF	COF	CBLOF	ECOD
Annthyroid	0.762 (3)	**0.810** (1)	0.716 (6)	0.760 (4)	0.739 (5)	0.710 (7)	0.659 (8)	0.789 (2)
Arrhythmia	0.812 (3)	0.794 (4)	0.813 (2)	0.683 (8)	0.784 (7)	0.794 (4)	0.790 (6)	**0.815** (1)
Breast	**0.994** (1)	0.981 (5)	0.987 (3)	0.983 (4)	0.385 (7)	0.332 (8)	0.970 (6)	0.991 (2)
Cover	0.816 (4)	0.873 (3)	0.904 (2)	0.704 (5)	0.555 (7)	***	0.630 (6)	**0.920** (1)
Hbk	**1.0** (1)	**1.0** (1)	**1.0** (1)	0.984 (7)	**1.0** (1)	**1.0** (1)	0.457 (8)	0.989 (6)
Http	0.998 (4)	**0.999** (1)	**0.999** (1)	**0.999** (1)	0.383 (7)	***	0.995 (5)	0.978 (6)
Ionosphere	0.829 (7)	0.848 (5)	0.843 (6)	0.890 (2)	0.860 (3)	0.859 (4)	**0.902** (1)	0.728 (8)
Mammography	0.839 (3)	0.643 (7)	0.869 (2)	0.585 (8)	0.719 (5)	0.716 (6)	0.815 (4)	**0.906** (1)
Pima	**0.701** (1)	0.683 (2)	0.676 (3)	0.600 (4)	0.538 (7)	0.519 (8)	0.577 (6)	0.594 (5)
Satellite	0.700 (2)	0.699 (3)	0.694 (4)	0.623 (5)	0.539 (7)	0.535 (8)	**0.731** (1)	0.583 (6)
Shuttle	0.992 (4)	0.995 (2)	0.993 (3)	**0.997** (1)	0.552 (7)	***	0.946 (6)	0.989 (5)
Smtp	0.880 (5)	0.886 (3)	0.868 (7)	**0.935** (1)	0.903 (2)	***	0.884 (4)	0.880 (5)
Wood	0.967 (3)	0.885 (4)	0.843 (5)	**1.0** (1)	0.703 (6)	0.453 (7)	0.141 (8)	**1.0** (1)
Mean	**0.868 (3.15)**	0.853 **(3.15)**	0.861 (3.46)	0.826 (3.92)	0.666 (5.46)	0.657 (5.89)	0.730 (5.31)	0.858 (3.77)

3.3 Clustering

In this section, the ability of CADI to build a partition of the regular instances in the data set is evaluated. Before evaluating the clustering process, the properties of the new trees and the DN leaves in particular are analyzed.

3.3.1 Data sets

The data sets used in this section are illustrated on Figure 3.5. They are constrained to 2D and 3D description spaces, to control the behavior of the method. Each data set contains clusters and anomalies: 2 clusters of regular data for $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_4$, 3 clusters of regular data for $\mathcal{D}_3$ and 4 clusters of regular data for $\mathcal{D}_5$. $\mathcal{D}_5$ is a three-dimensional data set in which each cluster is located in a 2-dimensional subspace. Its generation process is explained in [PHL04]. $\mathcal{D}_4$ is the data set *moons* composed of two interleaving half circles, to which anomalies have been added manually. $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$ and $\mathcal{D}_4$ also contain anomaly clusters.



(a) $\mathcal{D}_1$ (b) $\mathcal{D}_2$ (c) $\mathcal{D}_3$

(d) $\mathcal{D}_4$ (e) $\mathcal{D}_5$

Figure 3.5 – Data sets for the clustering experiments

3.3.2 Towards Identifying the Data Inner Structure

This part of the experiments aims at analyzing the properties of the CADI trees, in order to provide an answer to Q.2 (Do CADI's DN leaves gather compact and homogeneous sets of points that can be combined to retrieve a partition of the regular instances a the data set?). The goal is to investigate whether a relevant partition of the regular data points can be inferred from a CADI forest.

Leaf Cardinalities and Tree Depths

The impact of the split selection in CADI on the size of the leaves is first evaluated. With IF, the separations are completely random until a point is isolated or the depth limit is reached. It is therefore expected to have on one hand leaves containing isolated points (IN), and on the other hand leaves which have reached the depth limit and contain more than one point (DLN). With CADI, it is expected to have, in addition to IN and DLN leaves, leaves containing points that could not be separated, the so-called DN leaves. Ideally, there should be many DN leaves, depending on the chosen depth limit, because the objective is to preserve the clusters. Therefore, CADI trees should have a smaller height (the depth limit being more difficult to reach than in the classical version) and the leaves should contain more data points.

A classic IF and a CADI forest are built on each data set. The leaves containing isolated instances (IN leaves) are discarded. Then, the average cardinality of the leaves as well as the average depths of the trees of each forest type are computed. The means and standard deviations across 10 runs are reported in Table 3.6.

Table 3.6 – Statistics on the tree structures built by IF and CADI

Data set	Leaf sizes		Tree depths	
	IF	CADI	IF	CADI
$\mathcal{D}_1$	7.27 ± 0.30	28.37 ± 0.82	8.0 ± 0.0	4.77 ± 1.65
$\mathcal{D}_2$	8.76 ± 0.30	27.35 ± 1.12	8.0 ± 0.0	5.92 ± 1.06
$\mathcal{D}_3$	10.22 ± 0.46	21.52 ± 0.81	8.0 ± 0.0	6.12 ± 0.88
$\mathcal{D}_4$	6.64 ± 0.12	17.77 ± 0.49	7.99 ± 0.02	5.78 ± 0.80
$\mathcal{D}_5$	9.49 ± 0.63	13.34 ± 0.53	8.0 ± 0.0	7.50 ± 0.50

As expected, CADI leaves contain more points than the leaves of a classical isolation forest, and this on all the data sets. As for the trees, they are indeed more compact than

the classical isolation trees.

Types of Leaves

We then study the proportion of leaves that have reached the depth limit (DLN), as compared to the proportion of leaves containing points that are no longer separable (DN): a CADI forest is built and these two values are computed. Table 3.7 reports the means and standard deviations across 10 runs of this experiment.

Table 3.7 – Percentages of the different types of leaves

Data set	DLN leaves (%)	DN leaves (%)
$\mathcal{D}_1$	22.50 ± 3.40	77.50 ± 3.40
$\mathcal{D}_2$	36.85 ± 3.49	63.15 ± 3.49
$\mathcal{D}_3$	38.30 ± 2.15	61.70 ± 2.15
$\mathcal{D}_4$	24.16 ± 3.37	75.84 ± 3.37
$\mathcal{D}_5$	69.08 ± 3.10	30.92 ± 3.10

It appears that a significant proportion of leaves are DN. This phenomenon is verified on data sets $\mathcal{D}_1$ to $\mathcal{D}_4$, but not on data set $\mathcal{D}_5$. The latter also contains fewer points in the leaves, as compared to the other data sets and the trees of the CADI forest, although smaller than the classical isolation trees, are still deeper than those of the forests built on the other data sets (Table 3.6). This is explained by the fact that in $\mathcal{D}_5$, each cluster "exists" in only two of the three dimensions. However, the isolation process continues by separating the points on the third dimension, where they are distributed almost uniformly. The depth limit is not a function of the number of dimensions. However, as the dimension of the data set increases, there are more options for the split choice. As a result, the depth limit is more often reached. Increasing the depth limit taking into account the dimensionality mitigates the aforementioned problem. For example, by using a depth limit of 15 for $\mathcal{D}_5$ instead of the default value of 8, the percentage of DLN leaves decreases to 21.82 ± 2.03.

Data Point Proximity

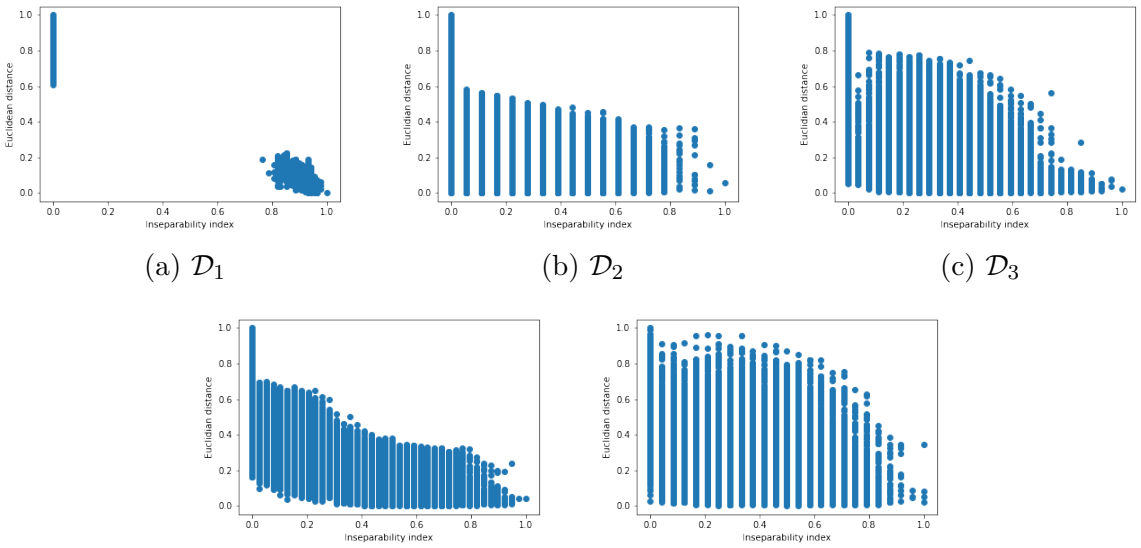
This part intends to check if CADI preserves groups of close data points. This proximity is measured in the original data space by the Euclidean distance. To measure

this proximity in CADI we introduce the *inseparability index* between two points as the average number of times they co-occur in the same DN. It is defined as follows:

$$sim(x_1, x_2) = \frac{1}{t} \sum_{l \in \mathcal{L}} \mathbb{1}_l(x_1, x_2) \tag{3.1}$$

with $\mathcal{L}$ the set of leaves of type DN in the forest, and $\mathbb{1}_l(x_1, x_2) = 1$ if $\{x_1, x_2\} \subseteq l$ and 0 otherwise.

For each pair of points in the data set, the Euclidean distance between them is calculated, as well as the inseparability index. Both values are min-max scaled. The results of the comparison between these two measures are displayed for each data set on Figure 3.6: for each pair of points, on the x-axis the inseparability index and on the y-axis the Euclidean distance.



(a) $\mathcal{D}_1$ (b) $\mathcal{D}_2$ (c) $\mathcal{D}_3$

(d) $\mathcal{D}_4$ (e) $\mathcal{D}_5$

Figure 3.6 – Euclidean distance vs inseparability index

Two seemingly counter-intuitive phenomena are observed when analyzing these results, but can be explained as follows. First, some data points, despite being close in Euclidean space (small Euclidean distance), are rarely found in the same leaf (inseparability index close to 0). This occurs when the two points, although close in the Euclidean space, are separable and thus belong to different clusters, for example the instances $(14.22, -0.70)$ and $(8.59, -0.89)$ in $\mathcal{D}_3$. It is especially the case when they have similar values on some dimensions. A separation between these two points can be kept if their neighborhood is

not dense. The aforementioned situation also occurs when one of the two points is very close to the cluster containing the other point, without being part of it, or when one of the two points is located at the border of the cluster and is therefore often separated from the others (e.g. points $(-7.10, 4.57)$ and $(-6.03, 3.16)$ in $\mathcal{D}_2$).

It can also be observed that some data points distant in the Euclidean space are sometimes found in the same leaves. This occurs when the two points, although distant in the Euclidean space, are part of the same cluster, for example when the cluster is stretched. This phenomenon frequently occurs in the data set $\mathcal{D}_5$ where all the four clusters are stretched.

This analysis suggests that three random points x_1 and x_2 then x_1 and x_3 can be located at the same Euclidean distance, but, using the information provided by the CADI forest, x_1 and x_2 are part of the same cluster, and x_3 is not, because many splits separate x_1 and x_3. The local density evaluation during the split selection therefore brings an additional knowledge useful for clustering.

Average Distances within and between Leaves

A cluster is a group of close points (compactness) which are separated from the groups of points in the other clusters (separability). This experiment therefore aims at checking whether CADI's leaves are portions of clusters containing close points forming compact and separated groups. To verify that, the average Euclidean distance between the points of each leaf and the center of the leaf is calculated for both forest types. It is the average distance within leaves. The average Euclidean distance between the centers of the leaves is also computed. The results are reported in Table 3.8. Ideally, the average distance within leaves should be small (compactness) and the average distance between leaves should be high (separability).

The average distance within leaves is larger in CADI in almost all data sets, which is understandable because classic isolation leaves contain less points as seen earlier in the experiments, and these data points are close. CADI's leaves in contrast contain larger groups of close data points. On $\mathcal{D}_5$, the average distance within leaves is larger in IF: for data points belonging to two clusters located in different subspaces, the distance is much larger. On the other hand, the distance between leaves is systematically larger in CADI, which reflects the fact that there is a better separability between leaves on CADI, in comparison to IF. CADI leaves are consequently more likely to be portions of clusters than IF leaves.

Table 3.8 – Average distances within and between leaves, means and standard deviations after 10 runs

	Average distance within leaves		Average distance between leaves	
	IF	CADI	IF	CADI
$\mathcal{D}_1$	**0.102 ± 0.002**	0.159 ± 0.002	1.684 ± 0.007	**2.467 ± 0.032**
$\mathcal{D}_2$	**0.233 ± 0.004**	0.395 ± 0.004	4.114 ± 0.034	**5.058 ± 0.043**
$\mathcal{D}_3$	**1.142 ± 0.037**	1.345 ± 0.033	13.250 ± 0.182	**18.597 ± 0.400**
$\mathcal{D}_4$	**0.117 ± 0.004**	0.143 ± 0.002	1.256 ± 0.006	**1.393 ± 0.011**
$\mathcal{D}_5$	0.341 ± 0.008	**0.246 ± 0.007**	1.419 ± 0.009	**1.630 ± 0.013**

Inseparability Index and Clustering

The purpose of this experiment is to verify whether, when two points have a low inseparability index (Eq. 3.1), they indeed belong to the same cluster. If it is the case, it would mean that CADI leaves contain points belonging to the same cluster.

For each data set, a CADI forest is built and anomalies are identified. The anomaly score threshold is set to $\gamma = 0.95$ [2]. Then, the inseparability index between each pair of points is computed and an Agglomerative Hierarchical Clustering (AHC) using average linkage is performed on the obtained similarity matrix. As the number k of clusters in the data set is known, the AHC is stopped when k groups are constructed.

The obtained clusters are compared to the expected ones using the *Adjusted Rand Index* (ARI), that equals 1 if the two partitions are identical. Table 3.9 shows the maximum ARI obtained on each data set, compared with the maximum ARI obtained when using the Euclidean distance as the distance measure for the AHC. In the ARI calculation, anomalies are considered as part of an isolated cluster.

Except on the data set $\mathcal{D}_1$, the ARI is higher when using the inseparability index. On $\mathcal{D}_3$, the separability information conveyed by the inseparability index allows to reconstruct the three regular clusters, whereas the Euclidean distance combines the upper-half of the biggest cluster to the cluster at the top left (Fig. 3.7a).

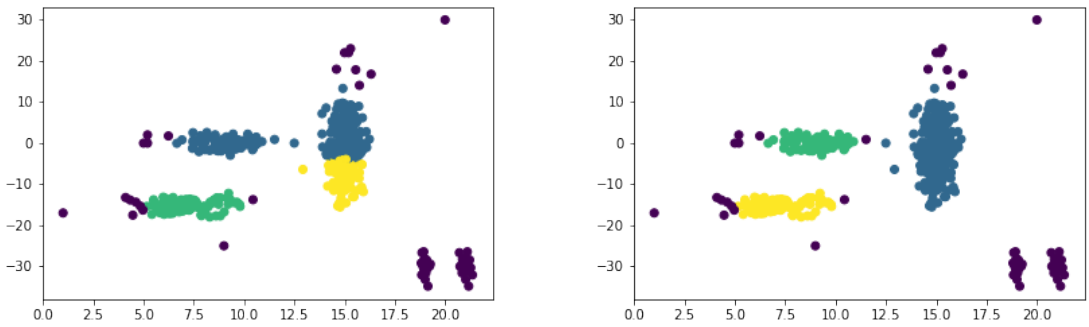
As DN leaves may be parts of clusters, their combination can lead to the discovery of non elliptic clusters. This is observed on data set $\mathcal{D}_4$.

On $\mathcal{D}_5$, the clusters are stretched and located in different subspaces. Consequently,

2. In practical anomaly detection, the user can either choose a threshold or consider as abnormal the p instances which receive the highest anomaly score, where p is a *small* percentage of the data set.

Table 3.9 – AHC on the data sets using inseparability index vs Euclidean distance: Adjusted Rand Indexes

Data set	Inseparability index	Euclidean distance
$\mathcal{D}_1$	1.0	1.0
$\mathcal{D}_2$	**0.927**	0.866
$\mathcal{D}_3$	**0.982**	0.485
$\mathcal{D}_4$	**0.972**	0.413
$\mathcal{D}_5$	**0.876**	0.623

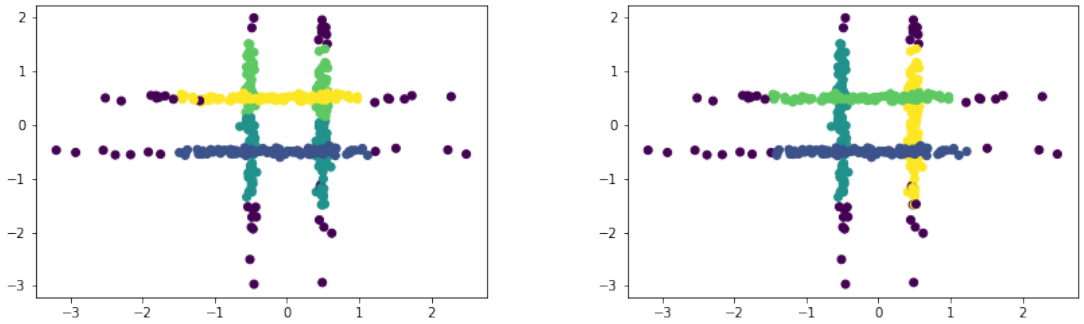


(a) Euclidean distance + AHC

(b) Inseparability index + AHC

Figure 3.7 – AHC results on $\mathcal{D}_3$: Euclidean distance vs inseparability index

and as observed on Figure 3.6e, points belonging to the same cluster may be far away from each other when considering the Euclidean distance. The inseparability index however is not tricked by that subtlety of the data set because the instances are frequently located in the same DN leaves. On the other hand, points belonging to different clusters are sometimes close when clusters are poorly separable. Again, whereas the combination Euclidean distance + AHC merges those two clusters, the inseparability index is able to keep them separated. Figure 3.8 illustrates these results.



(a) Euclidean distance + AHC (b) Inseparability index + AHC

Figure 3.8 – AHC results on $\mathcal{D}_5$

Points found in the same DN leaves are therefore indeed part of the same cluster. In the next paragraph, we will evaluate the leaves combination strategy.

3.3.3 Clustering Assessment

The previous experiments showed that DN leaves contain data points which are close to each other and may be combined to reconstruct the data inner structure. The combination strategy of DN leaves in order to retrieve a partition of the regular instances described in Section 2.3 is now assessed.

Baselines and Experimental Setting

CADI is compared with two baselines. The first baseline is the robust density-based clustering algorithm DBSCAN [Est+96]. The second one is the robust clustering algorithm $k-$means-- [CG13]. Both approaches were described in Sec. 1.4.

The choice of DBSCAN for comparison is motivated by three main reasons:

1. DBSCAN, like CADI, automatically discovers the number of clusters and therefore does not require it as an input parameter.
2. DBSCAN is able to discover non-elliptical clusters.

$k-$means-- on the other hand are a variant of the $k-$means clustering algorithm detecting clusters and anomalies simultaneously. It has two input parameters: the number k of clusters to produce and the number l of outliers to identify. At each step of the algorithm, $k-$means-- discard the l farthest instances when updating the clusters centers. The implementation of DBSCAN from scikit-learn [3] is used. The implementation of $k-$means-- used is part of the library ELKI [4].

The data sets from Fig. 3.5 are still employed.

As $k-$means-- takes as inputs the number of clusters and the number of anomalies to identify, the true values are passed as parameters to the algorithm. The most important hyper-parameter of DBSCAN, ϵ, controls the size of the neighborhood. The default value of ϵ is 0.5.

Results

The best ARI obtained when using CADI, DBSCAN and $k-$means-- on the data sets are reported in Table 3.10. These results are obtained sometimes after tweaking the hyper-parameters. The optimized values of the hyper-parameters are specified when different to the default ones.

Table 3.10 – Clustering performance: ARI

Data set	CADI	DBSCAN	$k-$means--
$\mathcal{D}_1$	**1.0**	0.968	**1.0**
$\mathcal{D}_2$	0.920	**0.990**	0.973
$\mathcal{D}_3$	**0.980**	0.961 ($\epsilon = 1.5$)	0.405
$\mathcal{D}_4$	0.957 ($\tau = 25^{th}$ *percentile*)	**0.996** ($\epsilon = 0.25$)	0.316
$\mathcal{D}_5$	0.871	**0.975**	0.338
Mean	0.946	**0.978**	0.606

On the data set $\mathcal{D}_1$ which is quite easy to cluster, CADI and $k-$means-- obtain a perfect ARI. DBSCAN obtains a slightly lower ARI, but this is due to the fact that it

3. https://scikit-learn.org/stable/modules/clustering.html#dbscan
4. https://elki-project.github.io

considers the anomalous cluster as a regular cluster. On $\mathcal{D}_3$ and $\mathcal{D}_5$, the same phenomenon observed when combining the Euclidean distance and the AHC occurs with $k-$means--, hence the poor performance on these data sets. $k-$means-- also struggles with $\mathcal{D}_4$, as the clusters are non-elliptical. DBSCAN and CADI are however able to discover these non-elliptical clusters. Noteworthy is that the threshold τ on the edges of CADI's graph was set to the 25^{th} percentile on this data set. The method was not able to discover two clusters with the default value $\tau = 0$. DBSCAN was also not able to discover two clusters with $\epsilon = 0.5$. The value had to be lowered to 0.25. The anomalous cluster was also treated like a regular cluster, hence the obtained ARI. On the data set $\mathcal{D}_3$, the parameter of DBSCAN was also fine-tuned. The default value $\epsilon = 0.5$ resulted in an ARI of 0.490. The ARI from Table 3.10 was obtained when setting ϵ to 1.5. The abnormal clusters were again considered regular by DBSCAN, lowering the ARI. Finally, the ARIs of CADI on $\mathcal{D}_2$ and $\mathcal{D}_5$ were also lowered because the anomaly score threshold: many instances were considered anomalous. With a higher threshold, better ARIs are obtained on these two data sets.

The two hyper-parameters of $k-$means--, namely the number of clusters and the number of anomalies, are crucial. However, as an extension of the $k-$means algorithm, the approach struggles with non-elliptical clusters, and modifying these hyper-parameters does not solve this issue. Conversely, DBSCAN and CADI can detect non-elliptical patterns. Furthermore, they do not require as input the number of clusters. However, the other inputs (the weights threshold τ for CADI and the size ϵ of the neighborhood) affect the number of clusters discovered by the method. Nevertheless, τ seems to have less impact on CADI's performance than ϵ on DBSCAN's.

Assessment on a non-synthetic data set: *Iris*

In this final experiment related to clustering, we use a small but non-synthetic data set: the well-known data set *iris*. It is a 4-dimensional data set containing 150 instances and 3 clusters of 50 instances each. It is assumed that it does not contain any anomaly. Consequently, we will perform clustering on this data set under this hypothesis. This means that the anomaly score threshold is set to $\gamma = 1.0$.

Classic $k-$means obtains an ARI of 0.730 on the data set iris for $k = 3$. $k-$means-- follows the trend, with an ARI equal to 0.716 for $k = 3$ and $l = 0$. With the default hyper-parameters, CADI obtains an ARI of 0.216 on iris. When the depth limit is set to $\Psi - 1$ (fully grown trees), the ARI reaches 0.676. Increasing only the margin width

($\alpha = 10\%$ instead of 5%) also results in a performance increase, with the ARI reaching 0.709. As for DBSCAN, the ARI on the iris data set is 0.521 using the default value of ϵ. It increases to 0.568 with $\epsilon = 1$. However, the method only discovers two clusters. For lower values of ϵ, like 0.25, DBSCAN obtains three clusters, but more than half of the data set is considered outlying.

The performance improvement observed when building deeper trees confirms that the dimensionality of the data set should be taken into consideration when setting the tree depth limit h_{lim} of CADI. When h_{lim} is too restrictive, there are many DLN leaves that are not used during clustering, and less DN leaves. Increasing the margin width also produces more DN leaves, hence the performance improvement. The sensitivity of DBSCAN to the choice ϵ was also corroborated.

3.4 Local Anomaly Detection

This section of the experiments evaluates the local/contextual anomaly detection component of CADI, viz. its ability to position anomalies in relation to clusters of regular instances as detailed in §2.4.1.

3.4.1 Data sets and Experimental Setting

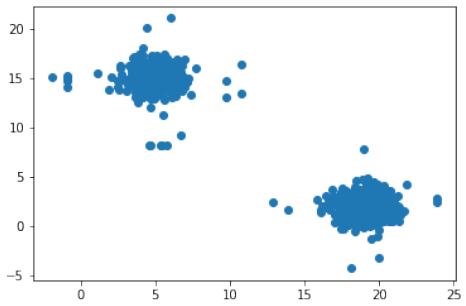
Unfortunately, there is no information on the real-world data sets from Table 3.1 on the locality of anomalies. We will therefore use synthetic data sets in this section. These data sets contain clusters and anomalies, and for each anomaly we know from which cluster(s) it is deviating. These data sets are similar to the ones used for the clustering assessment. They are described in Table 3.11 and illustrated on Figure 3.9.

In $\mathcal{D}_6$, each anomaly is close to only one cluster. The outlying attributes in relation to the corresponding cluster are the ground-truths. In $\mathcal{D}_7$, some anomalies share some attributes values with more than one cluster. $\mathcal{D}_8$ contains anomalies that deviate from all the clusters. $\mathcal{D}_9$ contains, in addition to local and global anomalies, a cluster of anomalies. To evaluate the ability of CADI to discover non-spherical clusters, $\mathcal{D}_7$ and $\mathcal{D}_8$ contain stretched clusters. In addition to that, $\mathcal{D}_9$ is the moons data set, to which we manually added outliers.

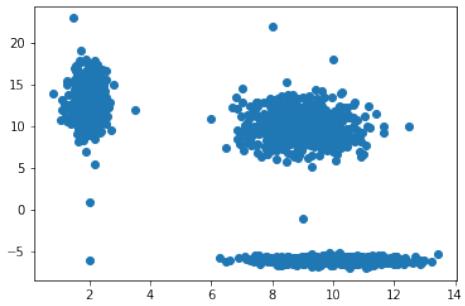
Anomaly detection and clustering are first performed on these data sets. The results of the clustering in terms of ARI are presented in Table 3.12. The performance is similar

Table 3.11 – Data sets for local anomaly detection

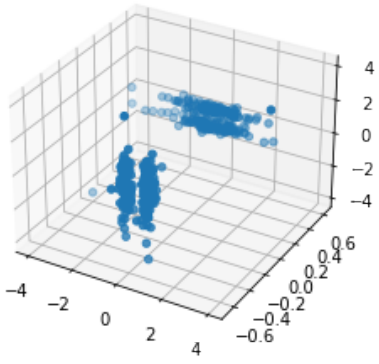
Data set	d	# clusters	n	# anomalies	Description
$\mathcal{D}_6$	2	2	900	25	Spherical clusters. Local anomalies only.
$\mathcal{D}_7$	2	3	1508	8	Two spherical and one stretched clusters. Local and global anomalies.
$\mathcal{D}_8$	3	4	408	8	Four stretched clusters. Each pair of clusters located in only two dimensions. Local and global anomalies.
$\mathcal{D}_9$	2	2	517	17	Two moons. One anomaly cluster. Local and global anomalies.



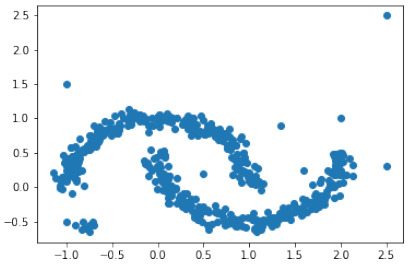
(a) $\mathcal{D}_6$



(b) $\mathcal{D}_7$



(c) $\mathcal{D}_8$



(d) $\mathcal{D}_9$

Figure 3.9 – Data sets for local anomaly detection

to the one observed in §3.3.3.

Table 3.12 – Clustering performance: ARI

Data set	CADI	DBSCAN	$k-$means--
$\mathcal{D}_6$	0.986	0.914	**1.0**
$\mathcal{D}_7$	0.970	0.963	**0.994**
$\mathcal{D}_8$	0.936	**0.971**	0.333
$\mathcal{D}_9$	0.992	**0.999**	0.287
Mean	**0.971**	0.962	0.653

$k-$means-- obtains the best ARI on $\mathcal{D}_6$ and $\mathcal{D}_7$. This behavior was expected, since the classic $k-$means algorithm is efficient for extracting spherical clusters. On $\mathcal{D}_8$ and $\mathcal{D}_9$, $k-$means-- struggles, just like the original $k-$means would do on these data sets. On $\mathcal{D}_8$, the same problem identified on Figure 3.8 with AHC combined to Euclidean distance is observed on $k-$means--. DBSCAN and CADI on the other hand do not struggle to discover non-elliptical clusters. CADI obtains the best ARI in average.

In addition to identifying anomalies and groups of regular data points, CADI provides some insight about the cluster(s) from which each anomaly may be deviating. As, to the best of our knowledge, no method in the literature is able to do so without relying on distances computations, there is no baseline for comparison. However, since the true cluster affectations are known for each anomaly in the generated data sets, the performance of CADI regarding the contextual anomaly detection can be evaluated. To do so, for each outlier o, let $\mathcal{P}$ be the set of predicted clusters for o, viz. the set of clusters from which o may be deviating according to CADI. Given an outlier o and the quantities $c(o, C_k)$ for $C_k \in \{C_1, ..., C_p\}$ (set of identified clusters) computed using Eq. 2.5, the set of predicted clusters for o is composed of the clusters C_k with maximum integer values of $c(o, C_k)$:

$$\mathcal{P} = argmax\lfloor c(o, C_k) \rfloor$$

Let $\mathcal{T}$ be the set of ground-truth clusters for o, viz. the set of clusters from which o is deviating. The precision and recall are computed as $Precision = |\mathcal{P} \cup \mathcal{T}|/|\mathcal{P}|$ and $Recall = |\mathcal{P} \cup \mathcal{T}|/|\mathcal{T}|$.

3.4.2 Results

For each data set $\mathcal{D}_6$ to $\mathcal{D}_9$, the precision and recall is averaged over the outliers. The results are shown in Table 3.13.

Table 3.13 – Contextual anomaly detection performance

Data set	Precision	Recall
$\mathcal{D}_6$	1.0	1.0
$\mathcal{D}_7$	1.0	1.0
$\mathcal{D}_8$	0.875	0.875
$\mathcal{D}_9$	0.941	1.0

CADI performs well on all the data sets, with perfect precision and recall on $\mathcal{D}_6$ and $\mathcal{D}_7$. The method is more challenged by $\mathcal{D}_8$, as this data set contains several data points deviating from more than one cluster.

3.5 Explanations By Structure Analysis

The last part of CADI's assessment is the evaluation of the generated contextual explanations.

3.5.1 Baselines and Experimental Setting

Evaluating an explanation on real-world data sets is not an easy task, because of the absence of ground-truths. With some knowledge about the data, it is possible to have an insight on these ground-truth explanations. Without that knowledge, some strategies must be developed. In the anomaly explanation literature, a common practice is to add so-called noise attributes [LSH18; CTS23]. The hypothesis behind this practice is that true explanations should lie among the original (viz. non-noise) attributes. We believe that an evaluation using this scheme is not faithful enough, since the true outlying attributes must be part of the original ones. This scheme therefore only evaluates the ability of a method to provide non-aberrant explanations. In [Xu+21], another technique to generate ground-truth explanations on real-world data sets is proposed. The outlying degree/score of true outliers in every possible subspace of the original feature space is computed using an anomaly detector. The ground-truth explanation is the subspace where the anomaly received the highest score. Three different anomaly detectors are used, among which the IF. Depending on the detector used, there are different ground-truth outlying attributes that are used separately during the evaluation. With this scheme, there is no information regarding the clusters of regular data points if any.

In contrast to real-world data sets, the true outlying attributes are known during the

generation of synthetic data sets. Furthermore, since the generation process is known, data sets containing clusters and local anomalies can be produced. It is the case for the data sets from Table 3.11. We followed a strategy similar to the one described in [LSH18] for the generation of synthetic data sets.

CADI outputs as explanation for an outlier o a list of discriminative feature weights with each identified cluster (e_{disc} from Eq. 2.7). COIN [LSH18] also outputs a list of feature weights, but with respect to the local context of o only, points close to o, and not the set of clusters in the data set. ATON [Xu+21] also outputs a list of feature weights, but does not return the local context of o as explanation like COIN, even though it is used to compute feature importance scores. Both COIN and ATON need the outlier o to explain as input to the methods. It is not the case for CADI which performs anomaly detection prior to the explanation. COIN and ATON will nonetheless serve as baseline for the evaluation. In addition to these two, CADI is compared to the ground-truth explanation extraction procedure introduced in [Xu+21]. The IF is used as detector for this method called ATON-GT from here onwards. ATON-GT outputs for each specified outlier o, the subspace in which it received the highest anomaly score. The implementations of COIN, ATON and ATON-GT were made available by the respective authors.

Considering all the above-mentioned differences across CADI, COIN, ATON and ATON-GT, we propose the following evaluation procedure to provide a fair comparison:

— Since COIN, ATON and ATON-GT do not identify outliers *per se*, explanations for *true* outliers only are requested from all the four methods. This allows to also evaluate the ability of CADI to provide accurate explanations even for outliers the method was not able to identify as such.

— The ground-truth explanations are a list of discriminating attributes with respect to each cluster. As a result, for the methods outputting feature importance scores (CADI, COIN and ATON), the top k discriminating features are retrieved, with k being the length of the *true* explanatory subspace.

— For all four methods, the precision and recall of the explanations are computed in a similar manner as during contextual anomaly detection performance evaluation (Sec. 3.4). If $\mathcal{T}$ is the ground-truth attribute subspace and $\mathcal{P}$ is the predicted attribute subspace, then the precision for a given outlier o is $|\mathcal{T} \cap \mathcal{P}|/|\mathcal{P}|$. The recall is computed for a single instance o with the formula $|\mathcal{T} \cap \mathcal{P}|/|\mathcal{T}|$. This procedure is also used in [Xu+21]. For each data set, the precision and recall over all the outliers are computed.

— As CADI should not only produce the good cluster(s) but also the good explanatory subspaces with respect to these clusters, the two information should match. Consequently, during the precision and recall computation, the predicted cluster is compared to the ground-truth cluster first, before comparing the explanatory subspaces.

— For COIN, ATON and ATON-GT, the generated explanatory subspace is compared with the explanatory subspace of o with respect to the true cluster(s) from which it is deviating, as these methods do not indicate from which cluster o may be deviating.

3.5.2 Results

The precision and recall for each data set and each method are shown in Table 3.14.

Table 3.14 – Outlier interpretation performance

	Precision				Recall			
	CADI	COIN	ATON	ATON-GT	CADI	COIN	ATON	ATON-GT
$\mathcal{D}_6$	**1.0**	0.76	0.76	0.76	**1.0**	0.76	0.76	**1.0**
$\mathcal{D}_7$	**1.0**	0.81	0.60	0.81	**1.0**	0.81	0.69	**1.0**
$\mathcal{D}_8$	0.89	0.69	0.87	**1.0**	**0.89**	0.69	0.87	0.75
$\mathcal{D}_9$	**1.0**	**1.0**	0.94	0.67	**1.0**	**1.0**	0.94	0.91

CADI has a high precision and recall on all the data sets, meaning that it is able not only to identify the cluster(s) from which an instance is deviating, but also to provide a faithful explanation in relation to these clusters in terms of discriminative attributes. COIN performs better than ATON on $\mathcal{D}_6$, $\mathcal{D}_7$ and $\mathcal{D}_9$. This may be because of the clustering step of COIN that allows to mitigate the influence of different group of points on the attributes importance. ATON and ATON-GT perform better than COIN on $\mathcal{D}_8$. In this data set, clusters are located in different subspaces and local anomalies can also be identified in these subspaces. And, as ATON-GT explores different subspaces during the explanation generation process, it has a slight advantage. CADI is also able to discover clusters (and consequently outliers) in subspaces because of the split generation procedure. As a result, it does not fall far behind ATON-GT in terms of precision on $\mathcal{D}_8$. In general, ATON-GT has a high recall, because it tends to overestimate the size of the explanatory subspace. For example, on the data set $\mathcal{D}_6$, the explanatory subspace returned by ATON-GT for the red square outlier on Figure 3.10 is the full attribute space. Although only

feature A_1 is sufficient (regardless of the local context or not), that outlier is more easily isolated in the full feature space than in A_1.

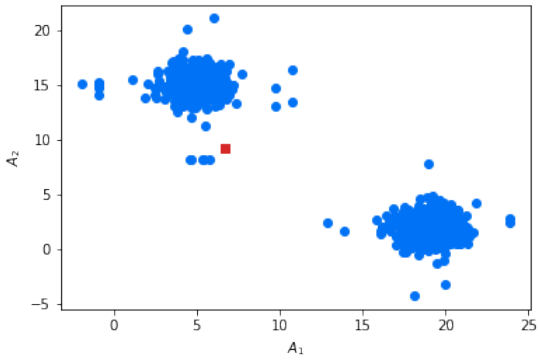


Figure 3.10 – Data set $\mathcal{D}_6$. For the outlier represented by a red square, ATON-GT returns as explanatory subspace the full feature space.

3.6 Summary

The objective of this chapter was to evaluate CADI. Due to the lack of unified method to perform anomaly detection, data clustering and generate explanations of anomalies in relation to clusters of regular data points, each component of CADI was compared to different baselines on several data sets. It appeared that CADI is competitive against other unsupervised anomaly detection algorithms, clustering algorithms and anomaly explanation approaches, and sometimes clearly more effective, while possessing the advantage that it does not rely on external methods. It was also shown that the results are relatively stable with the choice of the hyper-parameters. However, on higher dimensional data sets, it was observed that the depth limit has an influence on the clustering results, because of the small number of DN leaves that are combined to retrieve a partition. Thus, a research perspective would be to investigate the impact of the depth limit on the clustering results, and to propose a depth limit threshold which is a function of the dimensionality of the data set. Moreover, as the DLN leaves are not employed for clustering because they convey only a partial information (the points could not be separated after h_{lim} number of splits), it may also be worth investigating their combination when h_{lim} is not a function of the dimensionality.

Conclusion and Perspectives

Summary

In this dissertation, we addressed three scientific problems, namely anomaly detection, clustering and anomaly explanation, with a unified method called CADI. CADI, which relies on a modified version of the Isolation Forest algorithm, performs the three aforementioned tasks without relying on external algorithms. In contrast to the existing anomaly explanation approaches, CADI produces explanations about anomalies in relation to clusters of regular data, explanations which are important in the context of the Sea Defender project of which this thesis is a part. To extract these explanations, a split selection criterion is introduced during the construction of isolation trees to avoid as much as possible the separation of dense regions of points. A CADI tree therefore has three types of leaves, among which the leaves containing points that could not be separated are of paramount importance when trying to retrieve a partition of the regular instances. Whereas the so-called IN leaves containing isolated instances like in the classic IF approach are important for anomaly detection, the former terminal nodes called DN leaves correspond to portions of clusters that can be combined to retrieve a partition of the data set. The combination strategy is adapted from the one used in grid-based clustering and therefore allows to retrieve non-elliptical clusters and automatically discover the number of clusters in the data set. The same trees are finally analyzed to position the identified anomalies in relation to clusters, and extract contextual explanations of the anomalies in relation to the clusters.

Experiments conducted on real-world data sets for anomaly detection, and mostly synthetic data sets for the other components, showed that CADI demonstrates superior results compared to other unsupervised anomaly detection methods for anomaly detection, to robust clustering algorithms for clustering, and to anomaly explanation approaches taking into account the local context of the anomaly to explain. It was also shown that CADI is able to correctly position the anomalies in relation to clusters, therefore filling a gap in the local anomaly detection literature. The CADI approach was our second contribution, the first one being a taxonomy of anomaly explanation methods. Whereas

the existing taxonomies focus on the explanation extraction method, our taxonomy focus on the information conveyed by generated explanations. Four categories of explanations are therefore proposed, among which explanations by structure analysis, scarcely explored in the literature, is the one produced by CADI.

Limitations and Future Directions

The work presented in this dissertation can be improved and extended in several ways. First of all, besides the anomaly detection component, the approach has not been assessed on high dimensional data sets. This requires the generation of higher dimensional data sets containing anomalies and clusters, and ground-truth knowledge about the context and explanation of each anomaly. It is the primary future direction of this work. Along the same lines, the impact of the depth limit threshold h_{lim} has not been thoroughly investigated, as well as its relationship with the value of the hyper-parameter. During the experiments related to clustering (Sec. 3.3), it was observed that this parameter may be important when the dimensionality of the data set increases. In the original IF method and in most of its variants (including CADI), the depth limit is not set by taking into account the dimensionality of the data set. However, in [LTZ10] and [Cor21], it was suggested that deeper trees may produce better results even for anomaly detection. Still related to the depth of the trees, CADI, in contrast to classic IF, does not leverage the depth of the node containing the instance in the computation of its anomaly score (Eq. 2.2). It may be worth examining the combination of the depth information and the cardinality information during the computation of anomaly scores.

Explainable Artificial Intelligence has been trending for several years now. However, the evaluation of the explanations generated is still an open problem. As XAI is at the crossroad of AI and cognitive sciences, there are mainly two categories of approaches to evaluate explanations [VL20]: objective explanations which employ objective metrics, and human-centered evaluations relying on user studies which are therefore subjective. The objective metrics include the correctness and the length of the explanation, for instance. However, since explanations are intended for humans, they should not be left out during the evaluation stage, hence the need for user studies to make sure that the targets of the explanations are satisfied. In the anomaly explanation literature and in this work, only the correctness/fidelity of the explanations with respect to the true explanations is generally assessed [MCS21]. Nevertheless, to acknowledge the subjective nature of explanations, it may be of interest to devise objective metrics (like in [Nau+23], but with an emphasis

on anomaly explanation) among which the user can select the most appropriate ones according to his/her needs.

From an application perspective, although this thesis was part of the Sea Defender project, CADI has not yet been employed on international trade data, because the banks are reluctant to share their data, waiting to test the tools first. This application of CADI on the real-world data it was designed for, as well as the exploration of human knowledge integration (for example during the selection of the hyper-parameter α) are also two perspectives of this work.

BIBLIOGRAPHY

[AAB19] Samet Akcay, Amir Atapour-Abarghouei, and Toby P. Breckon, « GANomaly: Semi-supervised Anomaly Detection via Adversarial Training », *in*: *Computer Vision – ACCV 2018*, Springer International Publishing, 2019, pp. 622–637, ISBN: 978-3-030-20893-6, DOI: 10.1007/978-3-030-20893-6_39.

[AGA13] Mennatallah Amer, Markus Goldstein, and Slim Abdennadher, « Enhancing One-Class Support Vector Machines for Unsupervised Anomaly Detection », *in*: *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description*, Association for Computing Machinery, 2013, pp. 8–15, ISBN: 9781450323352, DOI: 10.1145/2500853.2500857.

[Agg16] Charu C. Aggarwal, *Outlier analysis second edition*, Springer, 2016, ISBN: 978-3-319-47578-3, DOI: 10.1007/978-3-319-47578-3.

[Agr+98] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan, « Automatic subspace clustering of high dimensional data for data mining applications », *in*: *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, Association for Computing Machinery, 1998, pp. 94–105, ISBN: 0897919955, DOI: 10.1145/276304.276314.

[AKM18] Kasun Amarasinghe, Kevin Kenney, and Milos Manic, « Toward explainable deep neural network based anomaly detection », *in*: *2018 11th International Conference on Human System Interaction (HSI)*, IEEE, 2018, pp. 311–317, DOI: 10.1109/HSI.2018.8430788.

[Ako21] Leman Akoglu, « Anomaly Mining: Past, Present and Future », *in*: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, Association for Computing Machinery, 2021, pp. 1–2, ISBN: 9781450384469, DOI: 10.1145/3459637.3482495.

[Alf+20] Antonio L. Alfeo, Mario G.C.A. Cimino, Giuseppe Manco, Ettore Ritacco, and Gigliola Vaglini, « Using an autoencoder in the design of an anomaly detector for smart manufacturing », *in*: *Pattern Recognition Letters* 136 (2020),

pp. 272–278, ISSN: 0167-8655, DOI: https://doi.org/10.1016/j.patrec.2020.06.008.

[AP02] Fabrizio Angiulli and Clara Pizzuti, « Fast Outlier Detection in High Dimensional Spaces », *in*: *Principles of Data Mining and Knowledge Discovery*, 2002, pp. 15–27, ISBN: 978-3-540-45681-0.

[AR14] Charu C. Aggarwal and Chandan K. Reddy, eds., *Data Clustering: Algorithms and Applications*, CRC Press, 2014, ISBN: 978-1-46-655821-2.

[ASR19] Liat Antwarg, Bracha Shapira, and Lior Rokach, « Explaining anomalies detected by autoencoders using SHAP », *in*: *arXiv preprint arXiv:1903.02407* (2019).

[Bar+22] Tommaso Barbariol, Filippo Dalla Chiara, Davide Marcato, and Gian Antonio Susto, « A Review of Tree-Based Approaches for Anomaly Detection », *in*: (2022), pp. 149–185, DOI: 10.1007/978-3-030-83819-5_7.

[Bas+16] Elisabeth Baseman, Sean Blanchard, Nathan DeBardeleben, Amanda Bonnie, and Adam 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, pp. 1–27.

[BCB22] Alberto Barbado, Óscar Corcho, and Richard Benjamins, « Rule extraction in unsupervised anomaly detection for model explainability: Application to OneClass SVM », *in*: *Expert Systems with Applications* 189 (2022), ISSN: 0957-4174, DOI: 10.1016/j.eswa.2021.116100.

[Bor+15] Christian Borgelt, Christian Braune, Marie-Jeanne Lesot, and Rudolf Kruse, « Handling Noise and Outliers in Fuzzy Clustering », *in*: *Fifty Years of Fuzzy Logic and its Applications*, Cham: Springer International Publishing, 2015, pp. 315–335, ISBN: 978-3-319-19683-1, DOI: 10.1007/978-3-319-19683-1_17.

[Bre+00] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander, « LOF: Identifying Density-Based Local Outliers », *in*: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, Association for Computing Machinery, 2000, pp. 93–104, ISBN: 1581132174, DOI: 10.1145/342009.335388.

[Bro+18] Andy Brown, Aaron Tuor, Brian Hutchinson, and Nicole Nichols, « Recurrent Neural Network Attention Mechanisms for Interpretable System Log Anomaly Detection », *in*: *Proceedings of the First Workshop on Machine Learning for Computing Systems*, Association for Computing Machinery, 2018, pp. 1–8, ISBN: 9781450358651, DOI: 10.1145/3217871.3217872.

[CBK09] Varun Chandola, Arindam Banerjee, and Vipin Kumar, « Anomaly Detection: A Survey », *in*: *ACM Comput. Surv.* 41.*3* (2009), ISSN: 0360-0300, DOI: 10.1145/1541880.1541882.

[CG13] Sanjay Chawla and Aristides Gionis, « k-means--: A unified approach to clustering and outlier detection », *in*: *Proceedings of the 2013 SIAM International Conference on Data Mining (SDM)*, 2013, pp. 189–197, DOI: 10.1137/1.9781611972832.21.

[CGR20] Siddharth Chaurasia, Sagar Goyal, and Manish 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, DOI: 10.1109/IC3A48958.2020.233273.

[Cha+22] Yousra Chabchoub, Maurras Ulbricht Togbe, Aliou Boly, and Raja Chiky, « An In-Depth Study and Improvement of Isolation Forest », *in*: *IEEE Access* 10 (2022), pp. 10219–10237, DOI: 10.1109/ACCESS.2022.3144425.

[Cha+23] Chun-Hao Chang, Jinsung Yoon, Sercan Ö Arik, Madeleine Udell, and Tomas Pfister, « Data-efficient and interpretable tabular anomaly detection », *in*: *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, 2023, pp. 190–201, ISBN: 9798400701030, DOI: 10.1145/3580305.3599294.

[Che+] Jinghui Chen, Saket Sathe, Charu Aggarwal, and Deepak Turaga, « Outlier Detection with Autoencoder Ensembles », *in*: *Proceedings of the 2017 SIAM International Conference on Data Mining (SDM)*, pp. 90–98, DOI: 10.1137/1.9781611974973.11.

[Cor21] David Cortes, « Revisiting randomized choices in isolation forests », *in*: *arXiv preprint arXiv:2110.13402* (2021).

[CTS23] Mattia Carletti, Matteo Terzi, and Gian Antonio Susto, « Interpretable anomaly detection with diffi: Depth-based feature importance of isolation forest », *in*: *Engineering Applications of Artificial Intelligence* 119 (2023), p. 105730, ISSN: 0952-1976, DOI: 10.1016/j.engappai.2022.105730.

[Dan+13] Xuan Hong Dang, Barbora Micenková, Ira Assent, and Raymond T. Ng, « Local Outlier Detection with Interpretation », *in*: *Machine Learning and Knowledge Discovery in Databases*, Springer Berlin Heidelberg, 2013, pp. 304–320, ISBN: 978-3-642-40994-3, DOI: 10.1007/978-3-642-40994-3_20.

[Dan+14] Xuan Hong Dang, Ira Assent, Raymond T Ng, Arthur Zimek, and Erich Schubert, « Discriminative features for identifying and interpreting outliers », *in*: *2014 IEEE 30th international conference on data engineering*, IEEE, 2014, pp. 88–99, DOI: 10.1109/ICDE.2014.6816642.

[Dav91] Rajesh N Dave, « Characterization and detection of noise in clustering », *in*: *Pattern Recognition Letters* 12.*11* (1991), pp. 657–664, ISSN: 0167-8655, DOI: 10.1016/0167-8655(91)90002-4.

[Dot+18] Francesco Dotto, Alessio Farcomeni, Luis Angel Garcıa-Escudero, and Agustın Mayo-Iscar, « A reweighting approach to robust clustering », *in*: *Statistics and Computing* 28.*2* (2018), pp. 477–493, ISSN: 1573-1375, DOI: 10.1007/s11222-017-9742-x.

[Est+96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al., « A density-based algorithm for discovering clusters in large spatial databases with noise. », *in*: *kdd*, vol. 96, 34, 1996, pp. 226–231.

[FK96] Hichem Frigui and Raghu Krishnapuram, « A robust algorithm for automatic extraction of an unknown number of clusters from noisy data », *in*: *Pattern Recognition Letters* 17.*12* (1996), pp. 1223–1232, ISSN: 0167-8655, DOI: 10.1016/0167-8655(96)00080-3.

[GA19] David Gunning and David W. Aha, « DARPA's explainable artificial intelligence program », *in*: *AI Magazine* 40.*2* (2019), pp. 44–58.

[GJ01] Patrick J.F. Groenen and Krzysztof Jajuga, « Fuzzy clustering with squared Minkowski distances », *in*: *Fuzzy Sets and Systems* 120.*2* (2001), pp. 227–237, ISSN: 0165-0114, DOI: 10.1016/S0165-0114(98)00403-5.

[GS19] Ioana Giurgiu and Anika Schumann, « Additive Explanations for Anomalies Detected from Multivariate Temporal Data », *in*: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, Association for Computing Machinery, 2019, pp. 2245–2248, ISBN: 9781450369763, DOI: 10.1145/3357384.3358121.

[GSW19] Parikshit Gopalan, Vatsal Sharan, and Udi Wieder, « PIDForest: Anomaly Detection via Partial Identification », *in*: 32 (2019), pp. 15809–15819.

[GU16] Markus Goldstein and Seiichi Uchida, « A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data », *in*: *PLOS ONE* 11.*4* (Apr. 2016), pp. 1–31, DOI: 10.1371/journal.pone.0152173.

[Guh+16] Sudipto Guha, Nina Mishra, Gourav Roy, and Okke Schrijvers, « Robust Random Cut Forest Based Anomaly Detection on Streams », *in*: *Proceedings of The 33rd International Conference on Machine Learning*, vol. 48, Proceedings of Machine Learning Research, New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 2712–2721.

[Gup+19] Nikhil Gupta, Dhivya Eswaran, Neil Shah, Leman Akoglu, and Christos Faloutsos, « Beyond Outlier Detection: LookOut for Pictorial Explanation », *in*: *Machine Learning and Knowledge Discovery in Databases*, Springer International Publishing, 2019, pp. 122–138, ISBN: 978-3-030-10925-7, DOI: 10.1007/978-3-030-10925-7_8.

[Han+22] Songqiao Han, Xiyang Hu, Hailiang Huang, Minqi Jiang, and Yue Zhao, « ADBench: Anomaly Detection Benchmark », *in*: *Advances in Neural Information Processing Systems*, ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, vol. 35, Curran Associates, Inc., 2022, pp. 32142–32159.

[Haw+02] Simon Hawkins, Hongxing He, Graham Williams, and Rohan Baxter, « Outlier Detection Using Replicator Neural Networks », *in*: *Data Warehousing and Knowledge Discovery*, Springer Berlin Heidelberg, 2002, pp. 170–180, ISBN: 978-3-540-46145-6.

[Haw80] Douglas M. Hawkins, *Identification of outliers*, vol. 11, Springer, 1980.

[HJS21] Swastik Haldar, Philips George John, and Diptikalyan Saha, « Reliable Counterfactual Explanations for Autoencoder Based Anomalies », *in*: *Proceedings of the 3rd ACM India Joint International Conference on Data Science & Management of Data (8th ACM IKDD CODS & 26th COMAD)*, Association for Computing Machinery, 2021, pp. 83–91, ISBN: 9781450388177, DOI: 10.1145/3430984.3431015.

[HKB21] Sahand Hariri, Matias Carrasco Kind, and Robert J. Brunner, « Extended Isolation Forest », *in*: *IEEE Transactions on Knowledge and Data Engineering* 33.*4* (2021), pp. 1479–1489, DOI: 10.1109/TKDE.2019.2947676.

[HM82] James A Hanley and Barbara J McNeil, « The meaning and use of the area under a receiver operating characteristic (ROC) curve. », *in*: *Radiology* 143.*1* (1982), pp. 29–36.

[HP19] Michael Hanni and Andrea Podestá, « Trade misinvoicing in copper products: a case study of Chile and Peru », *in*: *CEPAL Review* (2019).

[HXD03] Zengyou He, Xiaofei Xu, and Shengchun Deng, « Discovering cluster-based local outliers », *in*: *Pattern Recognition Letters* 24.*9* (2003), pp. 1641–1650, ISSN: 0167-8655, DOI: 10.1016/S0167-8655(03)00003-5.

[Jaj91] Krzysztof Jajuga, « L1-norm based fuzzy clustering », *in*: *Fuzzy Sets and Systems* 39.*1* (1991), pp. 43–50, ISSN: 0165-0114, DOI: 10.1016/0165-0114(91)90064-W.

[KK96] R. Krishnapuram and J.M. Keller, « The possibilistic C-means algorithm: insights and recommendations », *in*: *IEEE Transactions on Fuzzy Systems* 4.*3* (1996), pp. 385–393, DOI: 10.1109/91.531779.

[KMM20] Jacob Kauffmann, Klaus-Robert Müller, and Grégoire Montavon, « Towards explaining anomalies: A deep Taylor decomposition of one-class models », *in*: *Pattern Recognition* 101 (2020), ISSN: 0031-3203, DOI: 10.1016/j.patcog.2020.107198.

[KN99] Edwin M. Knorr and Raymond T. Ng, « Finding intensional knowledge of distance-based outliers », *in*: *Very Large Data Bases Conference*, vol. 99, Citeseer, 1999, pp. 211–222.

[KPH20] Martin Kopp, Tomáš Pevnỳ, and Martin Holeňa, « Anomaly explanation with random forests », *in*: *Expert Systems with Applications* 149 (2020), ISSN: 0957-4174, DOI: 10.1016/j.eswa.2020.113187.

[Kri+09] Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek, « Outlier Detection in Axis-Parallel Subspaces of High Dimensional Data », *in*: *Advances in Knowledge Discovery and Data Mining*, Springer Berlin Heidelberg, 2009, pp. 831–838, ISBN: 978-3-642-01307-2, DOI: 10.1007/978-3-642-01307-2_86.

[Kri+12] Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek, « Outlier detection in arbitrarily oriented subspaces », *in*: *2012 IEEE 12th international conference on data mining*, IEEE, 2012, pp. 379–388, DOI: 10.1109/ICDM.2012.21.

[KSZ08] Hans-Peter Kriegel, Matthias Schubert, and Arthur Zimek, « Angle-Based Outlier Detection in High-Dimensional Data », *in*: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, 2008, pp. 444–452, ISBN: 9781605581934, DOI: 10.1145/1401890.1401946.

[Li+18] Zhenchuan Li, Guanjun Liu, Shuo Wang, Shiyang Xuan, and Changjun 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, DOI: 10.1109/SmartWorld.2018.00217.

[Li+22] Zheng Li, Yue Zhao, Xiyang Hu, Nicola Botta, Cezar Ionescu, and George Chen, « Ecod: Unsupervised outlier detection using empirical cumulative distribution functions », *in*: *IEEE Transactions on Knowledge and Data Engineering* (2022), DOI: 10.1109/TKDE.2022.3159580.

[Liu+20] Haoyu Liu, Fenglong Ma, Yaqing Wang, Shibo He, Jiming Chen, and Jing Gao, « LP-Explain: Local Pictorial Explanation for Outliers », *in*: *2020 IEEE International Conference on Data Mining (ICDM)*, IEEE, 2020, pp. 372–381, DOI: 10.1109/ICDM50108.2020.00046.

[Liu+21] Hongfu Liu, Jun Li, Yue Wu, and Yun Fu, « Clustering With Outlier Removal », *in*: *IEEE Transactions on Knowledge and Data Engineering* 33.*6* (2021), pp. 2369–2379, DOI: 10.1109/TKDE.2019.2954317.

[LL17] Scott M Lundberg and Su-In Lee, « A Unified Approach to Interpreting Model Predictions », *in*: *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017.

[LL23] Zhong Li and Matthijs van Leeuwen, « Robust and Explainable Contextual Anomaly Detection using Quantile Regression Forests », *in*: *arXiv preprint arXiv:2302.11239* (2023).

[LP16] Jiongqian Liang and Srinivasan Parthasarathy, « Robust Contextual Outlier Detection: Where Context Meets Sparsity », *in*: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, Association for Computing Machinery, 2016, pp. 2167–2172, ISBN: 9781450340731, DOI: 10.1145/2983323.2983660.

[LSH18] Ninghao Liu, Donghwa Shin, and Xia Hu, « Contextual outlier interpretation », *in*: *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, AAAI Press, 2018, pp. 2461–2467, ISBN: 9780999241127, DOI: 10.5555/3304889.3305002.

[LTZ10] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou, « On detecting clustered anomalies using SCiForest », *in*: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2010, pp. 274–290, ISBN: 978-3-642-15883-4.

[LTZ12] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou, « Isolation-based anomaly detection », *in*: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 6.*1* (2012), pp. 1–39, ISSN: 1556-4681, DOI: 10.1145/2133360.2133363.

[LZV23] Zhong Li, Yuxuan Zhu, and Matthijs Van Leeuwen, « A Survey on Explainable Anomaly Detection », *in*: *ACM Trans. Knowl. Discov. Data* 18.*1* (Sept. 2023), ISSN: 1556-4681, DOI: 10.1145/3609333.

[MA18] Meghanath Macha and Leman Akoglu, « Explaining anomalies in groups with characterizing subspace rules », *in*: *Data Mining and Knowledge Discovery* 32.*5* (2018), pp. 1444–1480, DOI: 10.1007/s10618-018-0585-7.

[MB21] Antonella Mensi and Manuele Bicego, « Enhanced anomaly scores for isolation forests », *in*: *Pattern Recognition* 120 (2021).

[MCS21] Nikolaos Myrtakis, Vassilis Christophides, and Eric Simon, « A comparative evaluation of anomaly explanation algorithms », *in*: *24th International Conference on Extending Database Technology (EDBT'2021)*, 2021.

[Mej10] Manuel 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, DOI: 10.1109/CERMA.2010.12.

[Mia+19] Zhengjie Miao, Qitian Zeng, Chenjie Li, Boris Glavic, Oliver Kennedy, and Sudeepa Roy, « CAPE: Explaining Outliers by Counterbalancing », *in*: *Proc. VLDB Endow.* 12.*12* (Aug. 2019), pp. 1806–1809, ISSN: 2150-8097, DOI: 10.14778/3352063.3352071.

[Mic+13] Barbora Micenková, Raymond T Ng, Xuan-Hong Dang, and Ira Assent, « Explaining outliers by subspace separability », *in*: *2013 IEEE 13th international conference on data mining*, IEEE, 2013, pp. 518–527, DOI: 10.1109/ICDM.2013.132.

[Mil19] Tim Miller, « Explanation in artificial intelligence: Insights from the social sciences », *in*: *Artificial intelligence* 267 (2019), pp. 1–38, ISSN: 0004-3702, DOI: 10.1016/j.artint.2018.07.007.

[MLC07] Gerhard Münz, Sa Li, and Georg Carle, « Traffic anomaly detection using k-means clustering », *in*: *GI/ITG Workshop MMBnet*, 2007, pp. 13–14.

[Mok19] Tshepiso Mokoena, « Why is this an anomaly? Explaining anomalies using sequential explanations », PhD thesis, 2019.

[MP03] Junshui Ma and Simon Perkins, « Online Novelty Detection on Temporal Sequences », *in*: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003, pp. 613–618, ISBN: 1581137370, DOI: 10.1145/956750.956828.

[MP98] Geoffrey J. McLachlan and David Peel, « Robust cluster analysis via mixtures of multivariate t-distributions », *in*: *Advances in Pattern Recognition*, Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 658–666, ISBN: 978-3-540-68526-5.

[Mun+19] Mohsin Munir, Shoaib Ahmed Siddiqui, Ferdinand Küsters, Dominique Mercier, Andreas Dengel, and Sheraz Ahmed, « TSXplain: Demystification of DNN Decisions for Time-Series Using Natural Language and Statistical Features », *in*: *Artificial Neural Networks and Machine Learning – ICANN 2019: Workshop and Special Sessions*, Springer International Publishing, 2019, pp. 426–439, ISBN: 978-3-030-30493-5, DOI: 10.1007/978-3-030-30493-5_43.

[Nau+23] Meike Nauta, Jan Trienes, Shreyasi Pathak, Elisa Nguyen, Michelle Peters, Yasmin Schmitt, Jörg Schlötterer, Maurice van Keulen, and Christin Seifert, « From Anecdotal Evidence to Quantitative Evaluation Methods: A Systematic Review on Evaluating Explainable AI », *in*: *ACM Comput. Surv.* 55.*13s* (July 2023), ISSN: 0360-0300, DOI: 10.1145/3583558.

[Ngu+19] Quoc Phong Nguyen, Kar Wai Lim, Dinil Mon Divakaran, Kian Hsiang Low, and Mun Choon 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, DOI: 10.1109/CNS.2019.8802833.

[Pal+05] N.R. Pal, K. Pal, J.M. Keller, and J.C. Bezdek, « A possibilistic fuzzy c-means clustering algorithm », *in*: *IEEE Transactions on Fuzzy Systems* 13.*4* (2005), pp. 517–530, DOI: 10.1109/TFUZZ.2004.840099.

[Pan+21] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel, « Deep Learning for Anomaly Detection: A Review », *in*: *ACM Comput. Surv.* 54.*2* (Mar. 2021), ISSN: 0360-0300, DOI: 10.1145/3439950.

[Pan+22] Egawati Panjei, Le Gruenwald, Eleazar Leal, Christopher Nguyen, and Shejuti Silvia, « A survey on outlier explanations », *in*: *The VLDB Journal* 31.*5* (2022), pp. 977–1008, DOI: 10.1007/s00778-021-00721-1.

[PHL04] Lance Parsons, Ehtesham Haque, and Huan Liu, « Subspace Clustering for High Dimensional Data: A Review », *in*: *SIGKDD Explor. Newsl.* 6.*1* (June 2004), pp. 90–105, ISSN: 1931-0145, DOI: 10.1145/1007730.1007731.

[PW10] Wolfgang Polonik and Zailong Wang, « PRIM analysis », *in*: *Journal of Multivariate Analysis* 101.*3* (2010), pp. 525–540, ISSN: 0047-259X, DOI: 10.1016/j.jmva.2009.08.010.

[Qi+18] Di Qi, Joshua Arfin, Mengxue Zhang, Tushar Mathew, Robert Pless, and Brendan Juba, « Anomaly Explanation Using Metadata », *in*: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2018, pp. 1916–1924, DOI: 10.1109/WACV.2018.00212.

[Raj+19] Sreeraj Rajendran, Wannes Meert, Vincent Lenders, and Sofie Pollin, « Unsupervised wireless spectrum anomaly detection with interpretable features », *in*: *IEEE Transactions on Cognitive Communications and Networking* 5.*3* (2019), pp. 637–647, DOI: 10.1109/TCCN.2019.2911524.

[Ray16] Shebuti Rayana, *ODDS Library*, 2016, URL: http://odds.cs.stonybrook.edu.

[RB99] T.A. Runkler and J.C. Bezdek, « Alternating cluster estimation: a new tool for clustering and function approximation », *in*: *IEEE Transactions on Fuzzy Systems* 7.*4* (1999), pp. 377–393, DOI: 10.1109/91.784198.

[RL09] Konrad Rieck and Pavel Laskov, « Visualization and explanation of payload-based anomaly detection », *in*: *2009 European Conference on Computer Network Defense*, IEEE, 2009, pp. 29–36, DOI: 10.1109/EC2ND.2009.12.

[RRS00] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim, « Efficient Algorithms for Mining Outliers from Large Data Sets », *in*: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, Association for Computing Machinery, 2000, pp. 427–438, ISBN: 1581132174, DOI: 10.1145/342009.335437.

[RSG16] Marco Tulio Ribeiro, Sameer Singh, and Carlos 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*, Association for Computing Machinery, 2016, pp. 1135–1144, ISBN: 9781450342322, DOI: 10.1145/2939672.2939778.

[Rud19] Cynthia Rudin, « Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead », *in*: *Nature machine intelligence* 1.*5* (2019), pp. 206–215, DOI: 10.1038/s42256-019-0048-x.

[Ruf+21] Lukas Ruff, Jacob R. Kauffmann, Robert A. Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G. Dietterich, and Klaus-Robert Müller, « A unifying review of deep and shallow anomaly detection »,

in: *Proceedings of the IEEE* 109.*5* (2021), pp. 756–795, DOI: 10.1109/JPROC.2021.3052449.

[Sch+19] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Georg Langs, and Ursula Schmidt-Erfurth, « f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks », *in*: *Medical Image Analysis* 54 (2019), pp. 30–44, ISSN: 1361–8415, DOI: 10.1016/j.media.2019.01.010.

[Sch+99] Bernhard Schölkopf, Robert C Williamson, Alex Smola, John Shawe-Taylor, and John Platt, « Support Vector Method for Novelty Detection », *in*: *Proceedings of the 12th International Conference on Neural Information Processing Systems*, vol. 12, MIT Press, 1999, pp. 582–588.

[Sch96] E. Schikuta, « Grid-clustering: an efficient hierarchical clustering method for very large data sets », *in*: *Proceedings of 13th International Conference on Pattern Recognition*, vol. 2, 1996, 101–105 vol.2, DOI: 10.1109/ICPR.1996.546732.

[Seh+89] Allan Seheult, P. Green, Peter Rousseeuw, and Annick Leroy, « Robust Regression and Outlier Detection. », *in*: *Journal of the Royal Statistical Society. Series A (Statistics in Society)* 152 (Jan. 1989), p. 133, DOI: 10.2307/2982847.

[Shu+20] Amit K Shukla, Grégory Smits, Olivier Pivert, and Marie-Jeanne Lesot, « Explaining Data Regularities and Anomalies », *in*: *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, IEEE, 2020, pp. 1–8, DOI: 10.1109/FUZZ48607.2020.9177689.

[Sid+19] Md Amran Siddiqui, Alan Fern, Thomas G. Dietterich, and Weng-Keen Wong, « Sequential Feature Explanations for Anomaly Detection », *in*: *ACM Trans. Knowl. Discov. Data* 13.*1* (Jan. 2019), ISSN: 1556-4681, DOI: 10.1145/3230666.

[Son+07] Xiuyao Song, Mingxi Wu, Christopher Jermaine, and Sanjay Ranka, « Conditional anomaly detection », *in*: *IEEE Transactions on knowledge and Data Engineering* 19.*5* (2007), pp. 631–645, DOI: 10.1109/TKDE.2007.1009.

[Son+18] Fei Song, Yanlei Diao, Jesse Read, Arnaud Stiegler, and Albert 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, DOI: 10.1109/ICDMW.2018.00204.

[SRC19] Alison Smith-Renner, Rob Rua, and Mike Colony, « Towards an Explainable Threat Detection Tool », *in*: *IUI Workshops*, 2019.

[Tak19] Naoya 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, DOI: 10.1109/ICDMW.2019.00117.

[Tan+02] Jian Tang, Zhixiang Chen, Ada Wai-chee Fu, and David W. Cheung, « Enhancing Effectiveness of Outlier Detections for Low Density Patterns », *in*: *Advances in Knowledge Discovery and Data Mining*, Springer Berlin Heidelberg, 2002, pp. 535–548, ISBN: 978-3-540-47887-4.

[TK20] Naoya Takeishi and Yoshinobu Kawahara, « On anomaly interpretation via shapley values », *in*: *arXiv preprint arXiv:2004.04464* (2020).

[TSP21] Véronne Yepmo Tchaghe, Grégory Smits, and Olivier Pivert, « A Classification of Anomaly Explanation Methods », *in*: *Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, Springer International Publishing, 2021, pp. 26–33, ISBN: 978-3-030-93736-2, DOI: 10.1007/978-3-030-93736-2_3.

[VL20] Giulia Vilone and Luca Longo, « Explainable artificial intelligence: a systematic review », *in*: *arXiv preprint arXiv:2006.00093* (2020).

[Wan+18] Shaoni Wang, Haiyang Xia, Gang Li, and Jianlong Tan, « Group Outlying Aspects Mining », *in*: *Knowledge Science, Engineering and Management*, Springer International Publishing, 2018, pp. 200–212, ISBN: 978-3-319-99365-2, DOI: 10.1007/978-3-319-99365-2_18.

[Xu+21] Hongzuo Xu, Yijie Wang, Songlei Jian, Zhenyu Huang, Yongjun Wang, Ning Liu, and Fei Li, « Beyond outlier detection: Outlier interpretation by attention-guided triplet deviation network », *in*: *Proceedings of the Web Conference 2021*, Association for Computing Machinery, 2021, pp. 1328–1339, ISBN: 9781450383127, DOI: 10.1145/3442381.3449868.

[Xu+23] Hongzuo Xu, Guansong Pang, Yijie Wang, and Yongjun Wang, « Deep isolation forest for anomaly detection », *in*: *IEEE Transactions on Knowledge and Data Engineering* (2023), pp. 1–14, DOI: 10.1109/TKDE.2023.3270293.

[Yep+23] Véronne Yepmo, Grégory Smits, Marie-Jeanne Lesot, and Olivier Pivert, « Vers un partitionnement des données à partir d'une forêt d'isolation », *in*: *Conférence Extraction et Gestion de Connaissances 2023*, 2023, pp. 163–174.

[Yep+24] Véronne Yepmo, Grégory Smits, Marie-Jeanne Lesot, and Olivier Pivert, « CADI: Contextual Anomaly Detection using an Isolation Forest », *in*: *The 39th ACM/SIGAPP Symposium On Applied Computing*, 2024.

[YSP22] Véronne Yepmo, Grégory Smits, and Olivier Pivert, « Anomaly explanation: A review », *in*: *Data & Knowledge Engineering* 137 (2022), ISSN: 0169-023X, DOI: 10.1016/j.datak.2021.101946.

[YSZ02] Dantong Yu, Gholamhosein Sheikholeslami, and Aidong Zhang, « Findout: Finding outliers in very large datasets », *in*: *Knowledge and Information Systems 4* (2002), pp. 387–412, DOI: 10.1007/s101150200013.

[Zda09] John S. Zdanowicz, « Trade-based money laundering and terrorist financing », *in*: *Review of law & economics* 5.*2* (2009), pp. 855–878.

[Zha+19] Xiao Zhang, Manish Marwah, I-ta Lee, Martin Arlitt, and Dan 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, DOI: 10.1109/BigData47090.2019.9005989.

[ZNL19] Yue Zhao, Zain Nasrullah, and Zheng Li, « PyOD: A Python Toolbox for Scalable Outlier Detection », *in*: *Journal of Machine Learning Research* 20.*96* (2019), pp. 1–7, URL: http://jmlr.org/papers/v20/19-011.html.

COLLEGE DOCTORAL BRETAGNE
MATHS, TELECOMS
INFORMATIQUE, SIGNAL
SYSTEMES, ELECTRONIQUE

Université de Rennes

Titre : Contribution à la détection et à l'explication d'anomalies : une méthode unifiée basée sur les forêts d'isolation

Mot clés : détection d'anomalies, explication d'anomalies, clustering, forêt d'isolation, anomalie locale/contextuelle

Résumé : Cette thèse de doctorat se concentre sur l'explication des anomalies, problématique qui a été beaucoup moins explorée dans la littérature que l'explication des sorties des réseaux de neurones et des classifieurs. Sa première contribution est une taxonomie des méthodes d'explication des anomalies basée sur la nature de l'information qu'elle véhicule. La deuxième contribution est une méthode spécifique d'explication des anomalies, appelée CADI et reposant sur une version revisitée de l'algorithme des forêts d'isolation. Alors qu'une forêt d'isolation classique n'identifie que les anomalies, CADI reconstruit également une partition des instances régulières, puis positionne et explique chaque anomalie par rapport à ces groupes de régularités, tout ceci sans s'appuyer sur des algorithmes externes et sans trop rajouter de complexité à la méthode originale. Elle s'attaque par conséquent à trois problèmes avec une méthode unifiée : la détection des anomalies, le partitionnement et l'explication des anomalies. Des expériences menées sur des jeux de données réels et synthétiques démontrent l'efficacité et la robustesse de l'approche par rapport aux approches dédiées à l'une de ces trois tâches.

Title: Contribution to Anomaly Detection and Explanation: A Unified Method based on Isolation Forest

Keywords: anomaly detection, anomaly explanation, clustering, isolation forest, local/contextual anomaly

Abstract: This Ph.D. thesis focuses on anomaly explanation, which has been much less explored in the literature than the explanation of neural networks and classifiers outputs. Its first contribution is a taxonomy of anomaly explanation methods based on the information conveyed by the explanation. Its second contribution is a model-specific anomaly explanation method, called CADI, and relying on a revisited version of the Isolation Forest algorithm. Whereas a classic Isolation Forest only identifies the anomalies, CADI also clusters the regular instances, then positions and explains each anomaly in relation these groups of regular instances without relying on external algorithms and without increasing the complexity of the original method too much. It therefore tackles three problems with a unified method: anomaly detection, clustering and anomaly explanation. Experiments conducted on real-world and synthetic data sets demonstrate the effectiveness and the robustness of the approach when compared to state-of-the-art approaches realizing each of the three tasks separately.