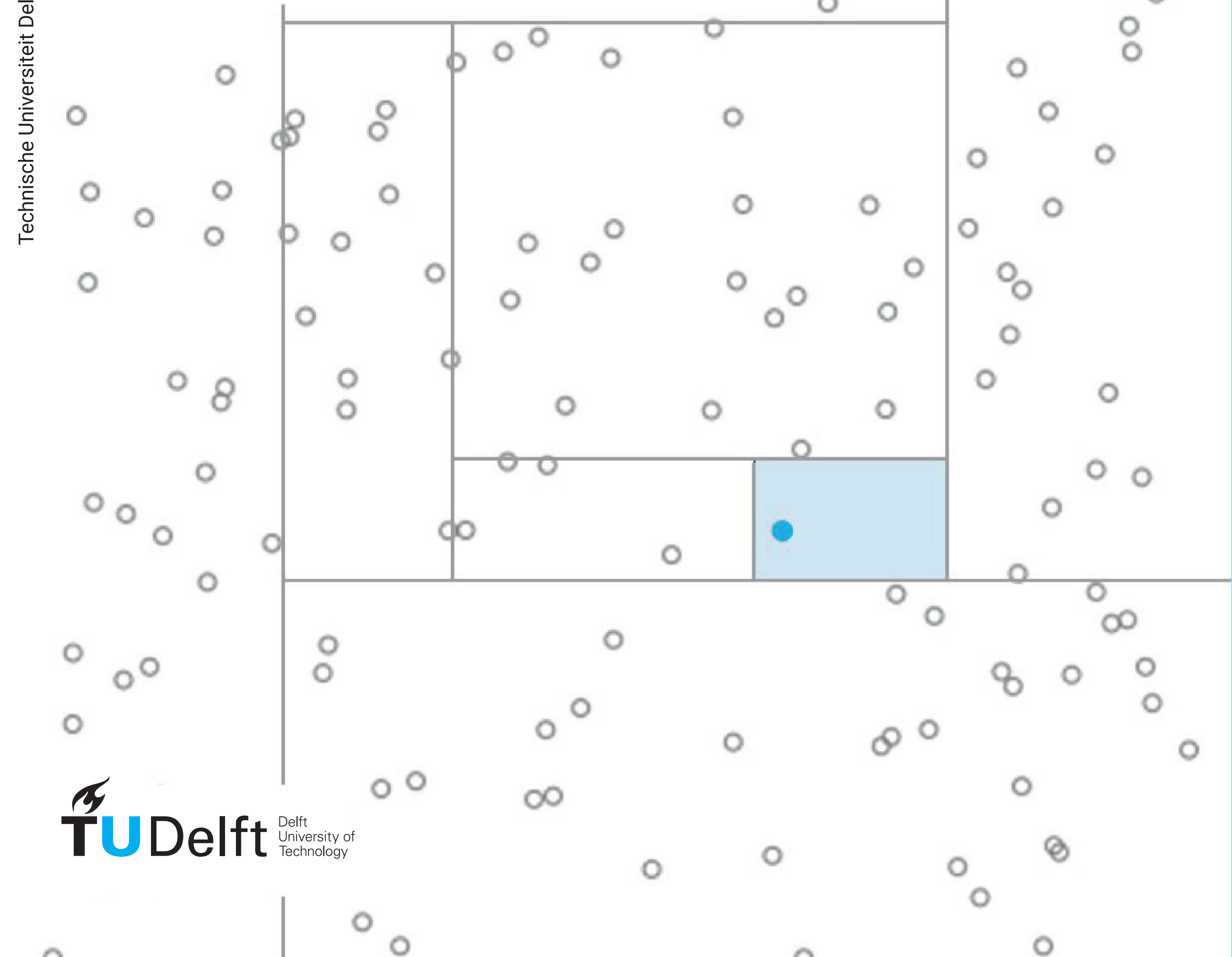
Locally Explainable Isolation Forest with Mixed-Attribute Data and Ternary Isolation Trees

Combatting Money Laudering with Anomaly Detection

M.E. Huistra



Technische Universiteit Delft

TU Delft Delft University of Technology

Locally Explainable Isolation Forest with Mixed-Attribute Data and Ternary Isolation Trees

Combatting Money Laundering with Anomaly Detection

by

M. E. Huistra

in partial fulfillment to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday October 1st, 2021 at 14:00 PM.

Student number:	4362829	
Project duration:	December 14th, 2020 – September 24th, 2021	
Thesis committee:	Prof. dr. ir. C. W. Oosterlee,	Utrecht University, supervisor
	Dr. ir. N. Parolya,	TU Delft
	Dr. ir. N. V. Budko,	TU Delft
	Dr. ir. E. Haasdijk	Deloitte
	L. A. Souto Arias MSc	CWI

Abstract

In the fight against money laundering, demand for data-driven Anti-Money Laundering (AML) solutions is growing. Particularly anomaly detection algorithms have proven effective in the detection of suspicious customer behaviour, as well as observing patterns otherwise hidden in customer transaction data. In this thesis, the Isolation Forest anomaly detection algorithm is studied in combination with the model-specific local explanation method, Multiple Indicator Local Depth-based Isolation Forest Feature Importance (MI-Local-DIFFI). To expand Isolation Forest to mixed-attribute data sets, the incorporation of nominal features is explored in more detail. This analysis resulted in the introduction of Isolation Forest with Categorical Sampling (iForest$_{CS}$), a methodology that directly incorporates nominal attributes into an isolation tree without the need of encoding it onto a numerical scale. This method is tested against different encoding strategies and Isolation Forest Conditional Anomaly Detection (iForest$_{CAD}$) using different synthetic data sets. The method shows improved performance to the utilization of encoding strategies for different parameters of the underlying synthetic data. Furthermore, this thesis explores the potential of ternary Isolation Forest, in which the branching strategy of an isolation tree is expanded to produce three child nodes. It is demonstrated using synthetic data, that particularly the performance of MI-Local-DIFFI reduces when applied to a ternary Isolation Forest. Finally, the research considers a practical use-case. Using customer transaction data from Triodos Bank, the locally explainable Isolation Forest is applied to mixed-attribute customer transaction data. This has provided useful insight and resulted in the detection of suspicious customer behaviour and the introduction of new rules into business practices. Although the most interesting customer behaviour did not directly emanate from the nominal attributes, the method of incorporating nominal features resulted in differences when considering the anomalies with the highest anomaly scores.

Preface

Nine months ago, the world seemed different. Trees were green, grew in forests, and were not constructed of lines of code. A category was something a contestant on “1 Tegen 100" could choose. Money laundering only occurred in movies.

The last nine months have been an interesting and educational roller-coaster. Although cropped up in my room for the majority of this project, the dynamic cooperation between the TU Delft, Deloitte and Triodos Bank almost got me thinking I was indeed at an office. It didn't take long before my room-mates coined the term "meeting Mark", which at times felt extremely accurate. Through this fruitful collaboration I have had the opportunity to learn from domain experts in the field of money laundering, financial fraud, and anomaly detection. For this, I am extremely grateful.

I would like to take this opportunity to express my sincerest gratitude to the people that have helped me along my research journey. My academic supervisor, Prof. Kees Oosterlee, my Deloitte supervisor, Dr. Evert Haasdijk, my Triodos Bank supervisor, Johan van Balken, and my daily supervisor, Luis Souto Arias, who have all helped me tremendously with their guidance, feedback, and our countless meetings. I also want to thank Dr. Nestor Parolya and Dr. Neil Budko for taking the time to be a part of my Thesis Committee. Additionally, a great thank you to all colleagues at Deloitte and Triodos Bank. I truly appreciated the time and energy invested to ensure the best possible (home-) working environment and involve me in the team. It was a joy to work together with you!

I also want to thank my friends and family. I will single out my parents, who's infinite love and support never ceases to amaze me. Thank you for everything! Finally, although I promised I wouldn't, I want to thank my girlfriend for always being there for me and bringing happiness wherever she goes.

Mark Huistra
Rotterdam, September 2021

Contents

List of Figures

List of Tables

Nomenclature

List of Symbols

$c(n)$	Average path length of a binary isolation tree ($c_t(n)$ for ternary)
d	Number of dimensions in a data set
$h(x)$	Path length of an observation x
k	Cardinality of a categorical feature
l	Height limit of Isolation Forest
n	Number of observations in a data set
o	An anomaly
p	Split value
Q	Feature in a data set
t	Isolation tree
T	Number of isolation trees
v_{ij}	A node in a tree
x	Observation in a data set
X	A $n \times d$ data set
γ	Euler-Mascheroni constant ($\gamma \simeq 0.57722$)
ψ	Sub-sampling size of Isolation Forest

List of Abbreviations

AI	Artificial Intelligence
AML	Anti Money Laundering
AUC	Area Under the ROC Curve
AUPRC	Area Under the Precision-Recall Curve
CDD	Customer Due Diligence
DIFFI	Depth-based Isolation Forest Feature Importance
DNB	De Nederlandsche Bank
FATF	Financial Action Task Force
FI	Feature Importance
FIU	Financial Intelligence Unit
i**Forest**$_{CAD}$	Isolation Forest Conditional Anomaly Detection
i**Forest**$_{CS}$	Isolation Forest with Categorical Sampling
HiCS	High Contrast Subspaces
MI-Local DIFFI	Multiple Indicator Local DIFFI
ROC	Receiver Operating Characteristic
WWFT	Money Laundering and Terrorist Financing (Prevention) Act

1 Introduction

By estimation of the Dutch Banking Association, 16 billion euro is annually laundered through Dutch financial institutes [1]. This money primarily finds its origins in drug trafficking and financial fraude, but emanates from human trafficking, child pornography or extortion as well. To protect the integrity, stability, and reputation of the Dutch financial systems, institutions collaborate with the government to combat money laundering. This collaboration aims at making the Netherlands an unattractive destination for money launderers.

Over the last few years, banks are leading the way in cementing their position as the gatekeeper of the integrity of the Dutch financial system. Heavy investments are being made to improve the bank's Customer Due Diligence (CDD) and transaction monitoring practices. With the increase in the quantity of financial and transaction data, the demand for data-driven Anti-Money Laundering (AML) solutions is growing. Whereas banks primarily use rule-based systems and domain-expert validation to detect suspicious activity, this thesis will further investigate the potential of utilizing data-driven anomaly detection to detect potential money laundering.

This thesis will focus on anomaly detection using Isolation Forest [2]. Isolation Forest has rapidly gained popularity after its implementation in Python's *Scikit-learn* and H_2O libraries and is applied in numerous domains. The methodology is particularly popular for its robust performance, low computational complexity, and general simplicity of the underlying algorithm. With little parameters to optimize, it is simple to implement. Furthermore, Isolation Forest lends itself to different local explanation methodologies, which is particularly useful in the context of anomaly detection in customer transaction data. Using a local adaptation to Depth-based Isolation Forest Feature Importance (DIFFI) [3], named Multiple Indicator Local DIFFI (MI-Local-DIFFI) [4], the question as to why a particular customer is an anomaly is addressed.

The first goal of this thesis is to investigate the performance of Isolation Forest applied to mixed-attribute data. Initially, Isolation Forest is constructed to incorporate continuously valued attributes. However, real data sets contain mixtures of attribute typologies. A nominal attribute is therefore unable to be considered in the algorithm without undergoing an encoding procedure that translates categories to a numerical scale. This thesis proposes a new method, Isolation Forest with Categorical Sampling ($i\text{Forest}_{CS}$), that directly incorporates categorical data without the need of encoding. This method is evaluated against commonly used encoding strategies utilized in the AML domain.

Furthermore, this thesis will evaluate the performance of a Ternary Isolation Forest. It is argued that the utilisation of ternary isolation trees can improve the detection of anomalies for particular data sets [4]. This thesis tries to substantiate this finding and extend the analysis to incorporate local explanation performance, as well as the incorporation of nominal attributes.

The thesis relies on different data sets to answer the research questions. A number of synthetically generated data sets are used to determine the performance of various Isolation Forest implementations. Through the use of synthetic data, experiments can be controlled and there is prior knowledge of true-outlying features and observations. Additionally, data from Triodos Bank is used to evaluate the potential of using data-driven anomaly detection as a tool to combat money laundering.

1.1. Research Objectives

Now that the thesis has been introduced, it is important to state the research objectives. The research objectives of this thesis are stated below:

RO1: *How should mixed-attribute data be incorporated into locally explained Isolation Forest?*

RO2: *Does a ternary tree structure improve an Isolation Forest's ability to detect anomalies and provide local explanations?*

Furthermore, the thesis provides more detailed insight into the Isolation Forest method in general. Different parameters that are introduced in the original Isolation Forest paper are evaluated, and this insight is valuable as well. From an additional practical perspective, data-driven anomaly detection to combat money laundering is further elaborated and experimented with.

1.2. Thesis Structure

In this section, the structure of the thesis is introduced:

In Chapter 2, an overview of the necessary background literature and related work is introduced. First, the chapter discusses money laundering and the current Anti-Money Laundering measures in place at financial institution. Then, a brief overview of existing anomaly detection methods are provided. This provides the background as to why this thesis focuses on Isolation Forest, which is explained in more technical details. After discussing the Isolation Forest algorithm, background into (local) explanation methods is provided. Finally, this chapter ends by providing insight into the incorporation of nominal features into anomaly detection algorithms that utilise continuously valued attributes only, such as Isolation Forest.

Chapter 3 discusses the methodology utilised throughout this thesis in more detail. First, a new adaptation to Isolation Forest is introduced that directly incorporates nominal features without requiring prior encoding. Second, an additional adaptation to Isolation Forest is introduced, namely a ternary Isolation Forest. This section will focus on the argumentation behind using ternary isolation trees, as well as derive theoretical expressions for a ternary tree's average path length. Next, the MI-Local-DIFFI method is explained in more detail, and changes to particular indicators are proposed. Finally, the chapter finishes with a description of evaluation metrics used when experimenting on data with available ground-truth data.

Experiments using synthetic data sets are discussed in Chapter 4. These experiments are conducted to gain a more thorough understanding of the Isolation Forest adaptation's performances. First, using independent data features, the performance of $i\text{Forest}_{CS}$ is compared to encoding strategies. Furthermore, the runtime of different implementations is addressed. Second, using conditionally dependent attributes, an analysis is performed into the use of nominal features for the detection of anomalies. Then, the ternary isolation forest is examined in more detail with respect to detection and explainability performance. Finally, the chapter revisits some of the parameters proposed in the original Isolation Forest paper [2] and provides insight into the performance sensitivity to these parameters.

In Chapter 5, the methodologies discussed in this thesis are applied to real customer transaction data. This chapter addresses the procedure of constructing relevant transaction features and the validation of anomalies without access to ground-truth information using domain expertise. Next, the submitted results to alert handlers are reviewed and the major findings are presented. Finally, a comparison is made between different Isolation Forest implementations and the results of the top anomalies detected to the bank's rule-based system.

The last chapter, Chapter 6, provides a conclusion of the results derived from this thesis. Furthermore, the chapter issues recommendations into promising further research directions.

2
Background & Related Work

In this section, an overview of the relevant background information and related work is provided. The chapter starts by providing a background into AML regulations and measures within banks in Section 2.1. With this background, the practical application of this thesis is immediately introduced.

Next, Section 2.2 will provide a background into different existing anomaly detection methods. This sets the tone for the Isolation Forest method, which is introduced in Section 2.3 and is the anomaly detection method of choice throughout this thesis. Since the Isolation Forest method forms the base of the thesis research, the method is discussed in detail.

A reason for using Isolation Forest as the anomaly detection method of choice throughout this thesis, is the possibility of applying local explanation methods to explain the algorithm's results. In Section 2.4, existing explanation methods are discussed, both model specific and model agnostic.

Finally, this chapter is concluded with an overview of the incorporation of mixed attribute data into Isolation Forest. Subsection 2.5.2 discusses common techniques of encoding nominal attributes, complications of categorical data in different anomaly detection methods, and existing approaches that incorporate nominal features directly into an Isolation Forest.

2.1. Anti-Money Laundering

The United Nations Office on Drugs and Crime estimates that the annual money laundered globally accounts for $2-5\%$ of the global GDP [5]. In order to combat these criminal transactions, governments establish AML regimes aimed at providing legal and regulatory tools necessary to combat the problem [6]. Financial institutions in the Netherlands are required to monitor and report unusual and suspicious behaviour, with the monitoring often done using a rule-based system with fixed historically derived thresholds. Although this interpretable system catches the most suspicious transactions, criminals can outwit fixed thresholds. This section will elaborate further on the Dutch AML regulations and practices.

Anti-Money Laundering and Counter Terrorism Financing Act

As a consequence of the international battle against money laundering, the Wet ter voorkoming van Witwassen en Financiering van Terrorisme (WWFT) (translated: Anti-Money Laundering and Counter Terrorism Financing Act) was implemented in the Netherlands in July 2008 [7]. The WWFT is an outcome of a merger between the laws addressing identification at service and reporting unusual transactions. The origins of the law can be traced to guidelines and recommendations for combatting money laundering set by the Financial Action Task Force on money laundering (FATF) [8]. It was deemed of utmost importance to protect the channels that can be exploited for money laundering from criminal abuse. By allowing cash flows from criminal misdemeanour to enter the financial system, perpetrators are able to enjoy their illegally obtained wealth and undermine the social fabric further.

The goal of the WWFT is to target and prevent money laundering and terrorist financing. Institutes that are covered by the WWFT are expected to uphold these goals in order to guarantee and protect the integrity, stability, and reputation of the Dutch financial institution. To achieve this goal, relevant institutions must uphold four core responsibilities [9]:

1. Perform thorough surveillance of clients through risk profiles.
2. Report unusual transactions to the Financial Intelligence Unit Netherlands (FIU).
3. Offer periodic training to personnel in recognizing unusual transactions and performing complete client surveillance and due diligence.
4. Adequately capture the results of risk profile ratings when requested by supervising entities.

2.1.1. Anti-Money Laundering Measures within Banks

Several Dutch banks have struggled with their role as gatekeeper of the stability, integrity and reputation of the Dutch financial system. Whereas numerous banks have received fines addressing their negligence, ING and ABN AMRO have recently been fined under the allegations of culpable money laundering. The conclusions from these criminal investigations by the Netherlands Public Prosecution Service into ING [9] and ABN AMRO [10] have shaken the Dutch financial sector and ignited the prioritisation of AML regulations.

By estimation of the Dutch Association of Banks [1], 16 billion euro is laundered in the Netherlands. When discussing the criminal exploitation of the financial systems, only estimations can be used. The laundered money finds its origin primarily in drug trafficking, but can also be derived from human trafficking, child pornography and extortion. This makes the Netherlands the worlds 8^{th} most popular money laundering destination, in which the banking institutions are deemed the number one money laundering risk. Banks must adhere to the responsibilities and goals of the WWFT. Thus, heavy investments are being made to rebuild and improve the functionality of the bank's customer due diligence and transaction monitoring.

Client behaviour and transactions are currently being monitored using rule-based systems. These rule-based systems monitor whether activity surpasses sets of thresholds. When this occurs, a client generates alerts, which are in turn checked by domain experts to determine the need of escalation to external organisations. The thresholds in this rule-based system are derived from historical activity and are often static with respect to the underlying data attributes. Using such rule-based systems allows for expert knowledge to drive decision making processes, while maintaining interpretability in the alert generation. However, there are some short-comings to such a rule-based decision system. First, a client that is constantly just below a given threshold will never be detected. Constant re-evaluation of the threshold boundaries is necessary with respect to the false positive and false negative (although difficult to confirm) alerts. Second, there exists no measure of suspicious activity between different attributes unless specified by a rule. Alerts and suspicious behaviour will go unnoticed unless specified in the rule-board, and suspicious customers from a completely data-driven standpoint might therefore be missed.

In this thesis, the investigation into the potential of anomaly detection techniques to improve the existing AML system of Triodos Bank is continued. It is of interest to explore the customer behaviour that is detected using data-induced anomaly detection and to what extent it is i) indeed suspicious with respect to money laundering, and ii) different to the findings of the rule-based customer monitoring. With the application of anomaly detection methods to combat money laundering, it is important to stress the ambiguity with respect to future regulations and practical applications. Therefore, the regulations proposed by the Dutch central bank are reviewed below, and will be considered specifically when applying anomaly detection to real-life customer transaction data.

2.1.2. Principles for Responsible AI in AML

The central bank of the Netherlands, De Nederlandsche Bank (DNB), acknowledges the increasing application and potential of Artificial Intelligence (AI) within the financial sector. To stimulate discussion concerning this topic, DNB presented its preliminary views on possible principles in a discussion paper in 2019 [11]. The paper provides a background in AI and its applications to the current and future Dutch financial sector. Furthermore, it presents general principles, derived from work of other regulatory bodies, that address responsible application of AI in the financial sector and instigate the dialogue of future regulations.

The principles presented by DNB function as a framework to assist firms in assessing the responsibility of their AI applications. These principles are abbreviated as SAFEST, and are divided into six key aspects, namely:

- **Soundness:** This is the primary concern of DNB. Applications in the financial sector should be both reliable and accurate. Furthermore, the behaviour of AI application is predictable and that they operate according to rules and regulations. A financial firm that applies any form of AI should be able to demonstrate that all measures are taken in order to ensure the business processes can continue effectively.
- **Accountability:** AI applications are complex and may cause deviating functionality and results that may damage a firms business practices or relevant stakeholders. The DNB requires firms to demonstrate awareness and understanding of their responsibilities with respect to AI applications. Furthermore, a firm must show that there is operational accountability within the organization, such that third party reliance shall not be used to limit accountability.
- **Fairness:** In order for society to trust the financial sector, AI applications may not disadvantage certain groups of customers. A financial organization should be able to demonstrate the appropriateness of their AI applications with respect to their defined concept of fairness.
- **Ethics:** It is important that the outcomes of AI applications do not violate the ethical standards held by the financial organization. There should be a guarantee that stakeholders can trust that no harm or mistreatment results from an organization's use of AI applications. Policies should reflect this moral obligation and must include criteria that assist decision making based on these applications.
- **Skills:** As decision making will start to rely on AI applications, it is necessary to ensure that an acceptable level of expertise is maintained by various functions throughout the organization, namely (senior) management, risk management, and compliance. It must be understood what the strengths and weaknesses of AI systems are, and how to prevent improper usage resulting in accidents.
- **Transparency:** When making use of AI applications, a financial organization should be able to explain its role within its business processes and appropriately describe how they work. This allows for proper risk management and auditing, but also allows for the application's supervision required to ensure stable and expected operations.

2.2. Anomaly Detection Methods

An anomaly is a data point that deviates from the remaining data. In many applications, it is attempted to determine an underlying model that governs the mechanisms and behaviour of a set of data. Any data point that behaves unusually with respect to this generated model can provide information about the atypical characteristics of the data. Recognizing these abnormalities is useful in multiple applications throughout a broad spectrum of varying domains. Examples of applications where anomaly detection is readily used include the financial, medical, quality control, web log, intrusion detection, and social media applications [12].

The majority of anomaly detection algorithms create some model that tries to capture the behaviour and mechanisms of the normal data. The choice of data model is therefore crucial to the overall results, yet the best choice of model often tends to be data specific. Using an incorrect or over-fitted/oversimplified model will likely provide poor results. To determine an anomaly, an anomaly detection algorithm outputs either an anomaly score or a binary label indicator [12]. This binary indicator emphasizes whether a data point is classified as an anomaly, which is frequently done through analysing the statistical distribution of the anomaly scores and setting a threshold.

This section will emphasize different types of anomaly detection models, although the remaining parts of this thesis will focus on the Isolation Forest [2, 13] method. The evaluation as to why this method is chosen, as well as a detailed description of the method, can be found in Section 2.3.

2.2.1. Existing Methods

An overview is presented of some of the existing anomaly detection methods. This section will not elaborate on the Isolation Forest method, as an entire section is dedicated to this methodology at a later stage.

Probabilistic and Statistical Models

When using probabilistic and statistical models for anomaly detection, model parameters θ are inferred to estimate the probability density function of the underlying data set X. Anomalies are identified as the observations with the smallest likelihood $\mathbb{P}(X|\theta)$ [14]. A common example of such methods is through probabilistic mixture modeling for which the parameters are estimated and optimized using the Expectation-Maximization (EM) algorithm.

An advantage to most probabilistic and statistical models is that they can readily be applied to mixed attribute data, but only when every mixture component has a generative model available [12]. For mixed-data, the product of attribute-specific likelihoods may be used to determine an anomaly measure. Considering the models use probabilities, data normalization is not necessary under the generative assumptions in order to determine the overall likelihood.

The major drawback to the typical probabilistic and statistical model is that the model attempts to fit the data to a certain distribution. However, there is no certainty regarding the correct underlying distribution of the data beforehand. Trying to fit the data to an incorrect distribution thus results in insufficient inference and improper conclusions regarding the data's probability density functions. Furthermore, when the total number of parameters increases in the probabilistic model, the risk of over-fitting increases as well [12]. In this case, the anomalies may affect the inference of the model parameters and inevitably the overall estimation of the data distribution.

Linear Models

The primary assumption for linear models applicable to anomaly detection is that data can be embedded onto a lower-dimensional space [12]. Any observation in the data that does not fit the embedded structure, can be classified as an anomaly. A method that finds its origin as early as 1901, and is still widely used for dimensionality reduction, is Principle Component Analysis (PCA). One-Class Support Vector Machines (SVM) [15] also form a competitive anomaly detection method [14]. Finally, neural networks and particularly Autoencoders can be utilized for anomaly detection.

In general, there are some limitations to linear modelling. First, the interpretability of the aforementioned models is relatively low. When embedding the data into a lower dimensional space, the physical significance of a particular dimension gets lost. This is because the sub-space dimensions are constructed from linear combinations of the original features. Thus, explaining why a specific observation is classified as an anomaly in this situation becomes impractical. Second, with large dimensionality in the data set, the computational complexity may become expensive. With a dimensionality of d, the covariance matrix becomes of size $d \times d$ [12], in the case of PCA, for example. With neural networks, the training stage remains computationally complex, even after recent algorithmic advancements.

Proximity-based Models

Within proximity-based models, an anomaly is defined as an observation for which the proximity is sparsely populated [12]. The advantage of proximity-based methods lies within the intuitive interpretation of the anomaly results. The notion that anomalies lie within low-density regions, have larger separation distances compared to their neighbours, or do not belong to a data cluster, is comprehensible and can be easily visualized. There are three subtly varying approaches to defining the proximity of a data point:

- **Cluster-based:** Assigning the data to a predetermined number of clusters allows for the quantification of an anomaly score. This can be done through consideration of cluster membership, distance from other clusters, and the size of the closest clusters [12]. Furthermore, clustering imposes a clear and intuitive relationship between data points; if a data point is not a member of a cluster, it can be perceived as an anomaly. Examples of clustering algorithms include: K-Means [16], *Density-based spatial clustering of applications with noise* (DBSCAN) [17] and Hierarchical clustering [18].

- **Distance-based:** Proximity is determined through evaluating distances between data points. Algorithms tend to define anomaly scores based on nearest neighbour distances. The most significant difference between clustering and distance-based algorithms is the detail of granularity in the analysis. When anomaly scores are desired for every data point, complexity of distance-based algorithms is proportional to $\mathcal{O}\left(N^2\right)$. Popular examples of distance-based methods are: *k-Nearest Neighbours* (kNN) and ORCA [19].
- **Density-based:** Density-based methods consider specific regions and use the number of points in these regions to determine the local density. Anomalies are then determined as data points situated in regions with lower local densities. Although extremely similar to clustering and distance-based evaluation, density-methods partition the data space whereas clustering partitions the data points [12]. The most popular density-based method in the field of anomaly detection is *Local Anomaly Factor* (LOF) [20]. Other methods include *Local Correlation Integral* (LOCI) [21], *Histogram-Based Anomaly Score* (HBOS) [22], and *Robust Kernel Density Estimation* (RKDE) [23].

Most proximity-based methods define anomalies with some perceived notion of distance at varying levels of granularity. In order to find the anomalies and distinguish these from normal data or noise, balancing global and local analysis is necessary. With a purely global analysis, certain anomalies may be missed due to the sensitivity to varying densities of data clusters. Yet, with a complete local analysis, small clustered anomalies may remain undetected as the local proximity becomes too great. Furthermore, in a highly dimensional data set, the quality of the anomalies deteriorates. As the dimensions increase, the contrast of distances becomes less evident and the stability of proximity-based methods deteriorates under certain distance measures [24] [25].

2.3. Isolation Forest

Isolation Forest [2, 13] is a model-based approach to anomaly detection that computes an anomaly score using a construction of so-called isolation trees. Rather than profiling normal points, Isolation Forest uses the concept of isolation to specifically isolate anomalies. The original paper shows through empirical evaluation that Isolation Forest performs favourably over numerous anomaly detection algorithms in terms of Area Under the Receiver Operating Characteristic (ROC) Curve (AUC) and processing time [2]. Furthermore, a survey of state-of-the-art anomaly detection methods was conducted [14]. Acknowledging average precision, algorithm robustness, memory usage, and computation time of fourteen different anomaly detection algorithms, using both synthetic and real-world data, Isolation Forest performed excellently. Specifically when dealing with large quantities of data, Isolation Forest should be the method of choice due to its scalability, limited memory requirements and efficient computation complexity.

This thesis continues with Isolation Forest due to the model specific explanation methods available and the interest in expanding research into the MI-Local DIFFI method [4], which will be explained further in Section 3.3. Furthermore, there is a potential performance improvement when using ternary isolation trees [4], which will be further explored in this thesis. More details on ternary trees will be provided in Section 3.2. This section will dive into the original Isolation Forest methodology.

2.3.1. Isolation and Isolation Trees

Liu et al. proposed the Isolation Forest algorithm [2, 13] based on the concept of isolation, defining isolation as: “separating an instance from the rest of instances" [2]. Typically, an anomaly is characterised as being sparse and different, becoming receptive to isolation when compared to normal instances. If a data-induced random tree were to be considered, Liu et al. observed that shorter paths were produced for anomalies through a random partitioning for two reasons. First, due to an unbalanced anomaly class in the data, the number of partitions needed to isolate a given anomaly is considerably shorter. This results in a shorter path in the random tree structure. Second, separation of data points with distinguishable feature-values is a likely occurrence in early partitioning. Thus, if a forest of random trees generates specific instances with considerably shorter path lengths, these points are more likely to be anomalies.

Definition 2.3.1. *An isolation tree t is a proper binary tree, meaning every internal node has exactly two children. Let v be a node of an isolation tree that is either an external-node, or an internal-node with one test. A test in node v consists of a feature value Q_i and a split value p such that the test $Q_i < p$ divides the data points into the left or right child nodes.*

Let $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ be an $n \times d$ data set with n number of instances and d dimensions. To construct an isolation tree, a sample of ψ instances $X' \subset X$ is used. X' is then recursively divided by conducting the test $Q_i < p$, where $Q_i \in Q_1, \dots, Q_d$ represents the feature selected with equal probability from the set of features and p is uniformly chosen from the range of Q_i. This is done until either the tree reaches a predefined height limit, only one instance remains in a node $\left(|X'| = 1\right)$, or all data points in a node have the same values. By assuming distinctness of all instances, a fully grown isolation tree will have ψ number of external nodes and $\psi - 1$ number of internal nodes. Thus, the total number of nodes is bounded by $2\psi - 1$. This results in a bounded memory requirements that grows linearly with ψ. A visualisation of the isolation tree's workings with respect to a normal and abnormal observation are shown in Figure 2.2.

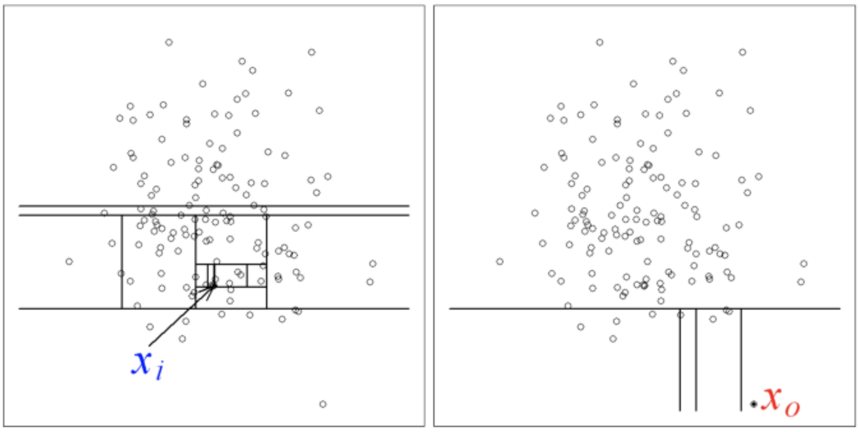


Figure 2.1: Visualisation of the Isolation Forest algorithm isolating an normal observation (left figure), and an abnormal observation (right figure). The figure is adopted from [2].

Definition 2.3.2. *The path length $h(x)$ of an observation x is measured by the number of edges that x traverses from the root node to its respective leaf node.*

The path length is used to indicate how susceptible an observation x is of being isolated in an isolation tree. Here, a short path length demonstrates high susceptibility to isolation, whereas a high path length demonstrates low susceptibility to isolation. This is best illustrated in Figure 2.2, where the path of both an anomaly and inlier are visualized for an individual isolation tree (Figure 2.2(a)) [26] and the construction of an Isolation Forest is visualised. (Figure 2.2(b)).

By using path lengths, isolation trees can be used to rank observations according to their degree of anomalous behaviour and are used to construct an anomaly scores. Now, the Isolation Forest algorithm will be described in more detail. Emphasis is placed on the training stage of the algorithm in Subsection 2.3.2 and the formulation of the anomaly score in Subsection 2.3.3.

2.3.2. Training Stage

An isolation tree is constructed through the partitioning of a random sub-sample of the data $X' \subset X$. This partitioning is continued until either the height limit is reached, or all instances are isolated or of similar value. Algorithm 1 describes the ensembling of isolation trees to construct an Isolation Forest, while Algorithm 2 describes the steps required to construct a single isolation tree.

The computational complexity of the training stage of Isolation Forest is $\mathcal{O}\left(T\psi\log_2\psi\right)$, where T represents the number of trees, and ψ represents the sub-sampling size. It is important to address some specific parameters of the Isolation Forest algorithm, namely the sub-sampling size ψ and the height limit l.

Sub-sampling size

The sub-sampling size ψ controls the training size of the algorithm. A sample of the overall data is taken randomly and used to construct an isolation tree. Empirically, the value of ψ is found to result in accurate anomaly detection across different data sets when set to 256.

Isolation Forest constructs a model using multiple sub-samples of the data, which reduces the effects of swamping and masking. *Swamping* refers to the occurrence of normal instances being labelled as

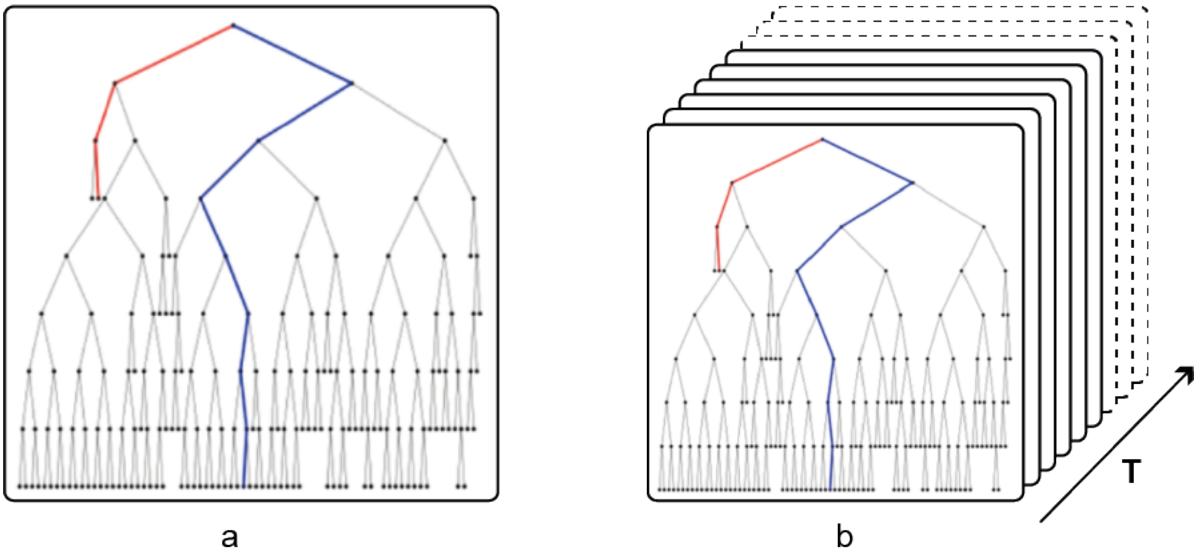


Figure 2.2: Visualisation of the behaviour of an anomaly (red) and of a normal observation (blue). Figure *a*) represents the behaviour in an individual isolation tree, where it is evident that the anomaly's path length is significantly shorter than that of the normal observation. The ensemble of different isolation trees leads to an Isolation Forest, as depicted in Figure *b*). When an observation consistently has short path lengths, it is likely to be an anomaly.

Algorithm 1 $IsolationForest(X, T, \psi)$

```
**Inputs:** $X$ - input data, $T$ - number of trees, $\psi$ - sub-sampling size
**Output:** A set of $T$ Isolation Trees
**Initialize** $Forest$
Set height limit $l = ceiling(log_2 \psi)$
**for** $i = 1$ to $T$ **do**
    $X' = sample(X, \psi)$
    $Forest = Forest \cup IsolationTree(X', 0, l)$
**end for**
**return** *Forest*
```

anomalies, of which the occurrence increases as the data size and the number of normal instances increases. *Masking* refers to the inability to detect anomalies due to the existence of too many anomalies in the data. This occurs when anomalies are clustered together, causing many anomaly detection methodologies to break down. By sub-sampling the data, Liu et al. argue that the effects of swamping and masking are significantly reduced. This is the result of reducing the data size for training isolation trees, causing normal instances to impact the isolation of anomalies to a lesser extent.

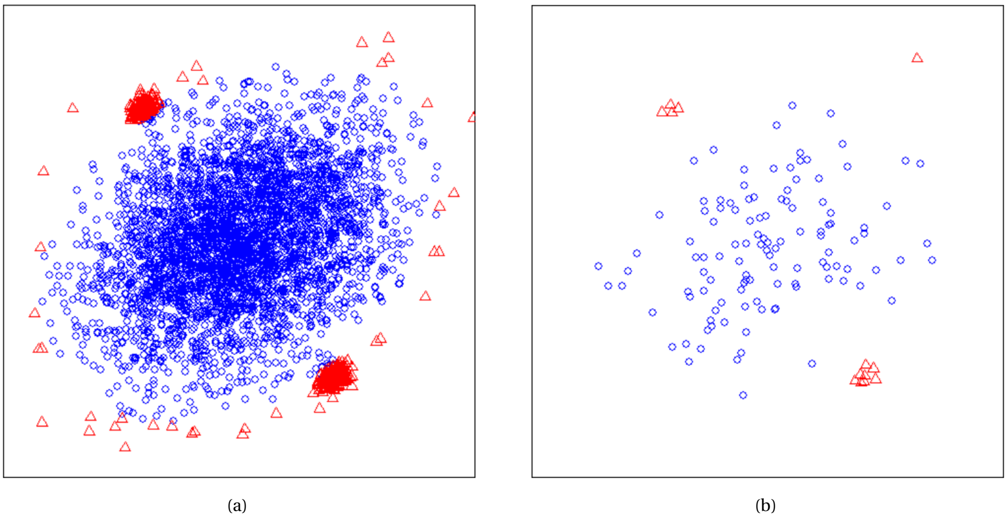
Making use of Figure 2.3, the authors of the paper set out to emphasize the effects of masking and swamping. Figure 2.3(a) shows 4096 instances artificially generated, with two distinct, dense anomaly clusters situated on the edges of a single, large cluster of normal instances. The normal instances situated close to the anomaly clusters cause a swamping effect. Furthermore, the dense anomaly clusters cause a masking effect. Through sub-sampling, displayed in Figure 2.3(b), the anomaly clusters become more distinct. Sub-sampling the data causes both the swamping and masking effects to diminish through clearance of the normal points close to the anomaly clusters and reducing the size of the anomaly clusters, respectively. As a result, anomalies and anomaly clusters become easier to isolate through the sub-sampling in an isolation tree.

However, subsampling can also hinder the process of identifying an anomaly [4]. Imagine an extremely imbalanced data set, where the anomalies comprise a significantly small percentage of the total data. With subsampling, the sub-population used for training the isolation trees will frequently contain purely normal instances. This will negatively impact the ability of the Isolation Forest method to isolate particular anomalies. Thus, it is argued that isolation trees be trained without subsampling when the overall data is

Algorithm 2 $IsolationTree(X', e, l)$

```
Inputs: $X'$ - input data, $e$ - current tree height, $l$ - height limit
Output: An Isolation Tree
if $e \geq l$ or $|X| \leq 1$ then
    return $exNode\{Size = |X'|\}$
else
    Let $Q$ be a list of attributes in $X'$
    Randomly select an attribute $Q_i \in Q$
    Randomly select a split point $p \in \left(min(Q_i), max(Q_i)\right)$
    $X_l = filter(X', Q_i < p)$
    $X_r = filter(X', Q_i \geq p)$
    return $inNode\{Left = IsolationTree(X_l, e+1, l),$
                   $Right = IsolationTree(X_r, e+1, l),$
                   $SplitAttribute = Q_i,$
                   $SplitValue = p\}$
```

significantly large and imbalanced. This will impact the runtime complexity notably, yet is considered viable when performance is valued over runtime, like in the case of money laundering detection. In Subsection 4.2.4, the effect of the sub-sampling size on the performance of Isolation Forest is evaluated.


(a)

(b)

Figure 2.3: Artificially generated data set to demonstrate the possible effects of swamping and masking, and how sub-sampling reduces these effects. Figure (a) shows the original data with 4096 instances, while Figure (b) shows a sub-sample of the data of 128 instances. Red triangles denote anomalies and blue circles denote normal instances [2].

Height limit

Using the argumentation that an anomaly is more susceptible to isolation and will therefore result in a shorter average path length, a height limit is introduced. With this height limit, complexity is reduced significantly as the trees are not grown to completion. This limit is a function of the sub-sampling size ψ, and is set as: $l = ceiling(\log_2 \psi)$. The ceiling is argued to be approximately equal to the average height of an isolation tree given ψ observations, $c(\psi)$, which is derived below. This is however not the case and will be discussed and evaluated further in Subsection 4.5.1.

The initial derivation of the expected path length of an isolation tree containing n observations exploits the similarity in the structure of isolation trees and binary search trees. Using the analysis into binary search trees, and explicitly the calculations into the average path length of an unsuccessful search, determines the average path of an isolation tree, $c(n)$, to be:

$$c(n) = 2H(n-1) - \frac{2(n-1)}{n}$$

where $H(\cdot)$ represents the harmonic number [2]. However, it is observed that this derivation does not consider that an isolation tree is a proper binary tree [4], thus having either 0 or 2 child nodes. When this is accounted for, the average path of an isolation tree becomes slightly different as stated in Theorem 2.3.1.

Theorem 2.3.1. *The average path length of a fully grown isolation tree t with n observations can be represented by:*

$$c(n) = \begin{cases} 0 & \text{if } n = 1, \\ 1 & \text{if } n = 2, \\ 2H(n) - 2 & \text{if } n > 2. \end{cases} \tag{2.1}$$

where $H(n)$ represents the n^{th} harmonic number.

Proof. First, in the case of $n = 1$, the isolation tree is already fully grown at its root node, and no additional paths are necessary for isolation. Therefore, $c(1) = 0$.

Let $E(n)$ denote the *external path length* of an isolation tree t with n observations, which represents the sum of the path lengths from the root node to every individual external node in the isolation tree. The average external path length will be used to determine the average path length $c(n)$, by dividing with the total number of external nodes.

Let $t_n(l)$ be an arbitrary isolation tree with a root node who's left child contains $l \in \{1, 2, ..., n-1\}$ external nodes and who's right child contains $n - l$ external nodes. The average external path length $E_t(n)$ can then be computed using the sum of average external path lengths of the root's left and right child nodes, $E_t(l)$ and $E_t(n - l)$ respectively, plus n. This last term accounts for the fact that $E_t(l)$ and $E_t(n - l)$ are a level deeper in the isolation tree with respect to the root node.

Considering there are assumed to be $n - 1$ unique allocations in the left and right child nodes, all with equal probabilities, the expectation is taken over all external path lengths of trees $l \in \{1, 2, ..., n-1\}$. For $n > 1$:

$$\begin{aligned} \mathbb{E}_t(n) &= \frac{1}{n-1} \sum_{l=1}^{n-1} \big(E_t(l) + E_t(n-l) + n\big) \\ &= \frac{1}{n-1} \sum_{l=1}^{n-1} \big(E_t(l) + E_t(n-l)\big) + n \\ &= \frac{2}{n-1} \sum_{l=1}^{n-1} E_t(l) + n. \end{aligned} \tag{2.2}$$

The result from Equation 2.2 can be used to show that $E_t(n) = 2$, and therefore that $c(2) = 1$. Continuing, for $n > 1$:

$$(n-1)\mathbb{E}_t(n) = 2 \sum_{l=1}^{n-1} E_t(l) + (n-1)n \tag{2.3}$$

and $n > 2$:

$$(n-2)\mathbb{E}_t(n-1) = 2 \sum_{l=1}^{n-2} E_t(l) + (n-1)(n-2). \tag{2.4}$$

Through subtracting Equation 2.4 from Equation 2.3 the two equations above, for $n > 2$:

$$\begin{aligned} (n-1)\mathbb{E}_t(n) - (n-2)\mathbb{E}_t(n-1) &= 2 \sum_{l=1}^{n-1} E_t(l) + (n-1)n - 2 \sum_{l=1}^{n-2} E_t(l) - (n-1)(n-2) \\ &= 2 \sum_{l=1}^{n-1} E_t(l) + (n-1)n - 2\Big(\sum_{l=1}^{n-1} E_t(l) - E_t(n-1)\Big) - (n-1)(n-2) \\ &= 2\mathbb{E}_t(n-1) + 2(n-1). \end{aligned}$$

This yields an expression for the average external path length:

$$\mathbb{E}_t(n) = \frac{n}{n-1}\mathbb{E}_t(n-1) + 2. \tag{2.5}$$

To derive the desired outcome, Equation 2.5 is simplified to a non-recursive form. This can be achieved by examining the following sequence:

$$\begin{aligned}
\frac{\mathbb{E}_t(n)}{n} &= \frac{1}{n-1}\mathbb{E}_t(n-1) + \frac{2}{n}, && \text{for } n > 2,\\
\frac{\mathbb{E}_t(n-1)}{n-1} &= \frac{1}{n-2}\mathbb{E}_t(n-2) + \frac{2}{n-1}, && \text{for } n > 3,\\
&\vdots\\
\frac{\mathbb{E}_t(n-k)}{n-k} &= \frac{1}{n-k-1}\mathbb{E}_t(n-k-1) + \frac{2}{n-k} && \text{for } n > k+2,\\
&\vdots\\
\frac{\mathbb{E}_t(3)}{3} &= \frac{1}{2}\mathbb{E}_t(2) + \frac{2}{3}.
\end{aligned}$$

Finally, this sequence of equations can be used to obtain:

$$\begin{aligned}
\frac{\mathbb{E}_t(n)}{n} &= \frac{1}{2}\mathbb{E}_t(2) + 2\sum_{i=3}^{n}\frac{1}{i}\\
&= 2H(n) - 2,
\end{aligned}$$

where $H(n)$ represents the n^{th} harmonic number. The corresponding asymptotic expansion of the harmonic number is: $H(n) \sim ln(n) + \gamma + \frac{1}{2n} - \frac{1}{12n^2} + \ldots$, where $\gamma \simeq 0.57722$ represents the Euler–Mascheroni constant. Therefore, the following is obtained:

$$\boxed{c(n) = \frac{\mathbb{E}_t(n)}{n} = \begin{cases} 0 & \text{if } n = 1,\\ 1 & \text{if } n = 2,\\ 2H(n) - 2 & \text{if } n > 2.\end{cases}}$$

which is the desired result. □

This final expression for the average path length of a fully grown isolation tree can be also expressed in terms of the asymptotic expansion of the harmonic number:

$$c(n) = \frac{\mathbb{E}_t(n)}{n} \approx \begin{cases} 0 & \text{if } n = 1,\\ 1 & \text{if } n = 2,\\ 2\big(ln(n) + \gamma - 1\big) & \text{if } n > 2.\end{cases} \tag{2.6}$$

2.3.3. Evaluation Stage

After having trained an isolation tree, an anomaly score s is derived. This is done using the average path length $\mathbb{E}\big[h(x)\big]$ of every observation x over all trees. By observing how x traverses along the isolation tree, the path length from the root node until x is terminated at an external node. When this is the case, there are two possibilities in calculating the path length $h(x)$:

1. Observation x is isolated in the external node. The path length is then determined through the count of the number of edges e from the root node to the external node.

2. The external node is terminated before x is fully isolated, meaning the external node has $Size > 1$. This can occur when the height limit is reached, or all observations are of similar value in the feature space. The path length is then determined as $e + c(Size)$, where the second term accounts for the average path length of the unbuilt sub-tree.

The complexity of this evaluation stage is $\mathcal{O}\left(nT\log_2\psi\right)$, resulting in the total complexity of Isolation Forest to equal $\mathcal{O}\left(T\log_2\psi(n+\psi)\right)$. Using the expected path length $\mathbb{E}\left[h(x)\right]$ of the observations, an isolation score s is defined as:

$$s(x,n) = 2^{\frac{-\mathbb{E}(h(x))}{c(n)}} \tag{2.7}$$

In Equation 2.7, $c(n)$ is used to normalize the expected path lengths of an observation. Notice that s maps all scores to the domain of $(0,1]$, and scores can be interpreted as follows:

- When $\mathbb{E}\left[h(x)\right] \to 0$, then $s \to 1$. Therefore, if an observation returns a score close to 1, there is evidence that this observation is an anomaly.
- When $\mathbb{E}\left[h(x)\right] \to n-1$, then $s \to 0$. Therefore, if an observation returns a score close to 0, there is evidence that this observation can be regarded as a normal observation.
- When $\mathbb{E}\left[h(x)\right] \to c(n)$, then $s \to 0.5$. Therefore, if all observations have a score of $s \approx 0.5$, the data has no distinct anomalies.

Even though the distribution of anomaly scores varies according to the underlying data sets, the anomaly scores can be used to determine which observations show anomalous behaviour. In this thesis, Isolation Forest will be extended to incorporate mixed-attribute data and ternary splitting strategies, which will be discussed in Chapter 3.

2.4. Explanation Methods

The anomaly detection methods discussed in Subsection 2.2.1 are all commonly used to detect anomalies by calculating an anomaly score or a binary label indicator. From these anomaly scores or binary label indicators, it is not immediately clear how the model's conclusions are achieved. For most anomaly detection models, it is difficult to obtain a description as to why a particular data point is classified as an anomaly.

In an AML use case, and particularly detecting suspicious customers, it is of utmost importance to be able to explain on both a global and local level what data attributes contribute to the overall results. Through a global feature evaluation, the attributes that are generally most important are identified. By this identification, one can validate the input data and gain information on the data subspace where anomalies are located. In the practical use-case of this thesis, global feature evaluation can be used to determine whether newly added data attributes, which deviate from the attributes accounted in the rule board, are deemed important in generally identifying anomalies. Furthermore, it allows analysis on the isolating capabilities of nominal attributes.

More important, however, is the local explanation of the detected anomalies. From a practical standpoint, any anomaly that is detected in the Triodos Bank customer monitoring data, will be communicated to domain experts to validate the money laundering suspicion. In this communication, providing additional information into the customer behaviour allows for a more detailed and controlled evaluation. Furthermore, in line with the regulations discussed by the DNB (SAFEST), the process of identifying suspicious customers must be transparent. Through local explanation, transparency is introduced where there was none initially.

2.4.1. Existing Explanation Methods

The anomaly detection method that will be used throughout this paper is an adaptation to Isolation Forest. Whereas an individual isolation tree is intrinsically comprehensible, an ensemble of them diminishes the overall explainability. A methodology is thus needed that provides insight into local behaviour after constructing the forest. Adaptions of an explanation method that was specifically designed for Isolation Forest, *DIFFI*, and one explanation method that is model agnostic, *TreeSHAP*, are briefly discussed.

(Local) DIFFI

In order to examine feature importance for Isolation Forest, the DIFFI [3] was introduced. This global explanation method has been adapted to local variants (discussed later) and does not require alterations to the original Isolation Forest methodology. The method defines features as “important" when the split test in these features allows for the isolation of anomalies while relegating inliers to leaf nodes deeper in the tree [27]. Furthermore, a feature splitting test is important when imbalanced for anomalies, yet balanced when only inliers remain in the node. The DIFFI method has a high computational complexity of $O(T \cdot n \cdot v)$, where T represents the total number of isolation trees, n the total number of observations, and v the total number of vertices.

The DIFFI method can be summarized as follows:

1. Using the results from the Isolation Forest, partition the data set $\mathscr{D}$ into the predicted inliers $\mathscr{D}_I$ and the predicted anomalies $\mathscr{D}_O$.

2. Determine for every node in every isolation tree the *induced imbalance coefficient*. This coefficient reflects the node's ability to isolate samples.

3. Register how often feature f appears in the path of every anomaly, and determine the cumulative importance of feature f, using information on the depth of the leaf nodes of specific samples. Do this for all features and also with respect to the inliers.

4. Determine the feature importance of a specific feature by normalizing the ratio of the cumulative importance of the feature for anomalies with respect to the cumulative importance of the feature for anomalies.

The authors of the DIFFI method extended their work on Isolation Forest specific explanation models, only this time focusing on obtaining a local explanation. This resulted in the Local-DIFFI method [27]. The method is derived from DIFFI, where the differences are mainly due to the inability to calculate certain values when focusing on the local explanation of a single observation. Namely, the induced balance coefficient can no longer be computed, as well as any normalizing quantities for specific to inliers.

MI-Local DIFFI

The Multiple Indicator Local-DIFFI (MI-Local-DIFFI) method [4] is an adaptation of the DIFFI method that provides local explanations. The method stems from the assumption that information contained in the nodes and the splitting structure of the isolation trees can be manipulated to determine a measure of feature importance.

In an isolation tree, an observation transcends down a particular path. The path an observation follows, is dictated by the features that are chosen in the tree and the observation's feature value with respect to the node's splitting criteria. In the MI-Local-DIFFI method, the path of an anomaly is observed, and indicator scores are assigned according to the following characteristics:

1. **Path length indicator:** Assigns a feature importance weight by considering the anomaly's path length in every isolation tree.

2. **Split proportion indicator:** Assigns a feature importance weight by observing the proportion of observations that are assigned to the same branch as the traced anomaly, for a particular feature split.

3. **Split interval length indicator:** Assigns a feature importance weight by considering the proportion of the sub-interval length that contains the anomaly to the overall feature range in a specific node.

The MI-Local DIFFI method will be explained further in Section 3.3, in which the indicators will be discussed in more detail and alterations to incorporate mixed-attribute data are discussed. MI-Local DIFFI showed excellent results for both runtime and performance compared to the TreeSHAP explanation method when conducting experiments with synthetic data [4].

TreeSHAP

The TreeSHAP method [28] is a model agnostic explanation method that makes use of Shapley values. Shapley values find its origins in coalitional game theory, and can be extended to the predictions of machine learning models. The TreeSHAP method is a continuation of the SHAP (SHapley Additive exPlanation) values proposed in [29]. For a more detailed derivation of the SHAP values, the reader is referred to the original paper.

In TreeSHAP, an algorithm for tree ensembles is derived that reduces the SHAP values' computational complexity from $\mathcal{O}(TL2^M)$ to $\mathcal{O}(TLD^2)$, where T is the number of trees, L represents the maximum number of leaves in a tree, M is the number of features, and D denotes the tree's maximum depth [28]. Thus, instead of the computational complexity being exponential with respect to the number of features, the TreeSHAP method has a computational complexity that is quadratic in a tree's maximum depth. Furthermore, the method is implemented efficiently in the $shap$ python library.

In this thesis, TreeSHAP is not used extensively in experiments or analysis. This is primarily due to the incompatibility of the proposed iForest$_{CS}$ method (introduced in Section 3.1) and the ternary Isolation Forest (introduced in Section 3.2 to the python libraries. In Section 6.3, a recommendation is made to address this issue.

2.5. Anomaly Detection and Categorical Data

Initially, the Isolation Forest algorithm is constructed to include continuously valued attributes only. In the original paper, all experiments were conducted only after nominal and binary valued attributes had been removed [2]. However, real data sets consistently contain a mixture of variable types; some variables assume quantitative values whereas other variables might be qualitative or categorical [30]. In a large variety of applications and domains mixed attribute data is evaluated. It is therefore valuable to consider different methodologies that address the incorporation of categorical data in an anomaly detection setting.

There are typically two types of categorical data, namely nominal and ordinal. Nominal data refers to data that is labeled without containing quantitative significance. These are the attributes this thesis is most interested in, as there is no underlying ordering to the categories. Examples include, but are certainly not limited to, color, nationality, or internet provider. Ordinal data, on the other hand, has a natural ordering to the variables, but the distance between orderings is not defined. Examples include the EU energy labels and eco-design of electrical appliances (A+++ through G), the Likert scheme, or education level. The natural ordering of ordinal data typically improves the identification of anomalies when compared to nominal data [31].

2.5.1. Encoding of Nominal Features

Many anomaly detection methods require the model's input and output variables to be numeric. Thus, nominal features are often encoded to a numerical or integer scale before evaluating an anomaly detection model. Here some of the most common nominal data encoding strategies that are used in practice and literature are discussed.

Before discussing common encoding approaches, it is useful to appropriately formalize definitions and notation. This is done using the relational database formulation from [32]. A *relation scheme* $\mathscr{R}$ is defined as a finite set of *feature names* $\{Q_1, Q_2, ..., Q_d\}$, otherwise known as feature names. For each attribute name Q_j, the set $\mathscr{D}_j$ refers to the *domain* of Q_j. A data table is a *relation* r on the scheme $\mathscr{R}$: it is a mapping $\{t_i : \mathscr{R} \to \bigcup_{j=1}^{d} \mathscr{D}_j, \quad i = 1, ..., n, \quad j = 1, ..., d\}$, where for every sample $\{t^i \in r, \quad t^i(Q_j) \in \mathscr{D}_j\}$. If dealing with a numerical attribute, then $\mathscr{D}_j \subseteq \mathbb{R}$. Otherwise, when an attribute is nominal and thus represented by strings, then $\mathscr{D}_j \subseteq \mathbb{S}$, where $\mathbb{S}$ represents the set of finite-length strings. Finally, the cardinality of a variable is denoted by $k_j = \text{card}\left(\mathscr{D}_j\right)$ [33].

When models require vector data, nominal attributes must undergo feature mappings that replace the categorical elements $t^i(Q_j)$ by feature vectors of dimension p_j:

$$\mathbf{x}_j^i \in \mathbb{R}^{p_j}, \quad p_j \geq 1.$$

Thus, by defining a categorical element before encoding $x_j^i = t^i\left(Q_j\right) \in \mathbb{R}^1$, the feature matrix $\mathbf{X}$ after encoding can be represented as :

$$\begin{bmatrix} \mathbf{x}_1^1 & \dots & \mathbf{x}_d^1 \\ \vdots & \ddots & \vdots \\ \mathbf{x}_1^n & \dots & \mathbf{x}_d^n \end{bmatrix} \in \mathbb{R}^{n \times p}, \quad p = \sum_{j=1}^{d} p_j.$$

The transformed feature matrix **X** therefore has p dimensions, rather than the original d feature dimensions, where the size p is dependent on the encoding strategy. With this formalization, common and practical encoding strategies will be discussed, using a single nominal feature Q for simplification, omitting column indices j.

Label Encoding

One of the most popular and simple methods is label encoding. In label encoding, the labels of a given feature are encoded with integer values. For example, imagine a feature that indicates colour, having unique labels such as Red, $Yellow$, and $Blue$. With label encoding, integer values are assigned to the unique instances in the feature. Thus, Red is assigned a value of 1, $Yellow$ as 2, and finally $Blue$ as 3. To formalize, let Q be a nominal feature with cardinality $k \geq 2$ such that the $\mathscr{D} = \{d_l, 1 < l \leq k\}$ and:

$$\mathbf{x}^i = \left[d^i = l\right] \in \mathbb{R}.$$

The i^{th} feature is therefore assigned the value l, where $l \in [1, k]$ and no increase in dimensionality results from label encoding. The obvious issue with label encoding is that a relation is introduced in labels where no relation or order is actually present. With ordinal features, this encoding is much more logical, and often referred to as **ordinal encoding**.

One-Hot Encoding

In one-hot encoding, a given nominal feature is transformed into a matrix of k columns, where k represents the cardinality of the feature. Every column represents a binary variable for each unique feature value. To formalize, let Q once again be a nominal feature with $k \geq 2$ such that the $\mathscr{D} = \{d_l, 1 < l \leq k\}$ and $t^i(Q) = d^i$. One-hot encoding then assigns an indicator vector over $\{d_l\}$, such that:

$$\mathbf{x}^i = \left[\mathbb{1}_{\{d_1\}}(d^i), \quad \mathbb{1}_{\{d_2\}}(d^i), \quad ..., \quad \mathbb{1}_{\{d_k\}}(d^i)\right] \in \mathbb{R}^k.$$

One-hot encoding is a popular encoding strategy with numerous variations proposed, and is intended to be used for mutually exclusive categories [34]. When applying one-hot encoding to training data, testing data that includes more than one new label will all be assigned a zero vector. This potentially creates overlap between encoded category labels, impacting performance. Furthermore, through one-hot encoding, nominal features with a high cardinality will result in a significant increase in the feature matrix' dimensionality. This increases the computational costs of performing anomaly detection on the transformed data. This particular problem can be handled by performing an additional dimensionality reduction pre-processing step, which comes at the cost of loss of information [33].

Frequency Encoding

With frequency encoding, the nominal feature is transformed to a vector who's entries correspond to the normalized frequency of the original feature value. To formalize, let Q once again be a nominal feature with $k \geq 2$ such that the $\mathscr{D} = \{d_l, 1 < l \leq k\}$ and $t^i(Q) = d^i$. The normalized frequency of a particular feature value d_l is represented as $\text{Freq}(d_l) = \frac{\sum_{i=1}^{n} \mathbb{1}_{\{d_l\}}(d^i)}{n}$. Then, the nominal feature can be represented as:

$$\mathbf{x}^i = \left[d^i = \text{Freq}(d_l)\right] \in \mathbb{R}.$$

Similarly to label encoding, frequency encoding does not increase the dimensionality of the original data set. However, one must be aware that the similarity of feature attributes is decided by the frequency of the training data. This can create inconsistencies in the frequency similarities when considering the testing data. Furthermore, frequency encoding cannot distinguish between categories with similar frequencies. When this is the case, categories in a feature will be assigned the same value, and will be treated identically in the encoded feature space.

2.5.2. Anomaly Detection for Nominal Data

Although the analysis of anomaly detection methods for quantitative and numerical data is plentiful, analysis of anomaly detection methods using nominal data is not as common. As mentioned earlier, however, nominal data is used and essential in numerous domains. Applications include, but are certainly not limited to: insurance fraud [35, 36], advertisement fraud [37], and geoinformatics [38].

In contrast to numerical data, determining a degree of similarity between values of an disordered nominal feature is difficult. This complicates the generalization of anomaly detection algorithms to the mixed-attribute or purely categorical domain and creates the following problems [12]:

- With statistical algorithms, the underlying mechanics to the data are determined. However, with nominal data, all statistical information can only be obtained through evaluation of the frequencies of the unique categories. Therefore, no notion of averages, for example, can be computed.
- Linear models are dependent on the continuous data attributes. Without this continuity, linear models cannot be directly applied without alterations to the general algorithms.
- Common measures of similarity or proximity do not hold for nominal data. Therefore, for proximity-based methods, all measures must be redefined to draw sensible conclusions from these methods.

2.5.3. Isolation Forest and Nominal Data

In accordance to the first research objects, the incorporation of mixed-attribute data into the Isolation Forest is investigated. The research is centralized around Isolation Forest due to its scalability, competitive performance, and the potential of obtaining local explanations of anomalies. As the original Isolation Forest paper mentions, Isolation Forest only incorporates numerical attributes into the algorithm. Although many papers recommend to some extent the incorporation of nominal attributes, limited research is conducted into the performance of specifically Isolation Forest when applied to mixed-attribute data. The following methodologies are known Isolation Forest adaptations that incorporate nominal features beyond the typical label encoding [39], one-hot encoding [40], frequency encoding [41], or omit nominal features altogether [2].

Isolation Forest Conditional Anomaly Detection

iForest$_{CAD}$ [36] is an extension to the classical Isolation Forest. Isolation Forest is designed as a global anomaly detector, using the concept of isolation applied on the entire data set to produce anomaly scores. By performing conditional anomaly detection (CAD) on well-defined data partitions, iForest$_{CAD}$ aims to identify “hidden" anomalies whilst incorporating both nominal and numerical attributes.

The algorithm uses the following steps as its high-level methodology, which are illustrated in Figure 2.4:

1. **Selection:** Driven by expert knowledge, nominal and numerical attributes are selected from the total data set.
2. **Partitioning:** Determine the possible combinations that the values of the nominal attributes can take on and divide the data according to the combinations. This is done by computing the Cartesian product of all nominal attributes selected during the Selection phase. Effectively, this results in the number of unique combinations to equal the product of the cardinality of every nominal attribute.
3. **Conditional anomaly detection:** Compute anomaly scores of observations contained in the data partition through Isolation Forest.
4. **Classifier training:** Train a binary classifier after using the anomaly scores to replace the selected attributes.

iForest$_{CAD}$ aims to use expert knowledge to integrate information contained by meaningful nominal attributes. By partitioning the data according to the combinations of instances in these nominal attributes, anomaly detection can be performed conditional on instances that share these characteristics. Thus, rather than identifying global anomalies, anomalies are detected within given partitions. This allows to detect anomalies that are typically concealed when considering the global data structure. To domain experts, these “hidden" anomalies can potentially be more interesting than global anomalies.

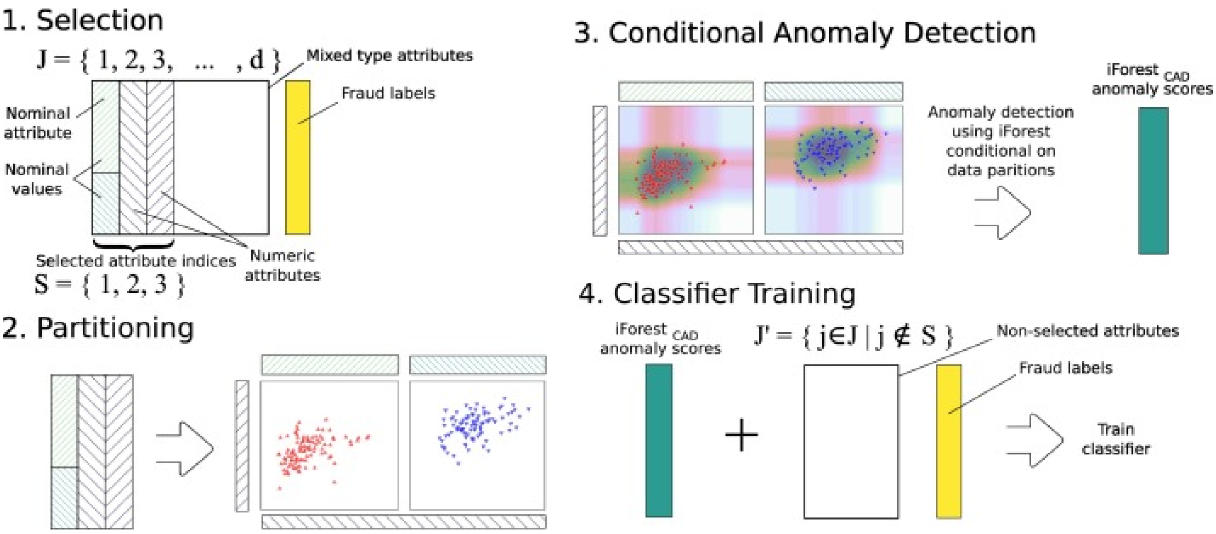


Figure 2.4: The operating principles of the $i\text{Forest}_{CAD}$ approach [36].

Furthermore, by performing anomaly detection conditioned on a data partition, performance becomes increasingly interpretable. Information from numerous attributes is summarized into a single anomaly score, which is then used as an input variable in the classifier training. This in turn causes a reduction in dimensionality, from the original d dimensions to $d' + 1$ for classification, where d' represents all dimensions that are not incorporated in the Conditional Anomaly Detection phase.

The pitfall in the $i\text{Forest}_{CAD}$ approach lies in the computation of the Cartesian product. When the cardinality of the nominal attributes increases, the number of data partitions increases with it. Therefore, selection of various nominal attributes quickly becomes limited, considering the induced sparsity in the data partitions increases when the Cartesian product becomes large. The author of $i\text{Forest}_{CAD}$ thus suggests attribute selection driven by expert knowledge, and limits the conditional anomaly detection phase to only three features in experiments.

Isolation Forest with Random Sampling of a Single Attribute Value

A straightforward extension to Isolation Forest is proposed in order to incorporate mixed-attribute data [42]. After randomly selecting an attribute, a test is performed on the attribute to determine the typology of the data. If it is a numerical attribute, the Isolation Forest continues with the typical splitting test and branching assignment. If the attribute is nominal, however, the procedure is altered. The algorithm then selects a single, unique value from the attribute, and assigns all instances that share this attribute value to the left branch. All instances that have a different value, are assigned to the right branch. In this paper, this method will be referred to as Isolation Forest with Single Sampling ($i\text{Forest}_{SS}$), for which the algorithms is described in Algorithm 3.

Algorithm 3 $IsolationTree(X', e, l)$

```
Inputs: $X'$ - input data, $e$ - current tree height, $l$ - height limit
Output: An Isolation Tree
if $e \geq l$ or $|X| \leq 1$ then
    return $exNode\{Size = |X'|\}$
else
    Let $Q$ be a list of attributes in $X'$
    Randomly select an attribute $Q_i \in Q$
    if $Q_i$ is nominal then
        Randomly select a value $q \in \mathrm{dom}(Q_i)$
        $X_l = filter(X', Q_i = q)$
        $X_r = filter(X', Q_i \neq q)$
    else
        Randomly select a split point $p \in \left(min(Q_i), max(Q_i)\right)$
        $X_l = filter(X', Q_i < p)$
        $X_r = filter(X', Q_i \geq p)$
    end if
    return $inNode\{Left = IsolationTree(X_l, e+1, l),$
                   $Right = IsolationTree(X_r, e+1, l),$
                   $SplitAttribute = Q_i,$
                   $SplitValue = p\}$
```

3
Methodology

In this chapter, the algorithms that have been introduced or adapted throughout this thesis are described in more detail. Note that the original Isolation Forest algorithm and methodology has been discussed in detail in Chapter 2 and will function as the cornerstone of the methods discussed in this chapter.

This chapter is divided into several sections. In order to address the first research objective, RO1, an approach to incorporate nominal attributes directly into an Isolation Forest is discussed in Section 3.1. This method depends on the sampling of categories in an nominal attribute, and allows nominal attributes to be incorporated into the Isolation Forest without using an encoding strategy. Additionally, a short proof of concept of the algorithm is demonstrated.

To address the second research objective, RO2, Section 3.2 will discuss the ternary Isolation Forest in more detail. The section dives into the motivation behind a ternary isolation tree and properly defines it. Then, using a similar strategy for binary isolation trees, the average path length derivations of the ternary isolation tree are revisited.

In Section 3.3, the MI-Local-DIFFI methodology is discussed in more detail. This local explanation method specific to Isolation Forest will be used throughout the remainder of this thesis. The indicators that composed the MI-Local-DIFFI method are evaluated with respect to $i\text{Forest}_{CS}$ and the ternary Isolation Forest.

Finally, in Section 3.4, the evaluation metrics used throughout the thesis experiments are discussed.

3.1. Isolation Forest for Mixed Attribute Data

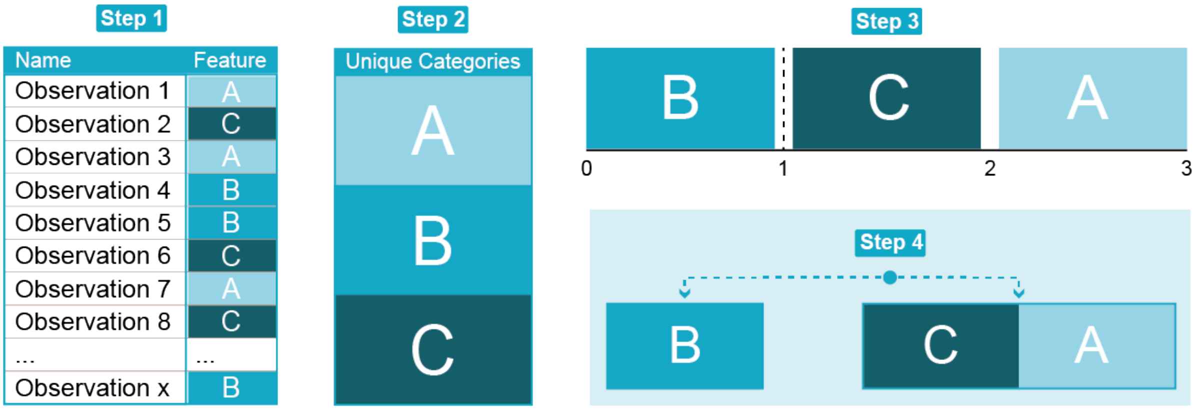
As discussed in Subsection 2.5.3, Isolation Forest cannot incorporate categorical data directly without encoding it to a numerical scale. In order to investigate and unlock the potential of categorical attributes in Isolation Forest, this thesis investigates the existing methods of categorical data incorporation. In the process, an additional method is proposed that directly incorporates categorical data without the need of prior encoding. Subsection 3.1.1 will elaborate on this method, which will be referred to as Isolation Forest with Categorical Sampling ($i\text{Forest}_{CS}$). Then, a short proof of concept is provided in Subsection 3.1.2, where the results of a label encoded Isolation Forest are compared to that of $i\text{Forest}_{CS}$.

3.1.1. $i\text{Forest}_{CS}$

The rationale behind $i\text{Forest}_{CS}$ stems from the desire to incorporate nominal data without encoding it beforehand, or inducing sparsity in data partitions through large Cartesian products, as in $i\text{Forest}_{CAD}$ [36]. In line with the essence of the Isolation Forest, the random, data-induced splits are maintained when considering a nominal attribute. Through testing for the typology of the feature in a given node, the splitting strategies will vary for nominal and numerical attributes. Furthermore, it avoids translation to a numerical scale, and thus the potential of assigning order where none exists, overlapping values, and increases in data dimensionality.

Consider a node v_{ij} of isolation tree t_i and let feature Q_{ij} be the feature for which a split test is performed in the node. This split test is dictated by the typology of the feature Q_{ij}. If feature Q_{ij} is a numerical attribute, the isolation tree continues in the usual manner as described in Section 2.3. If feature Q_{ij} is nominal, however, consider all unique values in the domain of Q_{ij} and let the cardinality of Q_{ij} be denoted by k_j. First, randomly select an integer $c \in [1, \lfloor \frac{k_j}{2} \rfloor]$, where symmetry of an isolation tree allows us only to consider the interval up to $\frac{k_j}{2}$. Next, select without replacement and with equal probabilities over all unique values, a subset $\mathcal{Q}$ containing c unique values from the domain of Q_{ij}. Now, the following splitting test is performed. All observations for which $Q_{ij} \in \mathcal{Q}$ are assigned to the left child node, and the remaining observations for which $Q_{ij} \notin \mathcal{Q}$ holds are assigned to the right branch.

To visualize this process, Figure 3.1 illustrates the different steps described when considering a nominal feature:



Name	Feature
Observation 1	A
Observation 2	C
Observation 3	A
Observation 4	B
Observation 5	B
Observation 6	C
Observation 7	A
Observation 8	C
...	...
Observation x	B

Figure 3.1: The procedure of splitting a nominal feature using iForest$_{CS}$. The steps are described in more detail in the text.

Using Figure 3.1 as an assisting visualization, the steps of the categorical sampling strategy are described step-by-step:

1. Imagine a particular node v_{ij}. In this node, a feature Q_{ij} is selected. In this case, the feature is categorized, with unique alphabetized categories.

2. Considering all observations in the node, the unique categories of the feature are appended. In this case, the feature has a cardinality of $k_j = 3$.

3. Randomly select an integer $c \in [1, \lfloor \frac{k_j}{2} \rfloor]$. In this case, $c = 1$. From the unique categories, randomly select $c = 1$ value, where the selection is performed without replacement and with equal probabilities. For illustrative purposes, this is visualized by aligning the unique categories in a shuffled way, and selecting the first category. This selection now represents subset $\mathcal{Q}$, which is category B in this case.

4. All observations for which $Q_{ij} \in \mathcal{Q}$, are assigned to the left branch. All other observations are assigned to the right branch. After this split, the Isolation Forest algorithm continues its process.

Through this splitting strategy, nominal data is incorporated into an isolation tree without prior encoding. It shares similarities with iForest$_{SS}$ [42], where a single attribute value is sampled every time a splitting test is performed on a nominal feature. Selecting a single category at a time, however, may reduce the probability of considering all categories under large cardinality and dimensionality. It is therefore interesting to consider whether selecting a set of attributes can improve the performance of Isolation Forest on mixed-attribute data. Experiments where iForest$_{CS}$ is evaluated against different nominal encoding strategies and other mixed-attribute adaptations of Isolation Forest, will be conducted in Chapter 4.

The algorithm for iForest$_{CS}$ is shown in Algorithm 4. As it is constructed in a similar fashion to Isolation Forest, the algorithm's complexity is similar. Notice, however, that the branching strategy changes when selecting a nominal feature. In Subsection 4.2.4, a runtime analysis is conducted to determine the sensitivity of the iForest$_{CS}$ to the percentage nominal features in the data and the respective cardinality.

Algorithm 4 Isolation Tree with Categorical Sampling

```
**Inputs:** $X'$ - input data, $e$ - current tree height, $l$ - height limit
**Output:** An Isolation Tree
**if** $e \geq l$ or $|X'| \leq 1$ **then**
    return $exNode\{Size = |X'|\}$
**else**
    Let $Q$ be a list of attributes in $X'$
    Randomly select an attribute $Q_i \in Q$
    **if** $Q_i$ is nominal **then**
        Randomly select an integer $c \in [1, \lfloor \frac{k_j}{2} \rfloor]$, where $k_j$ represents the cardinality of attribute $Q_i$
        Randomly select a set of $c$ attributes $\mathcal{Q} \in$ domain$(Q_i)$
        $X_l = filter(X', Q_i \in \mathcal{Q})$
        $X_r = filter(X', Q_i \notin \mathcal{Q})$
    **else**
        Randomly select a split point $p \in \big(min(Q_i), max(Q_i)\big)$
        $X_l = filter(X', Q_i < p)$
        $X_r = filter(X', Q_i \geq p)$
    **end if**
    **return** $inNode\{Left = IsolationTree(X_l, e+1, l),$
                 $Right = IsolationTree(X_r, e+1, l),$
                 $SplitAttribute = Q_i,$
                 $SplitValue = p\}$
```

3.1.2. Proof of Concept

In this section, a short proof of concept is discussed to demonstrate the bias induced by label encoding and the improved results of iForest$_{CS}$. It is argued that an Isolation Forest where all nominal attributes are encoded using label encoding is conceptually inadequate [36]. When a nominal attribute is encoded, the isolation tree stochastically samples over the interval induced by the integer valued encodings, ignoring that there should not be a relation between nominal attributes. This yields the consequence that observations assigned the lowest or highest mapped values will receive larger anomaly scores, purely because of the location in the mapped domain. This is illustrated in Figure 3.2, where anomaly scores are computed for two simple nominal attributes. The nominal attributes both have a cardinality $k = 5$ and every combination has an equal frequency.

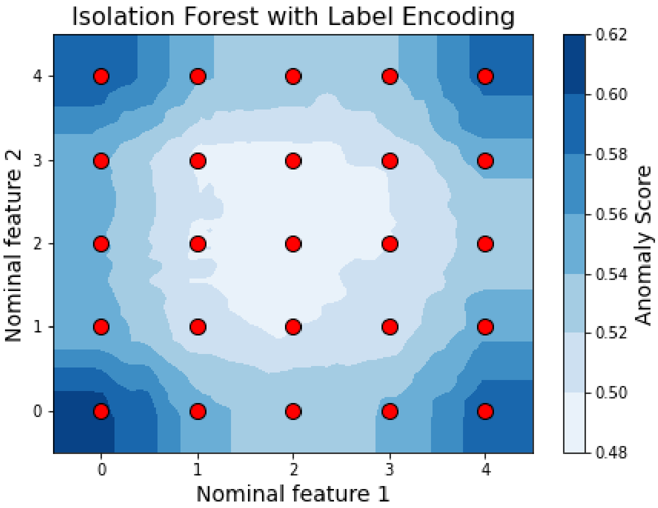


Figure 3.2: Two nominal attributes, each with 5 unique values, mapped to the numerical domain using label encoding. The contours demonstrate that the anomaly scores computed by Isolation Forest with label encoding increase towards the boundaries, whereas no ordering was originally present [36].

As shown in Figure 3.2, encoding of the nominal attributes with integer values introduces an order that is not necessarily there. This underlying order results in conceptually inadequate scoring of the observations. When using iForest$_{CS}$, no additional order is induced, and the resulting anomaly scores are more consistent. These are shown in Figure 3.3, where the anomaly scores are portrayed for the same two nominal attributes. Notice that no contour mapping is shown, as this is not possible in the discrete domain of a nominal attribute. The overall scores when using iForest$_{CS}$ are more consistent than the scores obtained through Isolation Forest with label encoding. Furthermore, the anomaly scores do not portray a bias towards observations lying on the boundary of the mapped domain. In fact, the anomaly scores for all observations are extremely similar with iForest$_{CS}$, whereas they deviated significantly when using Isolation Forest with label encoding.

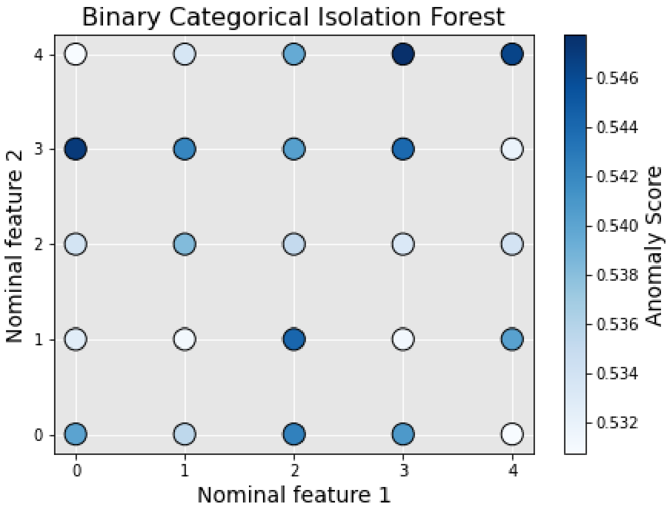


Figure 3.3: Two nominal attributes, each with 5 unique values. Using iForest$_{CS}$, anomaly scores are computed for all data sample. When compared to Figure 3.2, it is shown that the anomaly scores are i) consistent for all observations and ii) no bias towards boundary observations are introduced in the anomaly scores.

3.2. Ternary Isolation Forest

An adaptation to the classical Isolation Forest was proposed and analysed in the literature [4]. Rather than using binary trees, it was argued that the utilisation of ternary isolation trees can improve the detection of particular anomalies. This hypothesis was indeed confirmed using synthetic data constructed to test a density-based anomaly detection algorithm, namely High Contract Subspaces (HiCS) [43]. In this section, the methodology of a ternary Isolation Forest will be explained in more detail.

3.2.1. Ternary Isolation Tree

Before diving into the specifics, it is important to address the rationale behind the utilisation of ternary isolation trees. Let a *boundary anomaly* denote an anomaly situated in the extremes of an explicit feature distribution and let an *interior anomaly* be an anomaly that is not a boundary anomaly, as visualised in Figure 3.4. Through the Isolation Forest's random splitting of subspaces using binary isolation trees, a bias is induced towards boundary anomalies. Take for example the feature distributions of Figure 3.4. In the case of a boundary anomaly, using a binary splitting strategy can potentially isolate the anomaly with one test. However, for an interior anomaly, the best possible outcome is that the anomaly is detected after two splitting tests in Feature 1. Therefore, the probability of isolating a boundary anomaly is higher with a binary isolation tree than isolating an interior anomaly. Furthermore, when the dimensionality of the data increases, the probability of selecting a specific feature reduces proportionally. In this situation, it is likely that a particular feature occurs only once in a unique path. The probability of successfully isolating an interior anomaly therefore diminishes, unless the splitting structure of the tree is altered. To adjust for this situation, an Isolation Forest using ternary isolation trees is introduced.

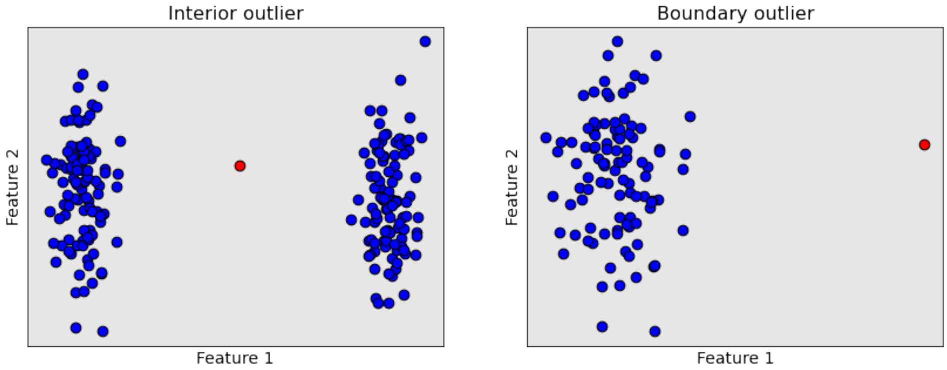


Figure 3.4: Visualisation of an interior and boundary anomaly with respect to feature 1. The anomaly is represented by the red observation.

In order to construct a Ternary Isolation Forest, the underlying isolation trees have to be redefined. These definitions are from [4]. The definition of a semi-proper ternary tree is used to introduce the ternary isolation tree:

Definition 3.2.1. *A semi-proper ternary tree t is a tree in which every node has 0, 2 or 3 children. When a node has 2 children, it is always the middle child that is missing*

Using this definition, a ternary isolation tree can be defined:

Definition 3.2.2. *A ternary isolation tree t is a semi-proper ternary tree. Let v be a node of a ternary isolation tree that is either an external node, or an internal-node with a test. A test in node v consists of a feature value Q_i and two split values p_1, p_2, where $p_1 < p_2$. The tests $Q_i < p_1$, $p_1 \leq Q_i < p_2$, $Q_i \geq p_2$ determine the assignment of a datapoint to the left, middle, or right node, respectively.*

A ternary isolation tree is a semi-proper ternary tree to account for the fact that observations do not necessarily have to fall within the feature range between p_1 and p_2. Thus, the middle child node can be empty. Yet, there will always be observations in the left and right child nodes, namely the minimum and maximum values in the feature range. In Figure 3.5, an example of a semi-proper ternary tree is visualized.

The construction of a ternary isolation tree is described in Algorithm 5 and the construction of the ternary Isolation Forest is described in Algorithm 6. The overall procedure of a ternary Isolation Forest is similar to that of a binary isolation tree, with a slight variation in the branching strategy. However, the height limit l is different for the two implementations and will impact the overall runtime as well. To determine the overall effect on runtime, Subsection 4.2.4 will evaluate the runtime of the different algorithms.

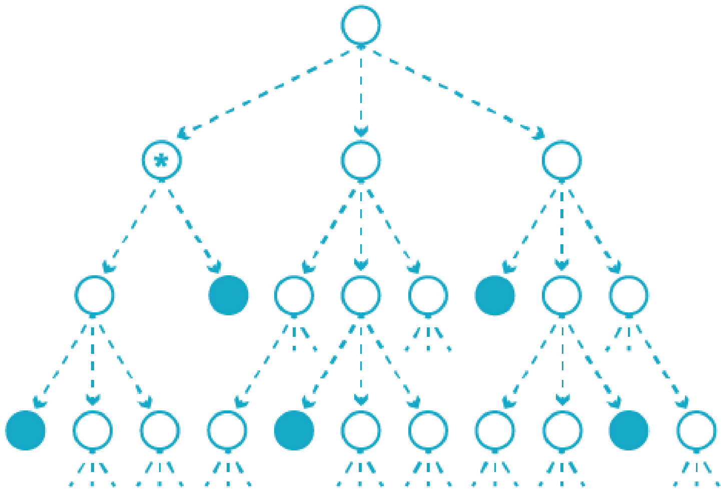


Figure 3.5: A semi-proper ternary tree. The * indicates a semi-proper splitting, with no observations being assigned to the middle child node.

Algorithm 5 $IsolationTree\_Ternary(X', e, l)$

```
Inputs: $X'$ - input data, $e$ - current tree height, $l$ - height limit
Output: An Isolation Tree
if $e \geq l$ or $|X| \leq 1$ then
    return $exNode\{Size = |X'|\}$
else
    Let $Q$ be a list of attributes in $X'$
    Randomly select an attribute $Q_i \in Q$
    Randomly select two split points $p_1$ and $p_2$ uniformly between the $max$ and $min$ values of attribute $Q_i$ in the data $X'$
    $X_l = filter(X', Q_i < p_1)$
    $X_l = filter(X', p_1 \leq Q_i < p_2)$
    $X_r = filter(X', Q_i \geq p_2)$
    return $inNode\{Left = IsolationTree(X_l, e+1, l)$,
                  $Middle = IsolationTree(X_m, e+1, l)$,
                  $Right = IsolationTree(X_r, e+1, l)$,
                  $SplitAttribute = Q_i$,
                  $SplitValue = (p_1, p_2)\}$
```

Algorithm 6 $IsolationForest\_Ternary(X, T, \psi)$

```
Inputs: $X$ - input data, $T$ - number of trees, $\psi$ - sub-sampling size
Output: A set of $T$ ternary Isolation Trees
Initialize $Forest$
Set height limit $l = ceiling(log_3 \psi)$
for $i = 1$ to $T$ do
    $X' = sample(X, \psi)$
    $Forest = Forest \cup IsolationTree(X', 0, l)$
end for
return $Forest$ =0
```

3.2.2. Average Path Length Ternary Isolation Tree

The calculation of an anomaly score for Ternary Isolation Forest is of similar fashion to that of the original Isolation Forest. Using the average path lengths of every observation, scores can be calculated to introduce a measure of susceptibility to isolation. However, it is important to note that the theoretical average path length given n observations is different for binary and ternary trees. This function $c(n)$ is namely used to adjust for prematurely pruned branches and as a normalising factor in the anomaly score, as discussed in Subsection 2.3.2.

In the original introduction of the Ternary Isolation Forest, an expression for the average path length of a ternary isolation tree, $c_t(n)$, was derived using a similar procedure as the average path lengths of the binary Isolation Forest. These derivation of the ternary theoretical average path lengths are revisited [4], and some mistakes are corrected. Results are then compared to an empirical evaluation of the average path lengths of a fully grown (ternary) Isolation Forest.

Similar to Subsection 2.3.2, the *external path length* of an isolation tree t is defined as the sum of the path lengths from the root node to every individual external node in the isolation tree.

Theorem 3.2.1. *The average external path length, $E_t(n)$ of a ternary isolation tree t grown to completion with n observations, is given by:*

$$E_t(n) = \begin{cases} 0 & if\, n = 1, \\ \frac{2}{n(n-1)} \left[\sum_{l=1}^{n-1} (3(n-l)-1)\, E_t(l)\right] + n & if\, n > 1. \end{cases} \tag{3.1}$$

Proof. First, assume the case $n = 1$. When dealing with a single observation, the observation is already isolated in the root node. Therefore, the average external path length is zero, $E_t(n)$.

Let $t_n(l)$ be an arbitrary ternary isolation tree with a root node who's left child contains $l \in \{1, 2, ..., n-1\}$ external nodes, who's middle child contains $m \in \{0, 1, ..., n-l-1\}$ and who's right child contains $n-l-m$ external nodes. The average external path length $E_t(n)$ can then be computed using the sum of average external path lengths of the root's left, middle, and right child nodes, $E_t(l)$, $E_t(m)$, and $E_t(n-l-m)$, respectively, plus n. This last term accounts for the fact that $E_t(l)$, $E_t(m)$, and $E_t(n-l-m)$ are a level deeper in the isolation tree with respect to the root node.

The number of unique allocations in the left, middle, and right child nodes, is represented by the expression:

$$\sum_{l=1}^{n-1}\sum_{m=0}^{n-l-1} 1 = \sum_{l=1}^{n-1}(n-l) = \frac{n(n-1)}{2}.$$

To determine the average external path length, the expectation over all subtree allocations is taken, assuming equal probability for every unique allocation. For $n > 1$, the following is derived:

$$\begin{aligned}
E_t(n) &= \frac{2}{n(n-1)} \sum_{l=1}^{n-1}\sum_{m=0}^{n-l-1} E_t(l) + E_t(m) + E_t(n-l-m) + n \\
&= \frac{2}{n(n-1)} \left[\sum_{l=1}^{n-1}\sum_{m=0}^{n-l-1} E_t(l) + E_t(m) + E_t(n-l-m)\right] + n \\
&= \frac{2}{n(n-1)} \left[\sum_{l=1}^{n-1}\left((n-l)E_t(l) + \sum_{m=0}^{n-l-1} E_t(m) + E_t(n-l-m)\right)\right] + n \\
&= \frac{2}{n(n-1)} \left[\sum_{l=1}^{n-1}\left((n-l)E_t(l) + \sum_{m=0}^{n-l} (E_t(m) + E_t(n-l-m)) - E_t(n-l) - E_t(0)\right)\right] + n \\
&= \frac{2}{n(n-1)} \left[\sum_{l=1}^{n-1}\left((n-l)E_t(l) - E_t(n-l) + 2\sum_{m=0}^{n-l} E_t(m)\right)\right] + n \\
&= \frac{2}{n(n-1)} \left[\sum_{l=1}^{n-1}\left((n-l)E_t(l) - E_t(l) + 2\sum_{m=0}^{n-l} E_t(m)\right)\right] + n \\
&= \frac{2}{n(n-1)} \left[\sum_{l=1}^{n-1}\left((n-l)E_t(l) - E_t(l) + 2\sum_{m=0}^{n-l} E_t(m)\right)\right] + n \\
&= \frac{2}{n(n-1)} \left[\sum_{l=1}^{n-1}(n-l-1)E_t(l) + 2\sum_{l=1}^{n-1}\sum_{m=0}^{n-l} E_t(m)\right] + n \\
&= \frac{2}{n(n-1)} \left[\sum_{l=1}^{n-1}(n-l-1)E_t(l) + 2\sum_{l=1}^{n-1}\sum_{i=1}^{n-l} E_t(l)\right] + n \\
&= \frac{2}{n(n-1)} \left[\sum_{l=1}^{n-1}(n-l-1)E_t(l) + 2\sum_{l=1}^{n-1}(n-l)E_t(l)\right] + n,
\end{aligned}$$

which yields the desired result:

$$\boxed{E_t(n) = \frac{2}{n(n-1)}\left[\sum_{l=1}^{n-1}(3(n-l)-1)\,E_t(l)\right] + n}.$$

□

This equation can be simplified to depend only on three preceding terms, rather than the computation of all preceding path lengths. This decreases computation time significantly. Here, the term $A(n) = n(n-1)E_t(n) - (n-1)(n-2)E_t(n-1)$ is used [4], and for $n > 2$ this can be expanded to:

$$\begin{aligned}
A(n) &= n(n-1)E_t(n) - (n-1)(n-2)E(n-1)\\
&= 2\sum_{l=1}^{n-1}(3(n-l)-1)\,E_t(l) + n^2(n-1) - 2\sum_{l=1}^{n-2}(3(n-1-l)-1)\,E_t(l) - (n-1)^2(n-2)\\
&= 2\left[\sum_{l=1}^{n-2}(3(n-l)-1)\,E_t(l) + 2E_t(n-1)\right] - \sum_{l=1}^{n-2}(3(n-1-l)-1)\,E_t(l) + (3n-2)(n-1)\\
&= 4E_t(n-1) + 6\sum_{l=1}^{n-2}E_t(l) + (3n-2)(n-1).
\end{aligned}$$

Therefore, for $n > 4$, the following two expressions hold:

$$\begin{aligned}
A(n) - A(n-2) = n(n-1)E_t(n) - (n-1)(n-2)E_t(n-1)&\\
-(n-2)(n-3)E_t(n-2) + (n-3)(n-4)E_t(n-3),&
\end{aligned}\tag{3.2}$$

and

$$\begin{aligned}
A(n) - A(n-2) &= 4E_t(n-1) + 6\sum_{l=1}^{n-2}E_t(l) + (3n-2)(n-1) - 4E_t(n-3) - 6\sum_{l=1}^{n-4}E_t(l) - (3n-8)(n-3)\\
&= 4E_t(n-1) + 6\sum_{l=1}^{n-2}E_t(l) + (3n-2)(n-1) - 4E_t(n-3)\\
&\quad -6\left(\sum_{l=1}^{n-2}E_t(l) - E_t(n-3) - E_t(n-2)\right) - (3n-8)(n-3)\\
&= 4E_t(n-1) + 6E_t(n-2) + 2E_t(n-3) + (3n-2)(n-1) - (3n-8)(n-3).
\end{aligned}\tag{3.3}$$

By equating Equation 3.2 and Equation 3.3, for all $n > 4$ the following expression can be derived:

$$\boxed{E_t(n) = C_1(n)\cdot E_t(n-1) + C_2(n)\cdot E_t(n-2) + C_3(n)\cdot E_t(n-3) + C_4(n),}\tag{3.4}$$

where the coefficients $C_i(n)$ for $i = 1,2,3,4$ are calculated as follows:

$$\boxed{\begin{aligned}
C_1(n) &= \frac{4+(n-1)(n-2)}{n(n-1)},\\
C_2(n) &= \frac{6+(n-2)(n-3)}{n(n+1)},\\
C_3(n) &= \frac{(2-(n-3)(n-4)}{n(n+1)},\\
C_4(n) &= \frac{(3n-2)(n-1)-(3n-8)(n-3)}{n(n+1)}.
\end{aligned}}$$

Using the expression of the average external path length in Equation 3.4, the average depth of the leaves can be calculated, considering $c_t(n) = \frac{E_t(n)}{n}$. To validate this expression, an experiment is conducted where both binary and ternary forests are grown to completion. This is done using a data set of varying sizes, sampled from a uniform distribution. By training an Isolation Forest on these data sets, without any maximum depth parameter, the average path lengths of all observations and trees can be calculated. This validation is shown in Figure 3.6.

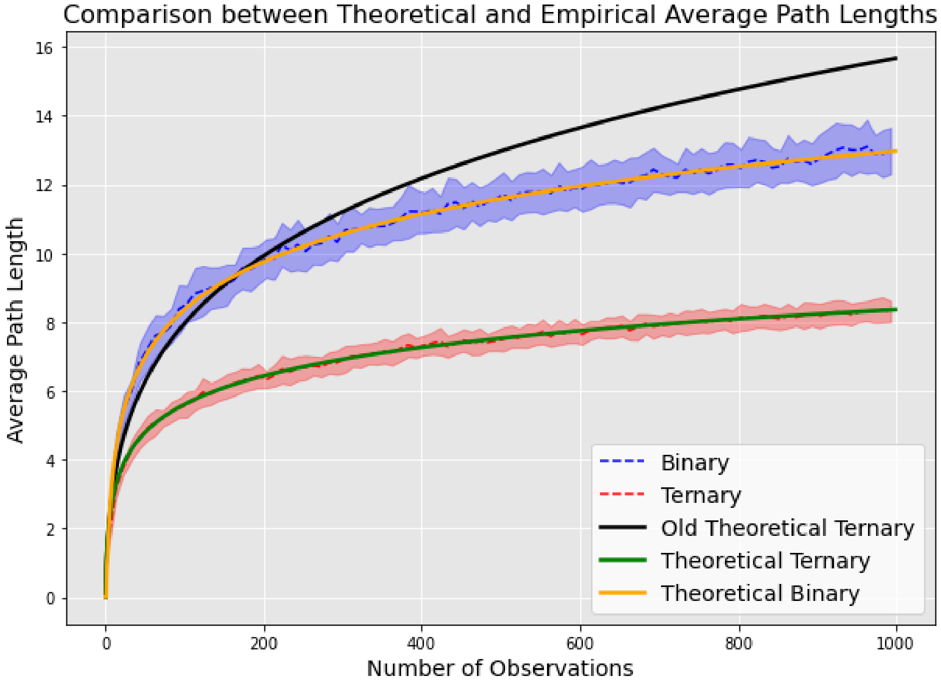


Figure 3.6: A comparison between the theoretically derived average path lengths of both binary and ternary trees, and the empirical average path lengths of complete binary and ternary Isolation Forest. The analysis was performed through the use of a uniformly sampled data set of varying number of observations, taking the average of the path lengths from all 50 trees in the forests. The shaded region around the experimental average represents the interval with respect to the standard deviation of the average path length. The dashed blue and red lines represent the empirically computed average path lengths for the binary and ternary Isolation Forest, respectively. Since the theoretically computed average path lengths coincide so closely, these dashed lines are difficult to observe. Note, they lie centrally within the shaded regions.

In Figure 3.6, it is observed that there is a large deviation between the theoretically derived ternary path lengths. When using the original theoretically derived average path length of a ternary isolation tree, some problems come to light. First, the average path length $c_t(n)$ is used as a normalization constant for the anomaly score computation, as shown in Equation 2.7. If $c_t(n)$ is consistently larger than the true average path length, the anomaly scores become inflated. Although the ordering of anomalies will remain consistent, the classification of anomalies based on an anomaly score threshold loses significance. Second, imagine a node at the height limit of a ternary tree. To compute the average path length of all observations in this node, the path length is added to the average path length of the unbuilt sub-tree, $c_t(Size)$. Therefore, any observation that reaches the height limit in an Isolation Tree will be assigned an incorrectly large path length, which in turn impacts the anomaly scores and the local explanations.

Also, Figure 3.6 demonstrates that the newly derived ternary average path length coincides with the empirical average. Furthermore, it demonstrates that the average path length of a ternary isolation tree is indeed lower than that of a binary isolation tree. This is in line with intuition, as a ternary tree divides the data into more branches containing fewer points. A path length of a ternary tree is thus shorter, as fewer nodes are required to isolate an observation.

With increased dimensionality, the extra branch allows for the welcome additional fragmentation of a feature. If the probability of selecting a particular feature shrinks due to increased dimensionality, it is favourable to have more detailed splitting strategies in every node. However, the question arises whether the lower average path length impacts the selection of a particular feature subspace of the data. If an anomaly lies in a given subspace of features, will it not be less probable to select this combination of features with a shorter average path length? This question is addressed further in Section 4.4.

3.3. MI-Local-DIFFI

The MI-Local-DIFFI method [4] is an adaptation of the DIFFI method [3]. Rather than the original global explanations provided by the DIFFI method, the purpose of the MI-Local DIFFI method is to provide local explanations. Thus, if a particular observation is classified as an anomaly, the MI-Local-DIFFI method provides insight into the most important features responsible for isolation. To achieve this goal, the assumption is made that information contained in the nodes and the splitting structure of the isolation trees can be manipulated to determine a measure of feature importance.

In Section 3.3, it was explained that the feature importance scores are assigned resulting from the following indicators:

1. **Path length indicator:** Assigns a feature importance weight by considering the anomaly's path length in every isolation tree.
2. **Split proportion indicator:** Assigns a feature importance weight by observing the proportion of observations that are assigned to the same branch as the traced anomaly, for a particular feature split.
3. **Split interval length indicator:** Assigns a feature importance weight by considering the proportion of the sub-interval length that contains the anomaly to the overall feature range in a specific node.

Now, these indicators will be explored in more detail, explaining the computation of weights that constitute the feature importance measure.

Path Length Indicator

Let $PL(o, t_i)$ represent the path length of the external node of an anomaly o in a specific isolation tree t_i. Then, the weight corresponding to the path length is defined as:

$$w^{PL}(o, t_i) := \max\left\{0.1, \quad \min\left[1, \quad 1 - \frac{PL(o, t_i) - PL_{lower}}{PL_{upper} - PL_{lower}}\right]\right\}, \tag{3.5}$$

where PL_{lower} and PL_{upper} are defined as lower and upper path lengths and are defined as:

$$\begin{aligned} PL_{lower} &= 1, \\ PL_{upper} &= \lceil 2(\log\psi + \gamma - 1) \rceil . \end{aligned}$$

Notice that the upper path length is defined as the average path length of a binary isolation tree, derived in Subsection 2.3.2. This is done since anomalies are more susceptible to isolation, thus having shorter path lengths than average. Equation 3.5 assigns a feature importance weight corresponding to the path length of the path an anomaly occurs in. Features occurring in a longer path are deemed less important and are therefore assigned a lower weight. However, it is not desirable to have a path length weight assigned to zero, as it penalizes a feature that occurs at the end of a path unreasonably harsh. Therefore, the lowest path length weight possible is set to 0.1 instead.

Split Proportion Indicator

In order to define the split proportion weight, some additional notation is clarified. First, let v_{ij} represent a node in the isolation tree t_i and let Q_{ij} be the feature for which a split test is performed in node v_{ij}. Let q_{ij} represent the number of observations in v_{ij}, and let q^o_{ij} denote the observations in the child node containing anomaly o of node v_{ij}. Then, let the vector $\overrightarrow{w}^{SP}(o, i) = \left(w_1^{SP}(o, i), \ldots\ w_{PL}^{SP}(o, i)\right)$, where PL represents $PL(o, i)$ defined above. Then:

$$w_j^{SP}(o, i) := \begin{cases} 0 & \text{if } q_{ij} = 2, \\ 1 - \frac{q^o_{ij} - 1}{q_{ij} - 2} & \text{if } q_{ij} > 2. \end{cases} \tag{3.6}$$

A feature that induces an extreme imbalance in the sizes of its child nodes, provides evidence of stronger isolating capabilities. In other words, when the child node of v_{ij} contains a significantly smaller data sample than the data size in v_{ij} originally, the split in feature Q_{ij} was a success with respect to anomaly o. Therefore, a high split proportion weight is assigned when $q^o_{ij} << q^{ij}$. Furthermore, there is a probability that a feature occurs more often in an anomaly path. When this is the case, only the highest w^{SP} is used for the overall feature importance computation. This is done to reward splits that induce imbalance and not penalise poor feature splits.

Split Interval Length Indicator

The final indicator of the MI-Local-DIFFI method is the split interval length indicator. This indicator incorporates information using the value of the split test, and its position with respect to the feature interval. It is assumed that a feature split is more imbalanced when a split value is chosen towards the extremes of a split interval. When this occurs, it becomes more likely to isolate a particular observation without it demonstrating clear anomalous behaviour in the feature, as demonstrated in the left figure of Figure 3.7. The split interval length indicator wants to penalise features for which the split value is close to the interval boundary, and reward features that isolate when split values are more centrally located. The feature in the right figure of Figure 3.7, will thus receive a higher split interval weight.

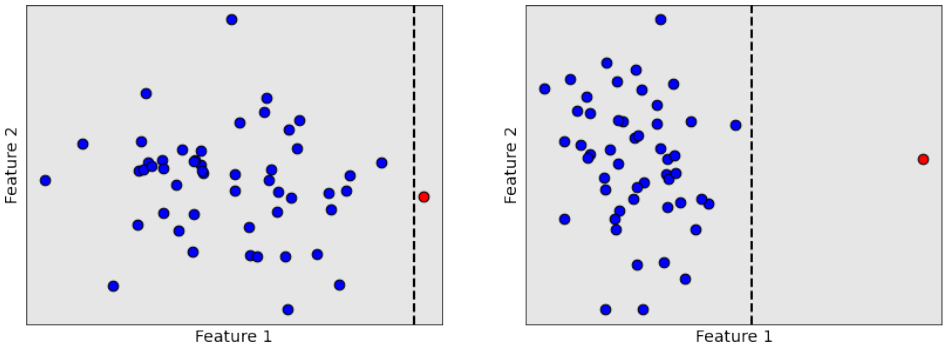


Figure 3.7: Two feature distributions, where the red observation is isolated through performing a split test at the dashed, black line in Feature 1. In the left figure, the split value is chosen close to the maximum value of the feature interval, which causes the red observation to become isolated. On the other hand, the right image isolates the red observation through a more centered split value. The split interval length indicator tries to account information regarding the choice of split value, and will assign a large weight to the feature in the right figure.

To quantify this, some notation has to be defined and clarified. Every node v in an isolation tree has a split feature Q, a split value p, and a data sample X'. The *feature interval* of a specific node is represented by the interval $[a, b]$, where $a = \min(X')$ and $b = \max(X')$. The *anomaly split interval* of node v with respect to anomaly o is defined as either the interval $[a, p]$ or $[p, b]$ depending on whether the anomaly o descends down the left or right branch, respectively. Using this information, the split interval of a node v_{ij} in the path of anomaly o in tree t_i is defined as:

$$si\left(o, v_{ij}\right) := \frac{\left|\text{anomaly split interval of } v_{ij} \text{ w.r.t } o\right|}{\left|\text{feature interval of } v_{ij}\right|} \tag{3.7}$$

The split interval weight is then defined as the vector $\vec{w}^{SI}(o,i) = \left(w_1^{SI}(o,i), \ldots\, w_{PL}^{SI}(o,i)\right)$, where PL represents $PL(o,i)$ defined above, where

$$w_j^{SI}(o,i) = 1.5 - \frac{1}{si\left(o, v_{ij}\right) + 1} \tag{3.8}$$

Algorithm MI-Local-DIFFI

Combining the defined indicator in the sections above, the MI-Local-DIFFI algorithm [4] is stated as in Algorithm 7:

3.3.1. Changes to MI-Local-DIFFI

In this thesis, MI-Local-DIFFI is used extensively to calculate the feature importance and explain anomalies at a local level. When the methodology was proposed in [4], it was compared to both TreeSHAP [28] and Alter-One-Feature, which is adapted from a method that examines individual columns contributions to classification errors [44]. It was determined that the MI-Local-DIFFI showed excellent performance with respect to runtime and overall performance when compared to the other two methods. It was shown that

Algorithm 7 MI-Local-DIFFI

```
Inputs: $o$ - anomaly, $IF = \{t_1, \dots, t_T\}$ - Isolation Trees, $PL(o,i)$ - Path length of $o$ in Isolation Tree $i$
Output: $FI$ - Feature Importance
Let $G_1, \dots, G_{n_F}$ be the unique features to create the Isolation Forest.
Initialize: $FI = \overrightarrow{0}$, with length equal to number of features $n_F$.
Initialize: occurrence$(F) = \overrightarrow{0}$, with length equal to number of features $n_F$.
for $t_i$ in $IF$ do
    Let $\overrightarrow{Q}_i = [Q_{i,1}, \dots, Q_{i,m}]$ represent the features in every node in $Path(o,i)$
    Calculate $w^{PL}(o,i)$ using Equation 3.5 .
    Calculate $\overrightarrow{w}^{SP}(o,i)$ using Equation 3.6 .
    Calculate $\overrightarrow{w}^{SI}(o,i)$ using Equation 3.8 .
    for $k$ in 1 to $n_F$ do
        if Feature $G_k$ occurs in $\overrightarrow{F}_i$ then
            Let $j_k$ be the location of the best split feature $G_k$ in $Path(o,i)$.
            occurrence$(k)$ = occurrence$(k) + 1$
            $\sigma(k) = w^{PL}(o,i) + w^{SP}_{j_k}(o,i) + w^{SI}_{j_k}(o,i)$
            $FI(k) = FI(k) + \sigma(k)$
        end if
    end for
end for
$FI = \frac{FI}{occurrence}$
return $FI$
```

AOF and MI-Local-DIFFI agree more extensively with ground-truth outlying aspects of the synthetic data [43] than TreeSHAP. On the other hand, MI-Local DIFFI is generally faster than TreeSHAP, which in turn is much faster than AOF.

This thesis will focus on the incorporation of categorical data, as well as further analysing performance of ternary Isolation Forest. In order to incorporate the MI-Local-DIFFI method, some changes have to be introduced. These are divided into three specific sections, namely Minor Changes, Changes for Ternary Isolation Forest, and Changes to Incorporate Categorical Data.

3.3.2. Minor Changes to MI-Local DIFFI

There is a minor change that is proposed for the split proportion indicator of the MI-Local-DIFFI. It is believed that the split proportion weight defined in Equation 3.6 is slightly biased towards features that isolate an anomaly further down the $Path(o,i)$, when fewer instances are in the nodes. To illustrate this, take the following examples, using the notation defined when defining the split proportion indicator:

- Imagine a node v_{ij} that contains 100 observations, $q_{ij} = 100$, and that the anomaly o is isolated after a split in v_{ij}, $q^o_{ij} = 1$. Then using Equation 3.6, the split proportion weight is calculated to be $w^{SP}_j(o,i) = 1$.

- Imagine a node v_{ij} that contains 3 observations, $q_{ij} = 3$, and that the anomaly o is isolated after a split in v_{ij}, $q^o_{ij} = 1$. Then using Equation 3.6, the split proportion weight is also calculated to be $w^{SP}_j(o,i) = 1$.

Equation 3.6 always assigns the maximum weight of 1 the moment a feature is used to isolate an instance. This means, however, that the split proportion weight of both examples are identical, even though example 1) clearly illustrates a better splitting feature than example 2). To adjust for this bias, simply consider the proportion of q^o_{ij} to q_{ij} as shown in Equation 3.9.

$$w^{SP}_j(o,i) := \begin{cases} 0 & \text{if } q_{ij} = 2 \\ 1 - \frac{q^o_{ij}}{q_{ij}} & \text{if } q_{ij} > 2 \end{cases} \tag{3.9}$$

Changes to MI-Local-DIFFI for Ternary Isolation Forest

To adapt MI-Local-DIFFI for Isolation Forest, the following changes are implemented:

- **Path Length Indicator:**
  For the path length indicator, only a slight alteration is necessary. Currently the upper path length is set to:

  $$PL_{upper} = \lceil 2(\log\psi + \gamma - 1) \rceil$$

  which represents the theoretically derived average path length of a binary isolation tree. Continuing the utilization of this specific upper path length for ternary Isolation Forest still coincides with the purpose of the path length weight. However, for proper form and following the intuition that an anomaly occurs in shorter path lengths, the theoretically derived average path length for ternary Isolation Forest should be used.

- **Split Proportion Indicator:**
  A ternary isolation tree is a semi-proper ternary tree to account for the fact that observations do not have to fall between the splitting values p_1 and p_2, where $p_1 < p_2$ as explained in Subsection 3.2.1. That this is a possibility, is not considered for the split proportion indicator applied to binary trees. It is however important to define the weight such that it reflects this semi-proper property of a ternary isolation tree. Equation 3.9 is adapted to reflect a ternary isolation tree:

  $$w_j^{SP}(o,i) := \begin{cases} 0 & \text{if } q_{ij} = 2 \\ 0 & \text{if } q_{ij}^o = 0 \\ 1 - \frac{q_{ij}^o}{q_{ij}} & \text{if } q_{ij} > 2 \end{cases} \tag{3.10}$$

- **Split Interval Length Indicator:** For the split interval length indicator, the split interval that coincides with the middle child node must be taken into account. Therefore, let the feature interval be once again defined as the interval $[a, b]$, where $a = \min(X')$ and $b = \max(X')$. Let p_1 and p_2 represent the split values in the feature Q in a specific node v. Then the *anomaly split interval* of node v with respect to anomaly o is defined as the interval [a, p_1] for the left child, [p_1, p_2] for the middle child, and [p_2, b] for the right child. This information can be used to compute the split interval of a node v_{ij} in the path of anomaly o in tree t_i with Equation 3.7. Then the split interval weight is computed using Equation 3.8.

Incorporating Categorical Data in MI-Local-DIFFI

This thesis focuses on mixed-attribute data sets and how nominal attributes should be incorporated when considering Isolation Forest and local explanation methods. When considering the MI-Local-DIFFI local explanation method tailored to Isolation Forest, it is interesting to see to what extent this framework can be maintained when different data typologies are incorporated.

First, consider the first two indicators of the MI-Local-DIFFI method, namely the path length indicator and the split proportion indicator. Both of these indicators address characteristics of the isolation tree that do not focus on the underlying feature data typology or distribution. Irregardless of the feature's data typology, a feature importance weight can be assigned by considering the path length of a particular anomaly in every isolation tree or the proportion of observations in a child node. These first two indicator are therefore applicable to nominal features and can be used to determine local explanations of mixed-attribute data.

The third indicator of the MI-Local-DIFFI method, the split interval length indicator, is dependent on the underlying feature distribution. Definitions of the anomaly split interval and the feature interval are all dependent on a notion of distance. The calculation of such a distance metric is not as straightforward when considering nominal attributes. With no clear underlying similarity or distance measures of categories in an attribute, the calculation of a split interval length indicator becomes counter-intuitive. Therefore, the third indicator in not employed when considering mixed-attribute data.

In Appendix C, some thought has been dedicated to the incorporation of a hybrid split interval length indicator. This indicator uses the cardinality of a nominal feature to determine a split interval weight for a nominal feature. Section 6.3 proposes to conduct further research into this indicator, as the effectiveness of the hybrid split interval length indicator has not been concluded in this thesis.

3.4. Evaluation Metrics

In the typical setting of transaction monitoring, there is no prior information available about suspicious transactions. Inherently it is thus an unsupervised problem, for which no ground-truth labels or outlying aspects are necessarily predefined. The question that naturally arises, is how the performance of anomaly detection problems can be quantitatively evaluated. A large proportion of literature uses either case studies or synthetically generated data to intuitively and qualitatively evaluate underlying anomalies in an unsupervised setting [12] when ground-truth labels are unavailable.

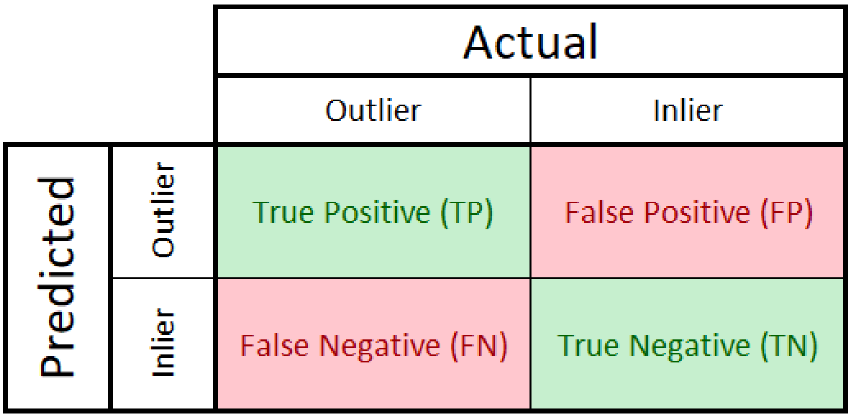
Fortunately, situations exist for which there is ground-truth data available. This thesis will make use of synthetic data sets that contain ground-truth labels of (non-trivial) anomalies. With this data available, multiple evaluation metrics are addressed that measure both the performance of the algorithms' detected anomalies as well as the effects of categorical encoding on the detection performance and explanation methods.

In this section, the measures used to quantify the performance of the different algorithms are described. The section is divided into several sub-sections. First, the overall performance in detecting anomalies of an anomaly detection algorithm is defined. Then a measure to quantify the performance of a local explanation method will be presented. Finally, a brief description is presented of the statistical significance test used to determine whether certain results presented in Chapter 4 demonstrate significant differences.

3.4.1. Evaluating Anomaly Detection Performance

The Isolation Forest algorithm outputs an anomaly score, which in turn may be used to determine a decision threshold between a normal instance and an anomaly. Optimizing this boundary is dependent on the weight that is placed on misclassification of either normal or outlying instances. Restricting this threshold to limit the number of declared anomalies, causes the algorithm to miss too many actual anomalies. The opposite, however, might just be as undesirable; resulting in an algorithm that classifies many normal points as anomalies. To conceptualize this necessary trade-off and define performance metrics, the confusion matrix is introduced.

The confusion matrix is a table layout that visualizes the performance of an algorithm's predicted class as opposed to the actual class of an instance or observation. Suppose an algorithm predicts whether a point is an anomaly or a normal observation. An algorithm can predict something correctly if it predicts an anomaly that is indeed a true anomaly (True Positive) or when a predicted normal observation is indeed a true normal observation (True Negative). On the other hand, an algorithm can make incorrect predictions as well. This occurs when an algorithm predicts a true anomaly as a normal instance (False Negative), or vice versa, when the algorithm predicts a normal observation as an anomaly (False Positive). By combining the possible combinations of true and predicted anomalies and normal observations, the confusion matrix is obtained as shown in Figure 3.8



		Actual	
		Outlier	Inlier
Predicted	Outlier	True Positive (TP)	False Positive (FP)
	Inlier	False Negative (FN)	True Negative (TN)

Figure 3.8: Confusion matrix comparing the predicted algorithm results to the true classes.

Using the terminology from the confusion matrix in Figure 3.8, further metrics can be defined that are important for the performance evaluation metrics [12]. Consider a threshold t of the set on anomaly scores. Then, the predicted anomaly set is indicated by $S(t)$. With a change in the value of the threshold t, this predicted anomaly set varies in size. Let G represent the ground-truth set of the overall data. For a given threshold value t, the *precision* is defined as the predicted anomalies that are indeed anomalies in the ground-truth set against the total size of predicted anomalies.

$$Precision(t) = \frac{|S(t) \cap G|}{|S(t)|} = \frac{TP}{TP + FP}.$$

The precision value is not strictly monotonic, as the predicted anomaly set size $|S(t)|$ and the correctly predicted anomalies $|S(t) \cap G|$ behave differently to changes in t. The *recall* is in turn defined as the predicted anomalies that are indeed anomalies in the ground-truth set against the total size of the ground-truth set.

$$Recall(t) = \frac{|S(t) \cap G|}{|G|} = \frac{TP}{TP + FN}.$$

The Precision and Recall as functions of t can be used to generate a curve. This curve is referred to as the *Precision-Recall curve* (PR) and will be used for evaluation purposes extensively. Note that this curve is not necessarily monotonic. Furthermore, the use of a *Receiver Operation Characteristics Curve* (ROC) is used, which is related to the PR-curve but slightly more intuitive. For this curve, the *True-Positive Rate* (TPR) and the *False-Positive Rate* are plotted against each other. The definition of the TPR is identical to that of the Recall, while the FPR is defined as the proportion falsely classified anomalies to the actual inliers. Thus, with proper notation, for all data D and the ground truth positives G:

$$TPR(t) = Recall(t),$$

$$FPR(t) = \frac{|S(t) \cap G|}{|D - G|} = \frac{FP}{FP + TN}.$$

Using these two curves, the performance of algorithms can be evaluated. For example, if a PR or ROC curve of a particular algorithm strictly dominates that of another algorithm, it can be concluded that the first algorithm is superior in terms of performance. However, when a curve is not strictly dominated, such straightforward conclusion can not be drawn. This situation indicates that the algorithms behave differently for varying thresholds and thus performance varies according to the anomaly score threshold. In practice, the area under the ROC curve (AUC) and the area under the PR curve (AUPRC) are used as a representation of overall effectiveness [12]. These measures will be used throughout this thesis, although it is acknowledged that AUC results are overly optimistic when dealing with unbalanced classes.

3.4.2. Evaluating Explanation Performance

For the evaluation of the local explanations, the feature importance scores and its respective ranking are used [4]. When considering a data set where the anomaly containing subspace is known, a binary indicator vector $\tau \in \{0,1\}^d$ can be constructed to indicate the features containing anomalies. Using this vector, and the feature importance ranking of the explanation method FI, the AUC and AUPRC can be computed in order to evaluate the performance of an explanation method. For every anomaly o in the data set, AUC($\tau(o), FI(o)$) and AUPRC($\tau(o), FI(o)$) are computed, after which the performance for the entire anomaly set is taken as:

$$AUC_{FI} = \frac{1}{n_o} \sum_{o=1}^{n_o} AUC(\tau(o), FI(o)), \tag{3.11}$$

$$AUPRC_{FI} = \frac{1}{n_o} \sum_{o=1}^{n_o} AUPRC(\tau(o), FI(o)), \tag{3.12}$$

where n_o represents the number of anomalies in the data set X. Using these measures, the results of the performance of MI-Local-DIFFI method will be quantified throughout the experiments in 4.

3.4.3. Testing Statistical Significance

With several experiments in Chapter 4, it is not immediately clear if differences in performance measures are statistically significant. To address this, the unpaired samples Student's t-test is used [45]. Under the null hypothesis that there is no significant difference in the population means, the t-statistic is calculated and used to determine whether the null hypothesis can be rejected. In order for this statistical test to hold, the following assumptions must be met:

- The data that forms the basis of the test should be sampled independently.
- The two populations that are compared should be homoscedastic. This means that the populations tested have similar variances, which will be tested in this thesis using the Levene's test.
- The population data should be approximately normally distributed. This is tested throughout this thesis with the Shapiro-Wilk test [46].

Note that violations of the last two assumptions are not as severe. The sampling distribution remains robust with an increased sample size and with similar population sizes. This is constantly ensured throughout the experiments conducted in this thesis.

4 Synthetic Experiments

Before utilising the newly proposed $i\text{Forest}_{CS}$ in the context of detecting anomalies in the real-life transaction data of Triodos Bank, experiments are conducted to explore the algorithm's performance. The methodology is compared to numerous encoding strategies used in practice, as well as the $i\text{Forest}_{CAD}$ method proposed in the literature [36]. In order to conduct these experiments, synthetic data sets are used and manipulated extensively. Using both independently sampled observations and conditionally sampled observations, the methods are evaluated on detection performance, sensitivity to data characteristics, explanation performance, and complexity.

Furthermore, in this section an analysis is conducted on the behaviour of ternary Isolation Forest. Through earlier research, the performance of a ternary Isolation Forest showed promising improvements to the capability to detect anomalies when compared to the standard binary Isolation Forest [4]. These experiments are revisited considering the newly calculated average path lengths, but also after reconsidering the height limit of an isolation tree. Moreover, the performance of a ternary Isolation Forest has never been evaluated with respect to the performance of local explanations.

This chapter is divided into specific sections. First, Section 4.1 will briefly elaborate on synthetic data and the benefits reaped from its analysis. Second, Section 4.2 will discuss the sensitivity and runtime analysis of $i\text{Forest}_{CS}$ using synthetic data with independent features. Third, Section 4.3 visualizes the benefits of incorporating nominal attributes to the detection of anomalies in a synthetic data set with nominal-numerical dependencies. This data set is then further utilized to test the MI-Local-DIFFI indicators as applied to mixed-attribute data. Fourth, Section 4.4 focuses on the ternary Isolation Forest, and evaluates both the anomaly detection performance as well as the performance of the MI-Local-DIFFI feature importance scores. Finally, in Section 4.5, the parameters defined in the original Isolation Forest paper are taken into consideration. The section addresses some inconsistencies in the standard implemented parameters.

4.1. Synthetic Data

This thesis depends on synthetic data to analyse and evaluate the different Isolation Forest implementations. Throughout this chapter, three distinct synthetic data sets approaches are defined and generated. In order to fully understand the purpose of utilising synthetic data, emphasis is first placed on the benefits of using synthetic data.

First, generating data allows for the complete control over the characteristics of the data. There is no limitation to the number of observations, underlying distributions, and number of attributes used to construct the data. Furthermore, for specifically nominal features, the total number of nominal attributes and an attribute's cardinality can be altered. This allows for experiments to range from intuitive and visually comprehensible, to complex and overly detailed. Moreover, changing the characteristics of the data allows further analysis into the performance and runtime sensitivity. Gaining an understanding of the impact data set characteristics have on the overall algorithm performance will prove beneficial in determining the effectiveness of an adaptation.

Finally, with knowledge of the underlying distribution used to generate the data, it is possible to express an anomaly in terms of a probability measure. Specifically, the joint probability density functions can be utilized to address whether an observation is an anomaly. Namely, the observations with the lowest joint probability density values can be defined as the data set's anomalies. Furthermore, the marginal probability density functions can be utilised in order to gain an indication of which attributes contributed most to the anomaly's behaviour.

Anomalies can also be defined using other measures of outlierness, such as the distance or density-based proximity measures. However, in this thesis a probability measure is used to define anomalies since the underlying distributions from which the data is sampled are defined beforehand. Therefore, the major drawback that typically accompanies probabilistic anomaly detection methods of attempting to fit the data to a distribution, is countered. Furthermore, the probability measure can be readily applied to mixed-attribute data.

4.2. Experiments with Independent Features

In this section, data sampled from independent features are used in order to address the sensitivity of different Isolation Forest approaches to characteristics of the synthetic data. First, Subsection 4.2.1 will elaborate on the purpose of the experiments and address the key findings the section hopes to clarify. Next, Subsection 4.2.2 will address the synthetically generated data that is utilised to achieve the results. Finally, Subsection 4.2.3 and Subsection 4.2.4 will discuss the results of the anomaly detection performance sensitivity and the runtime complexity of the different algorithms, respectively.

4.2.1. Purpose of Experiments

The entire intuition behind introducing the $i\text{Forest}_{CS}$ approach to handle mixed-attribute data with Isolation Forest is to circumvent the necessity of using categorical data encoding. It is argued that with the encoding of nominal attributes either a non-existing order is introduced through translation to a numerical scale, or information is lost in general. Through random sampling the categories in a nominal attribute, the essence of the Isolation Forest approach is respected, namely using data-induced random trees to determine the degree of isolation of a particular observation.

In this section, the purpose is two-fold. First, the experiments are conducted to gain an understanding of the performance sensitivity to various parameters to the synthetic data. By changing parameters of the data, insight into the most popular encoding strategies in practice are gained and compared to the behaviour of our $i\text{Forest}_{CS}$ approach. Next, the generation of synthetic data allows us to address the runtime sensitivity of $i\text{Forest}_{CS}$ to parameters of the data.

4.2.2. Synthetic Data with Independent Features

The first data set that is used throughout this chapter, independently samples observations from different distributions depending on the feature type. This data set is primarily used to address the sensitivity of the mixed-attribute Isolation Forest adaptations in terms of performance and runtime. It contains a mixture of numerical and nominal features, in which particularly parameters of the categorical feature distribution are altered.

All numerical observations are independently sampled from standard normal distributions. The tails of the Gaussian probability distributions allows for anomalies to be situated at extremes of the distribution, assisting the Isolation Forest method considering its bias towards distribution extremes. The nominal features are sampled using a Multinomial-Dirichlet distribution. The Multinomial-Dirichlet distribution is explained in slightly more detail in Appendix A. The Multinomial distribution allows for the sampling of categorical attributes, while the Dirichlet prior automates the selection parameters of the Multinomial distribution.

4.2.3. Results of Performance Sensitivity

In order to gain perspective into the effect different parameters have on the detection of anomalies, a sensitivity analysis is conducted. In this analysis, different parameters that particularly influence the nominal attributes of the mixed-attribute data set are altered. Using the distinct combinations of these parameters allows for the generation of multiple synthetic data sets that can be utilised.

In this sensitivity analysis, the Isolation Forest variant proposed in this thesis, $i\text{Forest}_{CS}$, is compared to the three most common encoding strategies discussed in Subsection 2.5.1. These are label encoding, frequency encoding, and one-hot encoding, which are explained in more detail in Subsection 2.5.1. For $i\text{Forest}_{CS}$, both a binary and a ternary implementation are evaluated.

The first distinct parameter that is changed is the percentage of nominal attributes present in the overall data set. This parameter directly impacts the total number of features that are nominal and thus sampled from the Dirichlet-Multinomial distribution. Figure 4.1 visualises the AUC and AUPRC results of the different Isolation Forest implementations. The number of observations ($n = 10000$), the cardinality of a nominal attribute ($k = 50$), and the total number of features ($d = 50$) are all kept constant. It is chosen to use these constant parameters, as these reflect the customer transaction data most accurately.

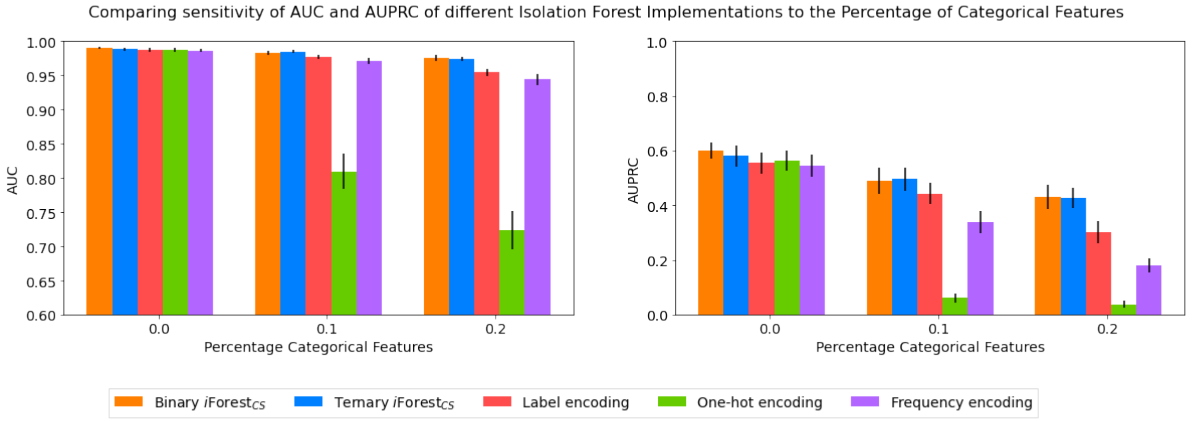


Figure 4.1: Sensitivity analysis of AUC and AUPRC when altering the percentage of nominal features. As can be seen, increasing the percentage of nominal features present in the data set has a deteriorating effect on the overall AUC and AUPRC results. However, the $i\text{Forest}_{CS}$ results deteriorate to a lesser extent than those of the encoding strategies. Particularly the performance of one-hot encoding becomes substantially worse in comparison with an increase in the percentage of nominal features in the data. The following characteristics of the data set are kept constant: the number of observations ($n = 10000$), the cardinality of a nominal attribute ($k = 50$), and the total number of features ($d = 50$). All results represent the average of 20 runs with error bars representing the standard deviation.

From Figure 4.1 it is evident that the results of the encoding strategies deteriorate more as the number of nominal features increases. This is as expected, considering the encoding of nominal attributes to a continuous spectrum introduces information that is not necessarily present. One-hot encoding in particular completely deteriorates with the increase of nominal attributes.

The second parameter that is altered is the total number of features of the data set. Figure 4.2 visualises the AUC and AUPRC results of the different Isolation Forest implementations with changes in the total number of features. The number of observations ($n = 10000$), the cardinality of a nominal attribute ($k = 50$), and the percentage of nominal attributes (10%) are all kept constant.

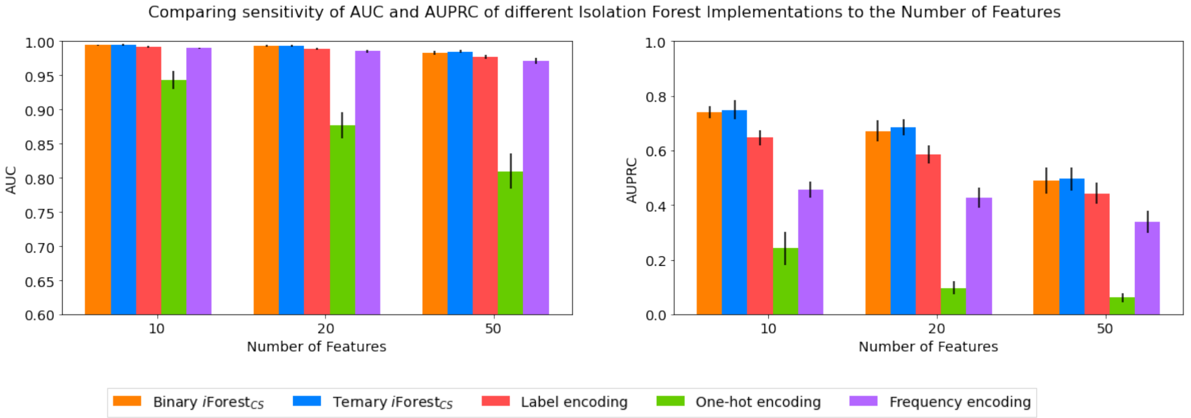


Figure 4.2: Sensitivity analysis of AUC and AUPRC when altering the total number of features of the data set. As the number of feature increases, the performance of Isolation Forest worsens in general. The following characteristics of the data set are kept constant: the number of observations ($n = 10000$), the cardinality of a nominal attribute ($k = 50$), and the percentage of nominal attributes (10%). All results represent the average of 20 runs with error bars representing the standard deviation.

From Figure 4.2 a consistent deterioration of results is visible across all implementations. As the dimensionality increases, the ability of an Isolation Forest to detect anomalies decreases. The overall patterns in the performance of the implementations remains constant, however. It is clear, that the iForest$_{CS}$ implementations consistently yields better results than when the nominal attributes are encoded. Overall, the worst results stem from one-hot encoding, whereas the performance the of the label encoding tends to yield better results than the frequency encoding.

In the Isolation Forest procedure, at every node a feature is chosen at random. When the dimensionality increases, it is intuitive that the probability to select a particular feature along an observation's path decreases. For an anomaly that is only detectable in this feature, or more generally in a sub-space of particular features, the increase in dimensionality yields a lower probability of detection. To formalize this, consider the feature set $\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_d\}$ from which we sample with replacement. Let A_i denote the event that feature Q_i is included in the path length of a particular anomaly. Let p represent the anomaly path of this anomaly. Then the probability that a given feature i is not sampled throughout the path length is:

$$\mathbb{P}(A_i^c) = \left(\frac{d-1}{d}\right)^p,$$

and therefore the probability that feature i is sampled during the anomaly's path length is:

$$\mathbb{P}(A_i) = 1 - \left(\frac{d-1}{d}\right)^p.$$

Whenever one-hot encoding is used, the original feature is expanded to a matrix of size k, where k represents the cardinality of the nominal feature. Therefore, the total dimensionality after one-hot encoding increases up to:

$$d \to d + \sum_{a=1}^{d_c} (k_a - 1),$$

where d_c represents the number of nominal features in the data and k_a the cardinality of the a^{th} categorical feature. This increase in dimensionality thus decreases the probability of an anomaly's path containing a particular feature i. This is the essence as to why the results of the one-hot encoding are considerably worse than that of the other methodologies.

Finally, the last parameter to the synthetic data that is altered, is that of the cardinality of the nominal attributes. It is interesting to acknowledge that the unique categories in a nominal attribute may impact the performance of particularly the encoding strategies. Nevertheless, it is expected that the increase in cardinality will also impact the performance of iForest$_{CS}$. The sampling over the unique categories will become increasingly less meaningful when the probability to fully isolate a distinct category decreases. The following characteristics of the data set are kept constant: the number of observations (n = 10000), the total number of features (d = 50), and the percentage of nominal attributes (10%).

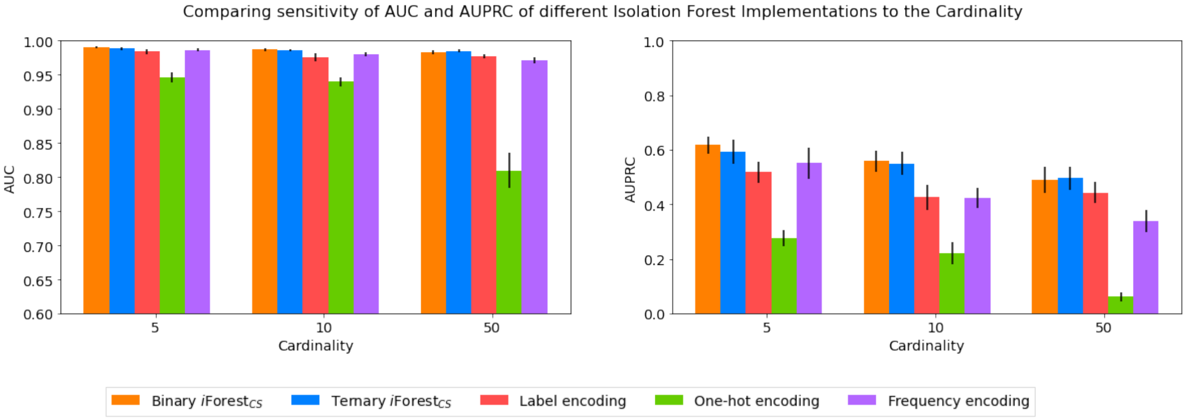


Figure 4.3: The following characteristics of the data set are kept constant: the number of observations (n = 10000), the total number of features (d = 50), and the percentage of nominal attributes (10%). All results represent the average of 20 runs with error bars representing the standard deviation.

In Figure 4.3, it is once again clear that one-hot encoding deteriorates due to the increase in dimensionality. Furthermore, what is observed throughout the different experiments, is that with an increase in cardinality the frequency encoding strategy performs worse than the label encoding. As Figure 4.1 and Figure 4.2 assume a constant cardinality of k = 50, this is also reflected in these figures. With lower cardinality this significant difference between frequency and label encoding is not present.

From the experiments conducted in this section, the sensitivity of the different Isolation Forest implementations have been tested. A clear trend is emphasized that iForest$_{CS}$ outperforms the encoding strategies throughout this sensitivity experiment. The parameter settings that most resemble the real-life customer transaction data used in Chapter 5 are: 50 features, a cardinality of 50, and 10% nominal attributes. This is the right-most experiment in Figure 4.3, for example. With these parameters, there was no statistically significant difference between the binary and the ternary iForest$_{CS}$ implementations (p-value AUC = 0.159, p-value AUPRC = 0.536). Between the label encoding and the binary iForest$_{CS}$, however, the difference between the means is statistically significant (p-value AUC = 1.41e-10, p-value AUPRC = 8.37e-12).

Thus, it is concluded that with data sampled from independent features, the iForest$_{CS}$ method provides improved anomaly detection results when compared to several encoding strategies. When considering the data characteristics of the customer transaction monitoring use-case, the binary and ternary iForest$_{CS}$ method did not demonstrate significant differences in the AUC and AUPRC results.

4.2.4. Runtime Evaluation

In this section, the runtime of different Isolation Forest adaptations for mixed-attribute data are compared. The experiments that are conducted will evaluate the execution time of an algorithm up until the calculation of the anomaly scores. First, the *scikit-learn* implementation of Isolation Forest after the nominal attributes have been encoded to a numerical representation is considered. Second, the implementation of iForest$_{CS}$ is evaluated. This is done for both the binary and ternary implementations. Finally, the iForest$_{CAD}$ methodology is evaluated. Here, evaluation is performed up until the conditional anomaly detection stage, under the assumption that all nominal attributes contribute to the detection of anomalies. Therefore, the partitioning is performed over all values of the nominal attributes in the data set. The *scikit-learn* implementation is then utilised to determine the anomaly scores specific to a particular data partition.

To measure the runtime of these different adaptations of Isolation Forest, different data sets are generated to evaluate the runtime sensitivity to particular characteristics of the data. The characteristics that will be altered are:

1. **Total number of features:** Three different values for the number of features are examined: {10, 20, 50}.

2. **Percentage of nominal features:** Considering the low percentage of nominal features in real-life data sets, three different percentages are evaluated: {0, 10, 20}. The case in which no nominal attributes are present in the data functions as a benchmark comparison between the *scikit-learn* implementation of Isolation Forest, and our own. Isolation Forest was built from scratch in this thesis to accommodate experiments regarding the incorporation of nominal attributes and the extension of binary isolation trees to ternary. The total number of nominal attributes is thus dependent on the total number of features and the percentage nominal features.
3. **Cardinality of a nominal attribute:** As a nominal attribute can take on distinct values, it is interesting to evaluate the effect on the algorithms' execution time when varying a feature's cardinality. The values considered are: {2, 5, 10}.

Furthermore, the experiments are run keeping the following parameters of an isolation tree constant. In every experiment $T = 100$ isolation trees compose a forest, and the sub-sampling size is taken to be $\psi = n$, where n represents the overall number of observations. The results of the runtime experiments are shown in Table 4.1.

		Nominal Features = 0%	Nominal Features = 10%			Nominal Features = 20%		
Number of features	**Implementation**		**Cardinality**			**Cardinality**		
			2	5	10	2	5	10
10	Encoding + sklearn	0.72	0.77	0.70	0.72	0.72	0.74	0.73
	$i\text{Forest}_{CS}$	13.09	9.95	16.43	19.64	6.77	17.01	23.37
	$i\text{Forest}_{CS}$ Ternary	17.24	14.18	20.59	22.52	11.87	20.23	26.95
	$i\text{Forest}_{CAD}$	NA	0.9	1.12	1.81	1.51	6.02	18.04
20	Encoding + sklearn	0.84	0.83	0.86	0.87	0.87	0.87	0.9
	$i\text{Forest}_{CS}$	12.70	14.06	20.44	22.76	12.12	23.46	28.37
	$i\text{Forest}_{CS}$ Ternary	15.83	18.24	23.62	25.82	16.21	26.74	31.70
	$i\text{Forest}_{CAD}$	NA	1.60	6.00	18.33	6.17	156.47	NA
50	Encoding + sklearn	1.32	1.36	1.36	1.35	1.42	1.40	1.39
	$i\text{Forest}_{CS}$	13.30	28.67	33.96	34.89	26.40	35.69	38.99
	$i\text{Forest}_{CS}$ Ternary	16.46	28.00	33.03	33.54	26.05	35.57	39.74
	$i\text{Forest}_{CAD}$	NA	19.06	933.22	NA	573.45	NA	NA

Table 4.1: The execution time in seconds of four different adaptations to Isolation Forest, when considering a data set of 10000 observations. The average of 5 runs is considered, with the exception for that of $i\text{Forest}_{CAD}$ where only 1 run is performed. Furthermore, the execution of $i\text{Forest}_{CAD}$ is aborted when the Cartesian product of the nominal attributes becomes larger than the number of observations or when there are no nominal features present in the data. Without nominal features, $i\text{Forest}_{CAD}$ reduces down to the *scikit-learn* implementation of Isolation Forest. These two situations are indicated with NA.

The first thing that should be clarified, is that purposefully some experiments are not conducted using the $i\text{Forest}_{CAD}$ implementation. Whenever there are no nominal features in the data (Nominal Features = 0%), the $i\text{Forest}_{CAD}$ implementation reduces down to a *scikit-learn* implementation of Isolation Forest. Furthermore, experiments are not conducted whenever the Cartesian product of the nominal features is greater than or equal to the total number of observations. Thus, whenever these two situations occur, the runtime is indicated with “NA".

What becomes immediately clear from Table 4.1, is the difference between the efficiency of the *scikit-learn* implementation and the implementation of Isolation Forest in this thesis. Whenever there are no nominal attributes in the data, the implementation of *scikit-learn* is between 10 – 20 times faster than the binary implementation in this thesis. Optimisation of the runtime of this thesis's implementation of Isolation Forest, however, was beyond the scope of this thesis. Thus, although some iterations were conducted to improve overall runtime, the efficiency was not the main focus of the research, making it insufficient to compare the algorithms is an absolute sense. Furthermore, it is important to note that the average height of the tree in *scikit-learn* is set as $log_2(n)$, as proposed in the original paper [2]. It is argued, however, that the average height of the tree is set to that of the average path length of a proper binary tree. Considering an isolation tree is a proper binary tree, the average path length of an observation will be equal to $2\big(ln(n)+\gamma-1\big)$, as derived in Subsection 2.3.2. This entails that the Isolation Forest implementation in this thesis is grown deeper than the *scikit-learn* implementation, thus requiring an extended runtime. The impact on the overall performance of an Isolation Forest with the newly proposed height limits is discussed in Subsection 4.5.1.

The shaded rows in Table 4.1 represent the runtimes for iForest$_{CS}$, which will be discussed in more detail. When considering the situation for which there are no nominal features present in the data, there is no significant difference in runtime with an increase in dimensionality. This is expected, as the complexity of the Isolation Forest is independent of the dimensionality. When introducing nominal features, however, the dimensionality increase results in an increased runtime. This gives the indication that the implementation of iForest$_{CS}$ is dependent on the number of features. Naturally, it is assumed that this must result from the increased number of nominal features. However, when considering the percentage of nominal features in the data with constant dimensionality, the runtime increase is not as significant. This indicates that the implementation of iForest$_{CS}$ can be improved, particularly in the nominal branching strategy. It appears that within this nominal branching strategy the dimensionality of the data is impacting the complexity, whereas this in not the case when considering no nominal attributes.

Furthermore, it is observed that an increase in the cardinality of a nominal feature results in an increase in runtime. As the nominal splitting strategy has to sort through more unique categories, the nominal branching strategy requires a longer runtime.

When comparing the binary and ternary variations of iForest$_{CS}$, it is clear that the implementations demonstrate comparable sensitivity to the characteristics of the data set. When no nominal features are present in the data, the runtime of the algorithm is independent of the total number of features in the data set. The ternary implementation always requires slightly more time to execute, which results from the additional splitting test performed at every node in the isolation tree. This increase in computation is balanced by the fact that the average path length of a ternary tree is shorter than that of a binary isolation tree (see Subsection 3.2.2), resulting in an earlier termination of a ternary isolation tree. After introducing nominal attributes, the runtime of the algorithm increases.

Finally, for the iForest$_{CAD}$ it is evident that as the Cartesian product of the nominal attributes increases, the runtime explodes. This is not difficult to imagine, after considering that Isolation Forest is performed over every data partition resulting from the Cartesian product. Therefore, with an increase in both the percentage of nominal features and cardinality, the runtime complexity of iForest$_{CAD}$ increases as well. This observation was already indicated in the paper proposing iForest$_{CAD}$ [36], yet with limited nominal attributes and low corresponding cardinality, the results of the methodology are promising.

4.3. Experiments with Conditional Features

In this section, the analysis in Section 4.2 is extended to consider data with dependencies between particular features. In particular, this section contains an intuitive data set in which population height and weight data is conditioned on the feature containing country information. This section is structured in a similar fashion to the experiments section with synthetic data with independent features. First, the purpose of these experiments are addressed in Subsection 4.3.1. Next, the synthetically generated data set is explained and visualised in Subsection 4.3.2. Subsection 4.3.3 then visualizes the improvement in anomaly ranking with the incorporation of nominal attributes.

4.3.1. Purpose of Experiments

In this section, the analysis is extended to incorporate a data set in which there are dependent features. This is primarily done to create an intuitive data set which is visually comprehensible, as well as useful when determining the performance of local explanation methods in a mixed-attribute setting. The benefits of incorporating categorical data is visualised using a simple version of the synthetic data set.

4.3.2. Synthetic Data with Dependent Features

This subsection will consider data sets containing intuitive, dependent attributes to illustrate the proposed approach in Section 3.1 and compare it to the encodings of categorical data. In order to achieve this goal, similar analysis is performed as in [36] using information regarding height and weight. Where originally the height and weight of men and women in the US was used to showcase the performance of the $i\text{Forest}_{CAD}$ approach, the analysis will now be extended to incorporate more countries. Nevertheless, the bivariate distribution parameters are inferred from the large population survey conducted in [47], aimed at modelling the relationship between height and weight of men and women in the United States. The bivariate distribution parameters are utilised in combination with more up-to-date country statistics obtained from Worlddata.info [48].

Using the bivariate distributions and the up-to-date country statistics, a data set is constructed containing three attributes, namely country (nominal), height in centimeters (numerical), and weight in kilograms (numerical). In this section, the experiments that are demonstrated consider only these three attributes under the assumption that they all contribute to the detection of anomalies. Furthermore, for illustrative purposes, the experiment is initially conducted using only three unique countries, namely the United States, the Netherlands, and Vanuatu. Table 4.2 summarises the information regarding the average height and weight of the male population in these specific countries.

Country	**Height (cm)**	**Weight (kg)**
United States	177	90.6
The Netherlands	184	87.9
Vanuatu	168	72.6

Table 4.2: The average height (in centimeters) and weight (in kilograms) of the male population of different countries. This information is obtained from [48].

The information contained in Table 4.2 is used to construct the bivariate normal distributions (Definition 4.3.1) used for experimentation purposes.

Definition 4.3.1. *Two random variables X and Y have a* ***bivariate normal distribution*** *with parameters μ_X, σ_X^2, μ_Y, σ_Y^2, and correlation coefficient ρ_{XY}, if their joint probability density function is given by:*

$$f_{XY}(x,y) = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho^2}} \cdot e^{-\frac{1}{2(1-\rho^2)}\left[\left(\frac{x-\mu_x}{\sigma_X}\right)^2+\left(\frac{x-\mu_X}{\sigma_X}\right)^2-2\rho\frac{(x-\mu_X)(y-\mu_Y)}{\sigma_X\sigma_Y}\right]}, \tag{4.1}$$

where μ_X, $\mu Y \in \mathbb{R}$, $\sigma_X, \sigma_Y > 0$ and $\rho_{XY} \in (-1,1)$.

Let $X_i \sim N(\mu_{X_i},\sigma_X)$ and $Y_i \sim N(\mu_{Y_i},\sigma_Y)$ represent two jointly normal random variables with correlation $\rho(X,Y) = \rho$, where X and Y represent the height and weight of males in a given country. The parameters ρ,σ_X,σ_Y are inferred from the large population survey conducted in [47], after undergoing conversions to SI units. For a visual illustration, 300 observations are sampled from the bivariate normal distribution conditioned on three countries, where μ_{X_i} and μ_{Y_i} are found in Table 4.2. The frequency of the observations sampled from the United States, the Netherlands, and Vanuatu is constant. The resulting data, alongside identifying the 5% of observations with the lowest joint probability density as anomalies, are visualised in Figure 4.4.

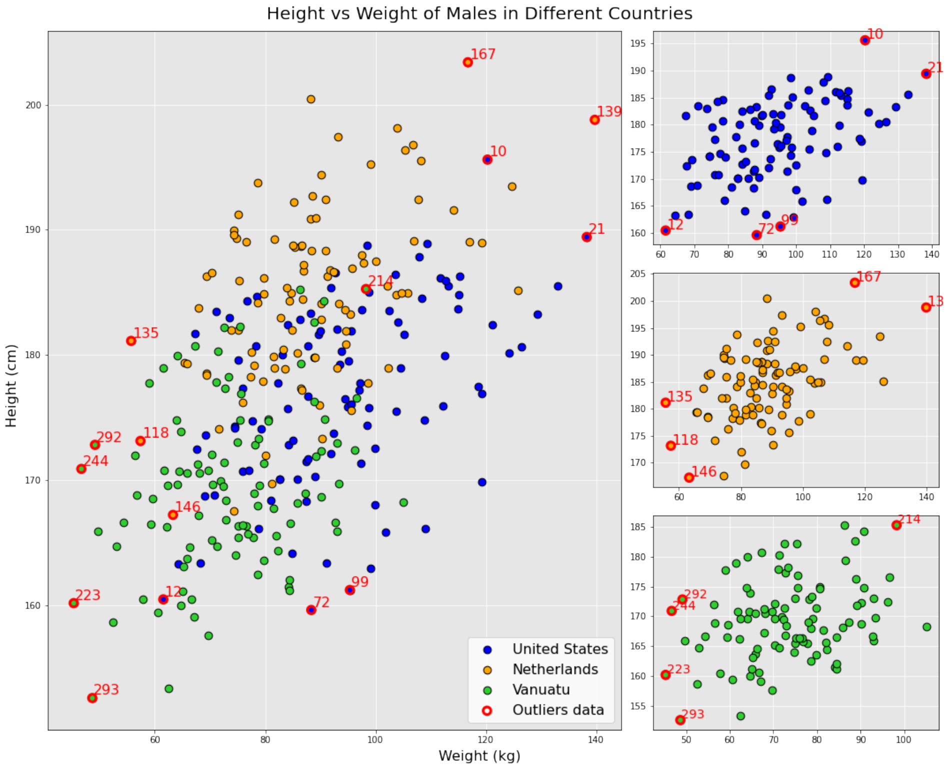


Figure 4.4: Random sampling from a bivariate normal distribution for three different countries to obtain a test set of the height and weight of the male population for three different countries. The data set contains $n = 300$ observations, such that there are 100 observations sampled from to the distribution corresponding to every country. The large left plot depicts the combined result, whereas the three right figures illustrate the distribution per country. The colours corresponding to the individual countries are stated in the legend. The 5% of observations with the lowest joint probability density are considered anomalies and are depicted with an additional thick, red edge.

As illustrated in Figure 4.4, there is overlap in the three bivariate normal distributions. If there was no distinction between the different countries, certain observations which are defined as anomalies become hidden. Take, for example, observation 214. When purely considering the bivariate distribution with the parameters corresponding to those of Vanuatu, observation 214 is more susceptible to isolation as shown in the bottom right plot of Figure 4.4. However, when visualising the data entirely, observation 214 becomes concealed by overlapping data of other countries. In this particular data set, four clear hidden anomalies can be identified, namely observations 12, 118, 146, and 214.

Now that the data set with dependent features have been introduced, the following section will describe the effect of incorporating nominal attributes into the anomaly detection algorithms.

4.3.3. Incorporating Nominal Attributes

When an Isolation Forest is trained after discarding the categorical attributes, it becomes evident that the hidden anomalies in Figure 4.4 become undetectable. Specifically, the observations in the extreme regions of the entire data set are most susceptible to isolation. This is illustrated in Figure 4.5, in which the contour maps of the Isolation Forest's anomaly scores are visualised when discarding nominal attributes.
Figure 4.5 demonstrates the loss of information that results from discarding nominal attributes altogether. Although the global anomalies remain detected, the hidden anomalies are concealed without incorporating country information. When considering the contour mappings, it is evident that only the observations located at the extremes of the overall distribution receive high isolation scores. Observation 214, which is emphasized in Figure 4.4 as an anomaly with respect to the data's joint probability distributions, remains hidden and receives a low anomaly score. This holds for all anomalies that are concealed by overlapping bivariate normal distributions.

To improve the performance of Isolation Forest in detecting anomalies, information contained in nominal

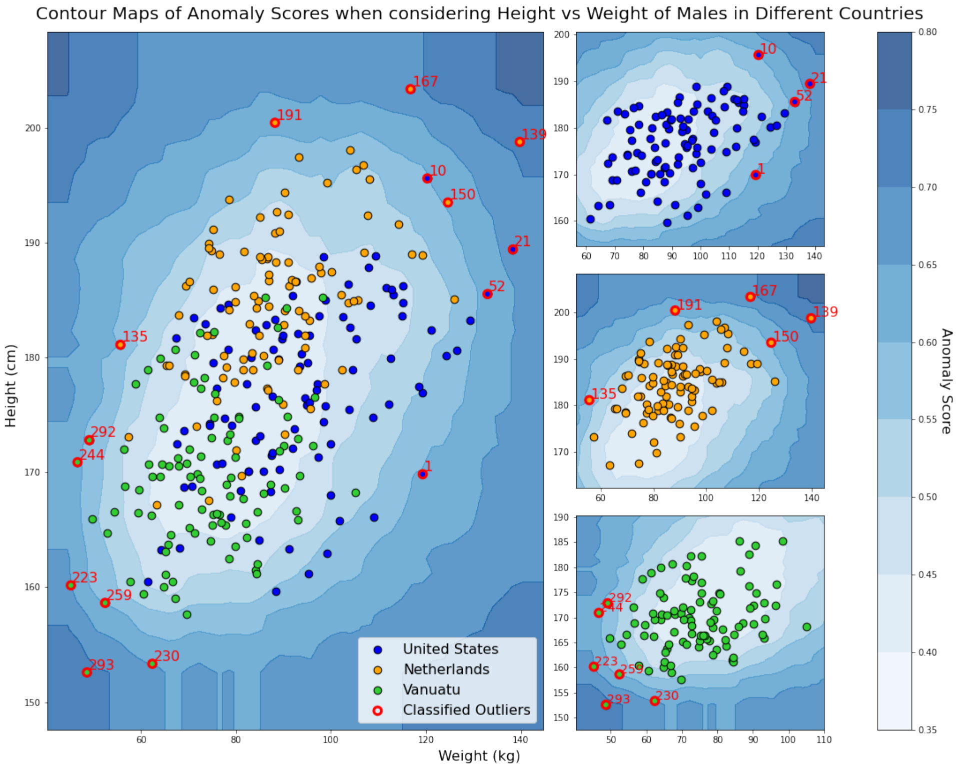


Figure 4.5: The contour maps of the resulting anomaly scores of an Isolation Forest trained on the same data as Figure 4.4, without incorporating the categorised country data. The contour map over all four plots are identical, yet the three smaller plots only consider a sample of the data specific to a country. The observations with a thick, red boundary and an annotation, which represents the index of the observation in the data set, represent the 5% of observations with the largest anomaly scores. It is thus important to note that these indicate the top anomalies identified by the Isolation Forest, and not the anomalies of the overall data set.

attributes is included. This is done using different approaches. First, the $i\text{Forest}_{CS}$ methodology from Section 3.1 is implemented. The results from this implementation are also visualised in Figure 4.6. From this figure, it is evident that incorporating nominal attributes allows for the identification of the hidden anomalies in the data set. Observations with index 12, 118, 146, and 214 are all identified and thus readily isolated.

Furthermore, the $i\text{Forest}_{CAD}$ [36] method is implemented up until the classified training stage. Through selection of the nominal attribute and the numerical attributes, a partitioning of the data is performed through computation of the nominal feature's Cartesian product. For every data partition, Isolation Forest is used to determine conditional anomaly scores. These scores are then combined and used to identify the top anomalies.

Finally, the data is encoded through the use of label encoding, as defined in Subsection 2.5.1. Through the utilization of Python's Scikit-Learn package, label encoding practically entails countries being assigned an integer value according to an alphabetical ordering. For this demonstration, only label encoding is utilized as a comparison encoding strategy. Due to the similar frequencies of the sampled country data, the performance of frequency encoding drops significantly as all countries will be encoded to the same value. Table 4.3 depicts the anomaly rankings of all probabilistic anomalies resulting from the Isolation Forest implementations.

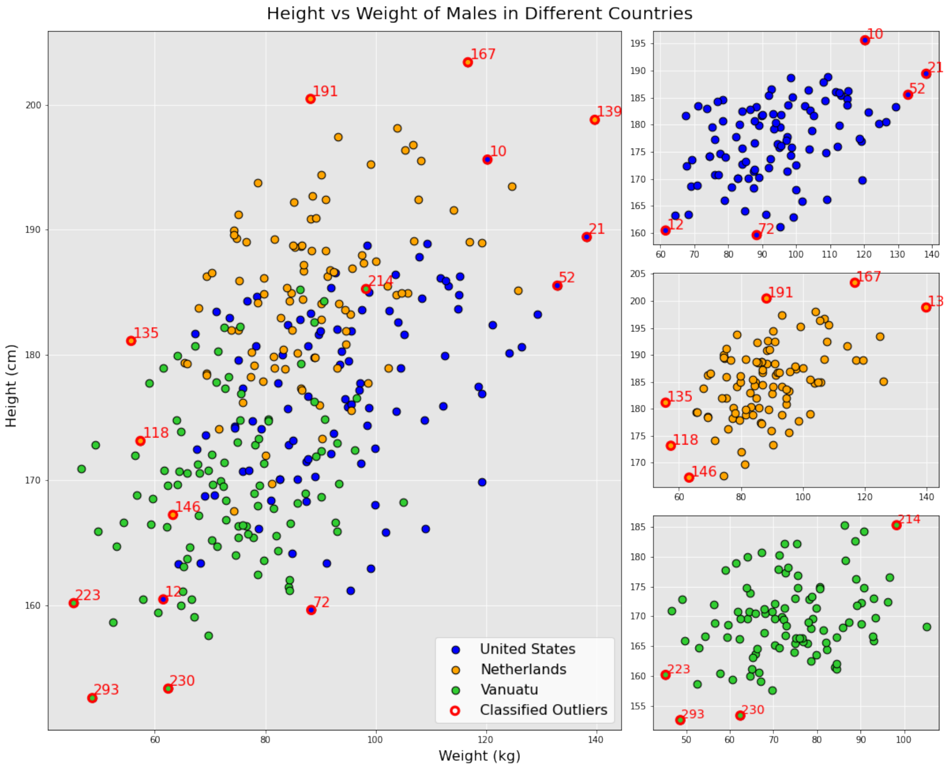


Figure 4.6: When incorporating the categorised country data and utilising iForest$_{CS}$ as explained in Algorithm 4, the top 5% of anomalies are identified and illustrated as the observations with the thick, red boundaries. Notice the hidden anomalies (12, 118, 146, and 214) being isolated through the utilization of the country data. Finally, note that this is the visualization of one particular Isolation Forest, whereas later results reflect multiple Isolation Forest runs.

		Implementation					
		iForest	iForest$_{LE}$	iForest$_{CS}$		iForest$_{CAD}$	
i	**Country**			**Binary**	**Ternary**		**Hidden Anomalies**
244	VA	(8)	(17)	(17)	(19)	(23)	
292	VA	(13)	(23)	(24)	(24)	(29)	
139	NL	(1)	(1)	(1)	(1)	(1)	
135	NL	(14)	(11)	(15)	(14)	(13)	
223	VA	(6)	(8)	(6)	(6)	(12)	
167	NL	(3)	(3)	(3)	(3)	(6)	
12	US	(55)	(10)	(7)	(8)	(5)	Yes
293	VA	(2)	(2)	(2)	(2)	(3)	
10	US	(16)	(5)	(5)	(5)	(4)	
21	US	(4)	(4)	(4)	(4)	(2)	
72	US	(21)	(21)	(16)	(20)	(16)	
118	NL	(49)	(9)	(12)	(10)	(10)	Yes
99	US	(25)	(30)	(25)	(26)	(26)	
146	NL	(103)	(7)	(11)	(9)	(8)	Yes
214	VA	(214)	(13)	(20)	(12)	(7)	Yes

Table 4.3: The ranks of the anomaly scores assigned to the observations with the lowest joint probability densities after conducting anomaly detection through numerous Isolation Forest adaptations. The implementation in which no nominal attributes are considered, iForest, is compared to four methods in which nominal data is considered. These are Isolation Forest with label encoding, iForest$_{LE}$, both binary and ternary implementations of Isolation Forest with Categorical Sampling iForest$_{CS}$, and iForestCAD [36]. Observe how the inclusion of the nominal attribute strongly improves the detection of the hidden anomalies.

As seen in Table 4.3, the ability to detect hidden anomalies strongly improves through the utilization of the nominal attribute involving country data. Whereas omitting the country data results in the failure to identify these anomalies completely, the other methodologies consistently detect the particular anomalies. Only the binary implementation of $i\text{Forest}_{CS}$ would fail to detect observation 214 after considering the results after 20 runs. It is, however, still a significant improvement when compared to omitting the data altogether.

It is important to critically evaluate this experiment and the results that can be drawn from the rankings. The number of countries incorporated into the countries attribute is limited, with only three unique categories to sample from. However, even with this simple data set, the intuition behind sampling a nominal attribute rather than encoding it to a numerical scale can be justified. As mentioned before, when utilizing the pre-processing procedure of encoding the country data using label encoding, the countries are sorted in alphabetical order. When this ordering is altered, it becomes apparent how dependent label encoding is to the underlying order in the encoded numerical space.

		Implementation		
i	**Country**	$i\text{Forest}_{LE_1}$	$i\text{Forest}_{LE_2}$	$i\text{Forest}_{LE_3}$
12	US	(10)	(6)	(6)
118	NL	(9)	**(19)**	(12)
146	NL	(7)	**(21)**	(10)
214	VA	(13)	(11)	**(24)**

Table 4.4: When purely considering the hidden anomalies, it is evident that the rank ordering of Isolation Forest is sensitive to the order introduced by the Label Encoder. For $i\text{Forest}_{LE_1}$, the introduced ordering is as follows: The Netherlands, United States, Vanuatu. For $i\text{Forest}_{LE_2}$, the introduced ordering is as follows: Vanuatu, The Netherlands, United States. For $i\text{Forest}_{LE_3}$, the introduced ordering is as follows: The United States, Vanuatu, the Netherlands. Notice the differences in the anomaly detection results. When considering the top 5% of anomalies, which corresponds to the top 15 anomaly scores, the bolded observations would not be considered an anomaly. These bolded observations all tend to belong to observations from a country that are centrally located in the encoding ordering, demonstrating a bias towards the extreme values of an encoding.

Lastly, it is important to note that the $i\text{Forest}CAD$ demonstrates great results with such data sets. This results from the limited number of countries incorporated into the nominal attribute, the presence of only one nominal attribute in the data, and that all partitions resulting from the Cartesian product contain plenty of observations. When applying Isolation Forest to real-life customer transaction data in Chapter 5, these conditions cannot be ensured. Furthermore, recalling Table 4.1, it is argued that with increased cardinality and nominal attributes the complexity of $i\text{Forest}CAD$ cannot be justified in a practical application. Therefore, it is argued that $i\text{Forest}CS$ can result in the improved detection of anomalies when more nominal attributes with high cardinality are incorporated into the data, as is the case in Chapter 5.

4.4. Experiments with a Ternary Isolation Forest

This section will elaborate on MI-Local-DIFFI when applied to a ternary Isolation Forest. Contrary to the remainder of the thesis, this is done with purely numerical data. It is opted to first analyse the results of MI-Local DIFFI when applied to numerical data, as there has not been an evaluation involving the results of a local explanation method applied to ternary Isolation Forest up to date.

4.4.1. HiCS data sets

To examine the results of a ternary Isolation Forest when subjected to local explainability, a similar approach is considered as in [4]. It is important to access data that allows for meaningful experiments with particularly local explanations in mind, and therefore the synthetic data proposed in the HiCS paper is used [49]. HiCS is a pre-processing step to anomaly detection algorithms, in which high contrast subspaces are searched that have a conditional dependence among the subspace dimensions. To validate this approach, the authors produce synthetic data sets containing anomalies in particular subspaces. Due to ground-truth knowledge of the features spanning the subspaces containing anomalies, this data set can be used to examine the performance of local explanation methods.

Within the data sets that are constructed for HiCS, $2-5$ dimensional subspaces are randomly selected. In these subspaces, high-density clusters are generated in which several observations are altered to become anomalies. These anomalies are non-trivial, as they deviate from the high-density clusters but remain undetected in a lower-dimensional representation of the subspace. Originally, there were 21 data sets in total, all containing 1000 observations but varying in feature size. However, after the experiments conducted in [4], it was determined that Isolation Forest subjected to data set number $\{1,2,3,5,6\}$ yielded AUC results of 0.75 or greater. Only these data sets were considered for the evaluation of the explanation performance, which is done now as well.

4.4.2. Ternary Isolation Forest Evaluation

Now that the data is explained, and the explanation performance metric is defined, the ternary Isolation Forest can be evaluated. Ternary Isolation Forest lacks efficient implementation in Python libraries, so in order to conduct these experiments the ternary Isolation Forest is implemented from scratch. This allows for the storage of all required node characteristics needed to construct a ternary variant of MI-Local-DIFFI. Furthermore, the indicators are slightly altered as stated in Subsection 3.3.1 to incorporate for ternary isolation trees. When utilising a binary and ternary isolation tree for the data sets, the performance comparison of MI-Local-DIFFI is visualised in Figure 4.7:

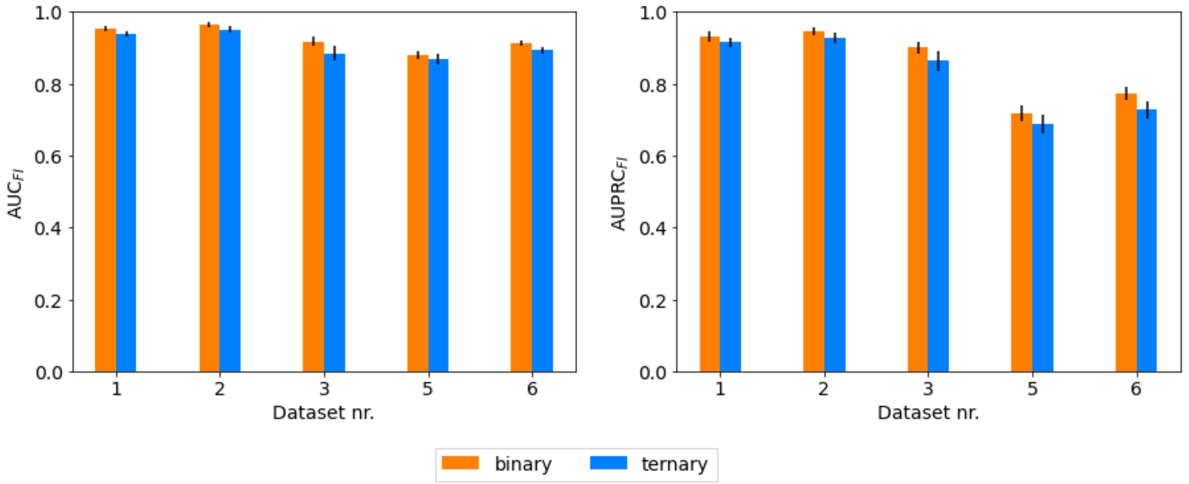


Figure 4.7: The results of the AUC_{FI} and $AUPRC_{FI}$ when using a binary and ternary Isolation Forest implementation. The experiments are conducted over five different synthetic data sets, all containing $n = 1000$ observations. data sets number $1-3$ contain 10 features, while data sets 5 and 6 contain 20 features. The results reflect the average AUC_{FI} and $AUPRC_{FI}$ of 50 Isolation Forests and MI-Local-DIFFI runs, with the error bars indicating the standard deviations of the results.

From Figure 4.7 it is evident that the ternary Isolation Forest produces consistently worse explanations in terms of both AUC_{FI} and $AUPRC_{FI}$ when compared to that of a binary isolation tree. From Appendix B, it is furthermore observed that the difference between the binary and ternary implementations are significant. Since the MI-Local-DIFFI explanation is model specific, the feature importance vectors are dependent on the overall predictions of the Isolation Forest. This may vary to the ground truth data of the HiCS data sets, depending on the accuracy of the overall model. In order to determine whether the model accuracy is worse for the ternary Isolation Forest, Figure 4.8 shows the computation of the AUC and AUPRC for the relevant HiCS data sets.

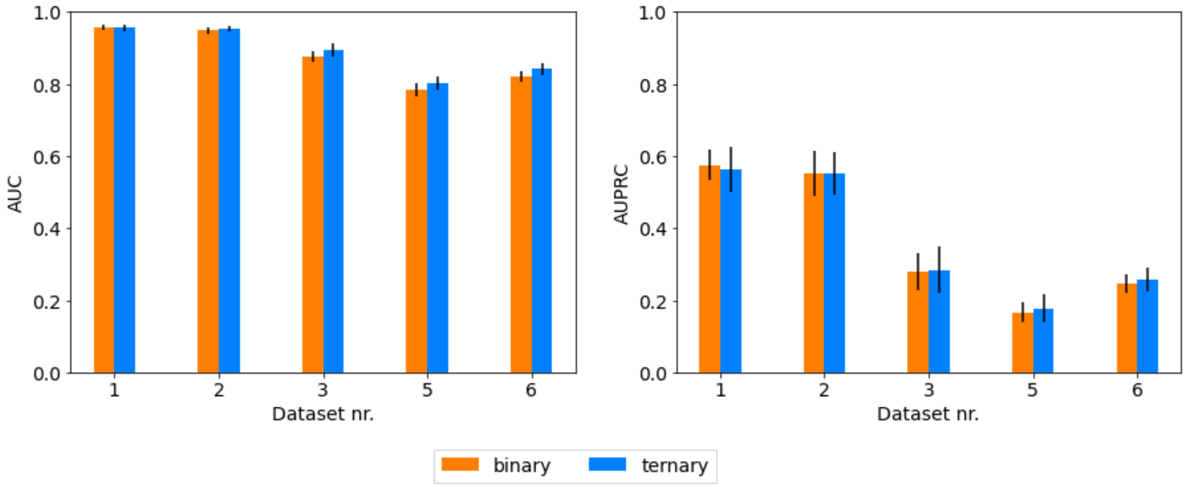


Figure 4.8: The computation of the AUC and AUPRC when using a binary and ternary Isolation Forest implementation. The experiments are conducted over five different synthetic data sets, all containing $n = 1000$ observations. data sets number $1-3$ contain 10 features, while data sets 5 and 6 contain 20 features. The results reflect the average AUC and AUPRC of 50 runs, whereas the error bars reflect the standard deviations of the computations.

It is apparent from Figure 4.8 that it is not necessarily the case that the binary implementation of Isolation Forest outperforms the ternary implementation. In fact, Appendix B indicates that there is no statistically significant difference between all AUPRC results and the AUC results of the first data set. Thus, the accuracy of the overall model does not necessarily have to be the reason that the ternary isolation tree produces consistently worse local explanations.

When deep-diving into the MI-Local-DIFFI algorithm, it is evident that the feature importance score is constructed from various weight indicators. These weight indicators address the path length of the anomaly's path, the split proportion in a particular node, and the split interval length in a particular node. Considering the high-density clusters of the HiCS span several features, the anomalies are only detected when certain combinations of features are selected. Thus, to detect an anomaly with Isolation Forest, all these features must be selected in the path of an isolation tree. When the dimensionality of a data set increases, however, the probability for a specific feature to occur in the path length of an anomaly reduces. When an anomaly is contained in a specific subspace of the data, the probability of selecting this combination in an unique path becomes even smaller.

Corresponding to the argumentation of the original Isolation Forest paper [2], a height limit is introduced to an isolation tree. Considering an anomaly is an observation prone to isolation, their corresponding path lengths are substantially shorter than that of a normal observation. This height limit marks the average path length, which is a function of the number of input samples. However, with this height limit in place, the probability of selecting the specific features necessary to isolate an anomaly reduces. With a ternary Isolation Forest, every node contains an extra split test and the observations are assigned to an additional child node. Subsequently, the additional fracturing of the data ensures that a ternary tree grows on average less deep than a binary tree. This was already established with the calculations performed in Subsection 3.2.2 of the average path length of a ternary isolation tree. With this significantly lower height limit, the probability of a ternary isolation tree to select a particular combination of features is substantially lower than for a binary isolation tree.

To quantify this, consider Theorem 4.4.1:

Theorem 4.4.1. *Let X be a data set with n observations and d features, such that $X \in \mathbb{R}^{n\times d}$. The probability that l specific features $q_1,\ldots,q_l \in \{1,\ldots,d\}$ all occur in a path of length p, $\mathbb{P}\left(\bigcap_{j=1}^{l} A_{i_j}\right)$, can be expressed as [4]:*

$$\mathbb{P}\left(\bigcap_{j=1}^{l} A_{i_j}\right) = 1 + \sum_{k=1}^{l}\left((-1)^k \binom{l}{k}\left(\frac{d-k}{d}\right)^p\right). \tag{4.2}$$

For an anomaly that is only detected through the combination of three features, for example, the probability that these specific features all occur in the path length p of anomaly o can be computed using Theorem 4.4.1. Thus, when considering the example in which a data set has $d = 10$ features and $n = 1000$ observations, and using the theoretically computed average path lengths for both the binary and ternary variants of Isolation Forest, the following probabilities are computed:

$$\textbf{Binary:} \quad \mathbb{P}\left(\bigcap_{j=1}^{3} A_{i_j}\right) = 0.3913$$

$$\textbf{Ternary:} \quad \mathbb{P}\left(\bigcap_{j=1}^{3} A_{i_j}\right) = 0.1704$$

This demonstrates that the probability of successfully selecting a specific subspace of the data in the path of an anomaly changes as the isolation tree structure is altered. With an increase in the splits in a particular node, and the corresponding decrease in average path lengths of the isolation tree, the probability of selecting a specific subspace of the data decreases.

Split Interval Indicator with n-ary Isolation Trees

A second reason of the possible deterioration of explanation results of the MI-Local-DIFFI when expanding the binary isolation tree to a ternary isolation tree, is due to the split interval indicator of the MI-Local-DIFFI. As the number of splitting criteria in a node increases, the defined split interval length will reduce in comparison to the overall feature interval length. As this split interval length reduces in size, less information is contained in the location of the split.

To generalize this, the expected split interval length of a binary and ternary isolation tree is expressed. Let $X_1, X2, \ldots, X_n$ be a random sample of size n from a continuous distribution with a cumulative distribution function F and a probability distribution function f. Utilising order statistics notation, define $X_{(1)}$ and $X_{(n)}$ as the minimum and maximum random variable from the sample $X_1, X_2, \ldots, X_n$, respectively. To compute the probability density function $f_{X_{(1)}}(x)$, start with the computation of the cumulative density function $F_{X_{(1)}}(x)$:

$$\begin{aligned} F_{X_{(1)}}(x) &= \mathbb{P}(X_1 \leq x) = 1 - \mathbb{P}(X_1 > x) \\ &= 1 - \mathbb{P}(X_1 > x, \ldots, X_n > x) \\ &= 1 - \mathbb{P}(X_1 > x)\mathbb{P}(X_2 > x) \ldots \mathbb{P}(X_n > x) \\ &= 1 - \big(\mathbb{P}(X_1 > x)\big)^n \\ &= 1 - \big(1 - F(x)\big)^n \end{aligned}$$

This result can be used to determine the probability density function of the minimum through differentiating $F_{X_{(1)}}(x)$:

$$f_{X_{(1)}}(x) = n\big(1 - F(x)\big)^{n-1} f(x).$$

Likewise, the probability density function of the maximum can be computed to be:

$$f_{X_{(n)}}(x) = n\big(F(x)\big)^{n-1} f(x).$$

This analysis can be extended to express all order statistics. Now, let's apply this result to the situation at hand. For an n-ary isolation tree, the $(n-1)$ splitting values are sampled from the uniform distribution over the feature interval $[a, b]$. For simplicity, let $X_1, X_2, \ldots, X_{n-1} \sim Unif(0,1)$ represent the $(n-1)$ splitting values chosen within the node. Then:

$$f_{X_{(1)}}(x) = (n-1)\big(1 - x\big)^{n-2} I_{(0,1)}(x),$$

which corresponds to the probability density function of a Beta distribution with the parameters $\alpha = 1$ and $\beta = n - 1$. Thus,

$$X_{(1)} \sim Beta(1, n-1),$$

and

$$\mathbb{E}[X_{(1)}] = \frac{1}{n}.$$

With similar calculations it can be concluded that $X_{(n)} \sim Beta(n-1,1)$, and thus the expected value of $X_{(n)}$ can be expressed as:

$$\mathbb{E}[X_{(n)}] = \frac{n-1}{n}.$$

Finally, using that if a random variable $Z \sim Unif(0,1)$, then $a + (b-a)Z \sim Unif(a,b)$, the results above can be generalized to a particular feature range $[a,b]$:

$$\mathbb{E}[X_{(1)}] = a + (b-a)\left(\frac{1}{n}\right) = \frac{1}{n}\big(b + (n-1)a\big)$$
$$\mathbb{E}[X_{(n)}] = a + (b-a)\left(\frac{n-1}{n}\right) = \frac{1}{n}\big((n-1)b + a\big)$$

By symmetry, it is interesting to notice that the expected interval between order statistics is equal to that of $\mathbb{E}[X_{(1)}] - a$ or $b - \mathbb{E}[X_{(n)}]$. Thus, the expected split interval of a n-ary isolation tree is a factor $\frac{1}{n}$ of the total feature interval. With increasing n, this feature interval reduces and the feature importance indicator of the MI-Local-DIFFI method provides less information.

To evaluate this, the HiCS data sets are used to compute the local explanation performance when purely considering the split interval indicator. The results are depicted in Figure 4.9.

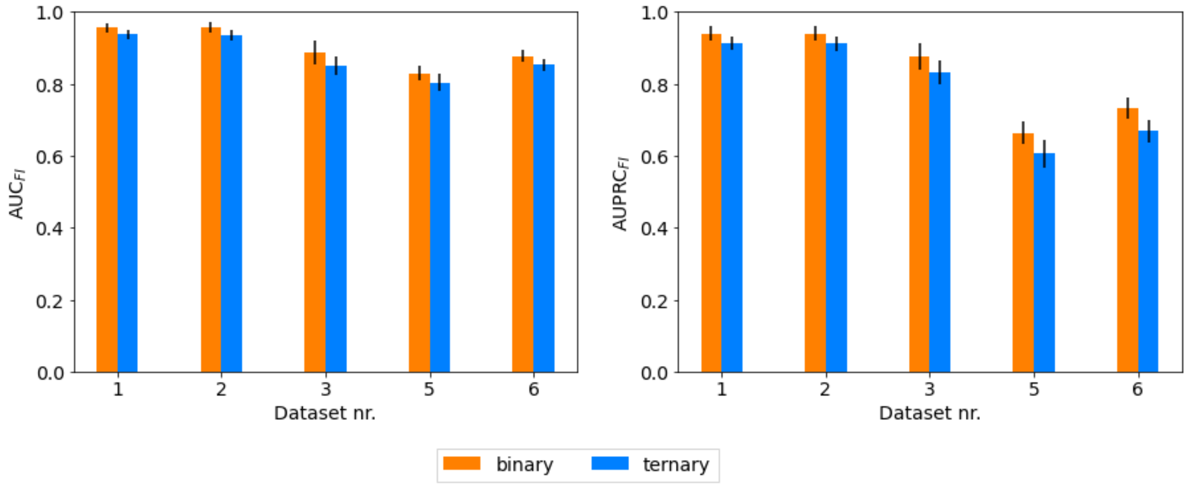


Figure 4.9: Performance of MI-Local-DIFFI for a binary and ternary Isolation Forest. Only the split interval indicator is considered in the local explanations, so the path length and split proportion indicators are omitted. The results reflect the average AUC_{FI} and $AUPRC_{FI}$ of 50 Isolation Forests and MI-Local-DIFFI runs, with the error bars indicating the standard deviations of the results.

Indeed, Figure 4.9 demonstrates that the ternary Isolation Forest does not identify the anomaly containing subspaces of the data as accurately and precisely as the binary Isolation Forest while only using the split interval indicator of MI-Local-DIFFI. Consistently, the average AUC and AUPRC results of 50 runs for the ternary Isolation Forest is lower than that of the binary implementation. In fact, Appendix B indicates that the difference is statistically significant. The standard deviation of these runs, however, are significantly larger than when comparing the full MI-Local-DIFFI method in Figure 4.7. This results from the fact that the anomalies are easily detectable in the features spanning the high density clusters. Therefore, since only the split interval indicator is considered, the feature importance scores are highly dependent on the stochasticity of the particular isolation tree and where the splits are made with respect to the feature interval.

It is therefore concluded that the ternary implementation of an Isolation Forest hinders the accuracy of detecting particular outlying subspaces when compared to the binary implementation. This is due to two reasons in particular;

1. The average path length of the ternary isolation tree is shorter than that of a binary isolation tree. This ensures that the algorithm's maximum height is set at a lower threshold, which reduces the probability of successfully selecting the attributes of the subspace containing an anomaly. In Subsection 4.5.1, the impact of the height limit parameter is examined in more details.
2. The split interval indicator of MI-Local-DIFFI loses significance, as the expected anomaly split interval reduces with an increasing n-ary isolation tree. In Section 6.3, the recommendation is taken into account to further explore new indicator possibilities that address this issue.

4.5. Isolation Forest Parameter Evaluation

Throughout this thesis, there have been multiple moments of critical evaluation of the underlying Isolation Forest algorithm as proposed in [2] and implemented in $scikit-learn$. Particularly with the underlying parameters expressed and utilized, it was found that some further evaluation is necessary. In Subsection 4.5.1, the maximum height limit of the Isolation Forest is investigated, and how the overall allowable depth of the Isolation Forest contributes to results. This is done by evaluating both the performance results when detecting anomalies and the performance of identifying the correct subspaces containing particular anomalies.

Next, Subsection 4.5.2 investigates the performance of an isolation tree with varying sub-sampling sizes ψ. This is also done by evaluating both the performance results when detecting anomalies and the performance of identifying the correct subspaces containing particular anomalies.

4.5.1. Isolation Forest Depth

In Subsection 2.3.2, the derivations of the expected path lengths of an isolation tree are elaborated on. In the original paper, using analysis of binary search trees and the average path length of an unsuccessful search, the average path of an isolation tree, $c(n)$, was computed to:

$$c(n) = 2H(n-1) - \frac{2(n-1)}{n},$$

where $H(n)$ represents the n^{th} harmonic number, for which the corresponding asymptotic expansion is: $H(n) \sim ln(n) + \gamma + \frac{1}{2n} - \frac{1}{12n^2} + \ldots$, where $\gamma \simeq 0.57722$. Furthermore, a height limit to the isolation tree, l, is introduced as:

$$l = ceiling(\log_2 \psi).$$

This height limit is introduced as it approximates the average tree height according to [50] and significantly reduces the complexity by preventing an isolation tree from growing until completion. However, when considering a proper binary tree, let N denote the maximum number of nodes in a binary tree, and h the height of the tree. Then:

$$\begin{aligned} N &= \sum_{k=0}^{h} 2^k, \\ &= 2^{h+1} - 1. \end{aligned}$$

Through rearranging and solving for the minimum height, the following expression is determined for the minimum height:

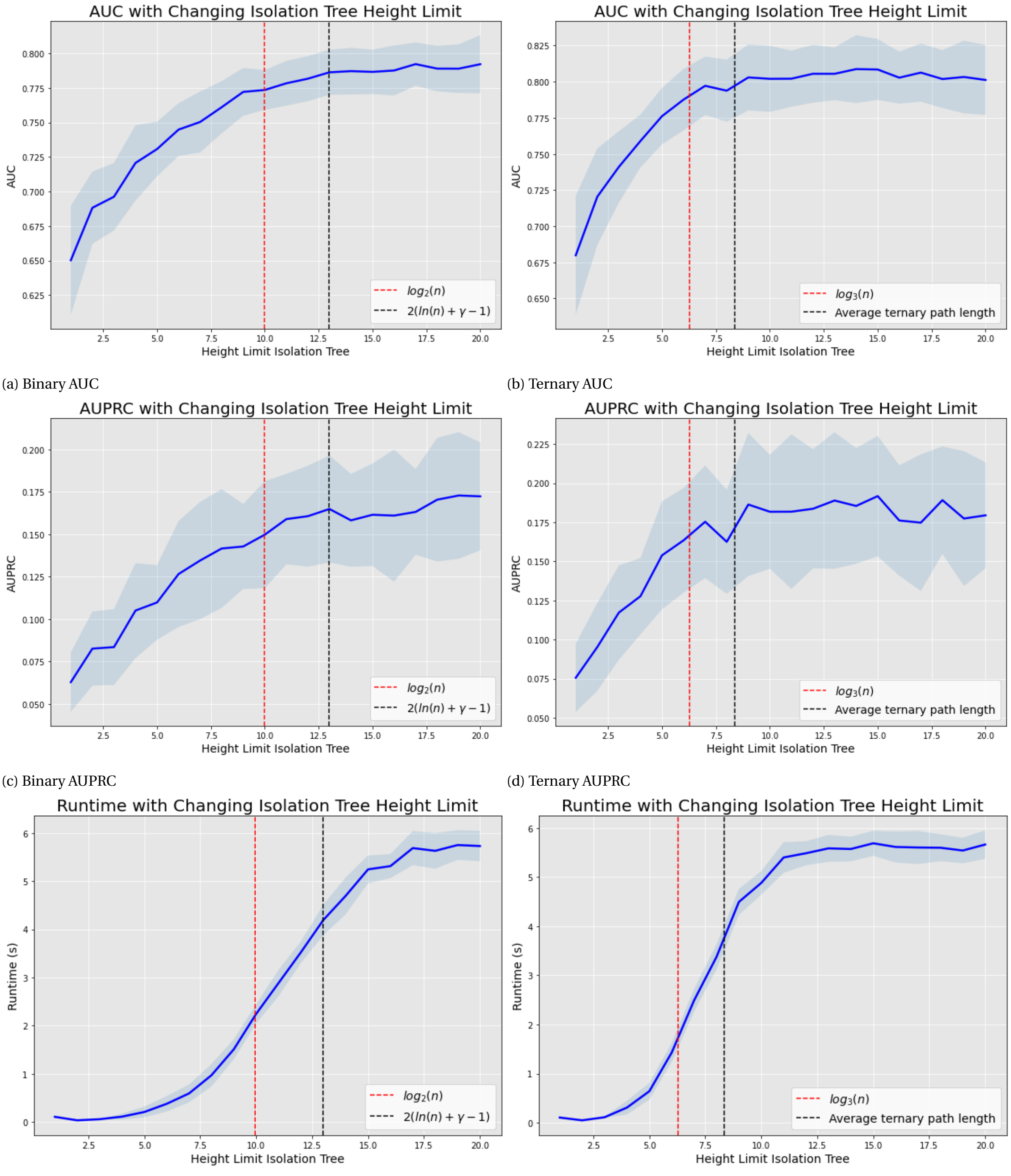
$$h = ceiling\left(\log_2(N+1) - 1\right),$$

where the height is rounded up to ensure an integer valued tree-height. Therefore, the expression that is introduced as a height limit in the original Isolation Forest paper, in fact approximates the minimum tree height given the total number of nodes in a binary tree.

In this thesis, the expected path length of an isolation tree is taken to be equal to the height limit of an isolation tree. Under the intuition that an anomaly is easier to isolate, and will therefore have a shorter than average path length in an isolation tree, taking the average path length seems appropriate. Therefore, the ceiling of the average path length, as derived in Subsection 2.3.2 and slightly different to the original $c(n)$ in [2], is taken as the height limit l:

$$l = ceiling(c(n)) = ceiling(2H(n) - 2)$$

To evaluate the newly proposed height limits, and consider the overall sensitivity of results with a varying height limit, a HiCS data set is used. By varying the height limit and then running the anomaly detection and local explanation procedures, the sensitivity of the height limit is illustrated. First, the performance of the Isolation Forest is depicted in Figure 4.10 for both a binary (left column) and a ternary (right column) implementation:



(a) Binary AUC

(b) Ternary AUC

(c) Binary AUPRC

(d) Ternary AUPRC

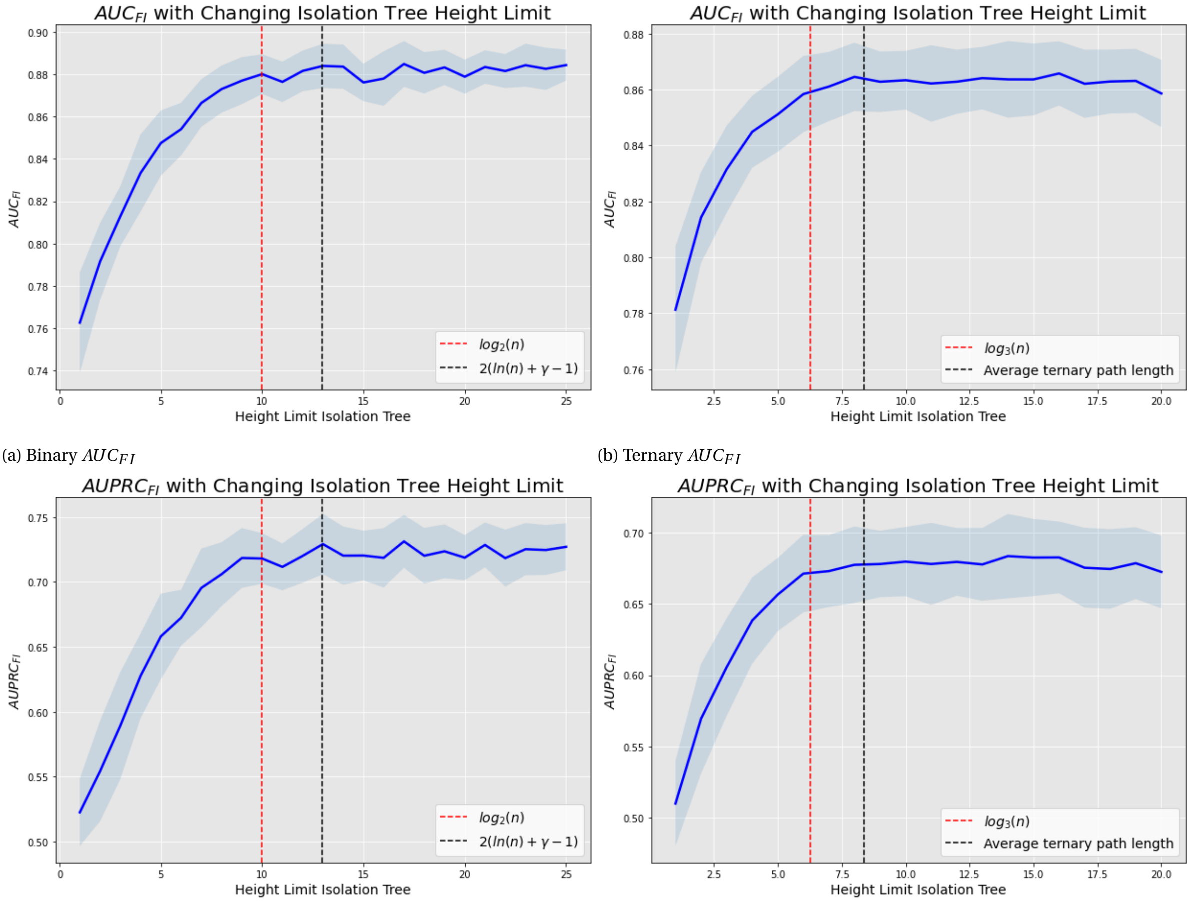
(e) Binary Runtime

(f) Ternary Runtime

Figure 4.10: Performance of binary and ternary Isolation Forest with variation in the algorithm's height limit parameter, averaged over 20 runs. The shaded region around the average result represent the standard deviation of the runs. Figure (a), (c), and (e) represent the results of the AUC, AUPRC, and runtime of a binary implementation, respectively. Figure (b), (d), and (f) represent the results of the AUC, AUPRC, and runtime of a ternary Isolation Forest implementation, respectively. The red vertical dotted lines represents the height limit in line with the original Isolation Forest paper [2], whereas the black line represents the theoretically computed average path length.

From Figure 4.10, it is evident that both the AUC and AUPRC increase significantly with increasing height limits up until a certain stagnation point. This stagnation for a binary tree occurs at a height limit of approximately 17, based on the stagnation of plot (e), whereas for a ternary tree this stagnation occurs at a height limit of approximately 12. When comparing the originally proposed height limit, marked by the vertical red line, to the theoretically computed average path length, it is evident that there is a slight improvement in terms of AUC and AUPRC. Allowing the isolation trees to grow to a larger extent therefore improves the results of the performance of both the binary and ternary Isolation Forest implementations. However, it should be noted that this comes at an increased runtime cost, as the training and evaluation stage of the Isolation Forest have a complexity of $\mathcal{O}(T\psi\log(\psi))$ and $\mathcal{O}(nT\log(\psi))$, respectively [2]. However, in the context of this thesis and detecting suspicious customers in the Triodos Bank client-base, it is argued that the performance of the algorithm is valued above the runtime. Finally, it is observed that the AUC and AUPRC results of a ternary isolation tree are better than that of the binary implementation for this particular data set.

Next, the results of the local explanation performance is addressed. Knowing the anomaly containing subspace of the data, the AUC_{FI} and the $AUPRC_{FI}$ can be computed with a varying isolation tree height limit. This is demonstrated in Figure 4.11



(a) Binary AUC_{FI}

(b) Ternary AUC_{FI}

(c) Binary $AUORC_{FI}$

(d) Ternary $AUPRC_{FI}$

Figure 4.11: Performance of MI-Local-DIFFI applied to binary and ternary Isolation Forest with varying height limits. The results are average over 50 runs, with the shaded region representing the standard deviation. Figure (a) and (c) represent the AUC_{FI} and $AUPRC_{FI}$ of a binary Isolation Forest, respectively. Figure (b) and (d) represent the AUC_{FI} and $AUPRC_{FI}$ of a ternary Isolation Forest, respectively. The vertical dashed lines represent the original height limit (red) and the theoretically computed average path length (black).

In Figure 4.11, it is shown that there is a steep increase in performance for the lower height limits, until the increasing performance stagnates. This is comparable to the behaviour demonstrated in Figure 4.11. Although the performance when using the theoretically computed path length as a height limit is slightly better than the originally proposed limit, the difference is not as distinct when compared to the AUC and

AUPRC results. What is interesting, is that both the results for AUC_{FI} and $AUPRC_{FI}$ are better for the binary implementation when compared to the ternary implementation. This again strengthens the findings of Section 4.4.

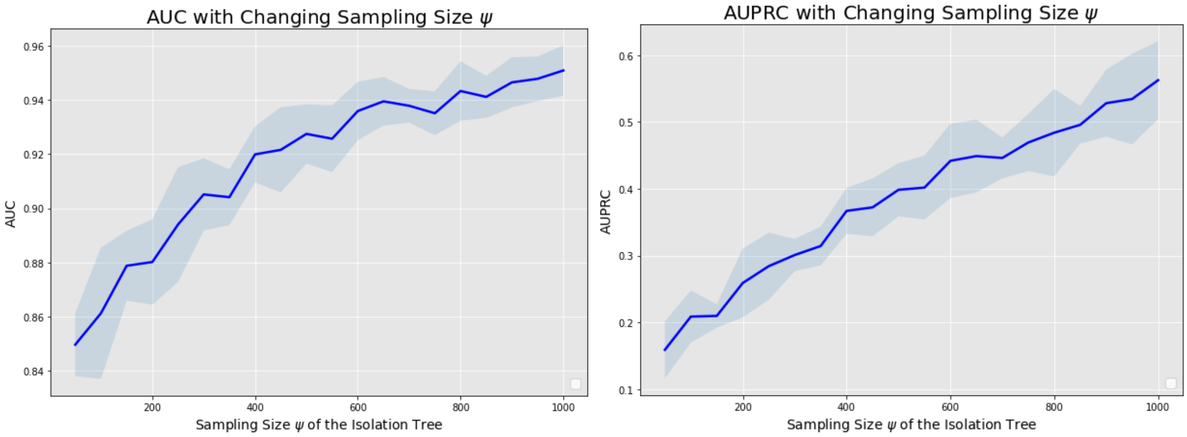
4.5.2. Isolation Forest Sub-sampling Size

In Subsection 2.3.2 the sub-sampling of the isolation tree is addressed. The construction of the Isolation Forest model is originally proposed using multiple sub-samples of the data. This sub-sampling reduces the effect of swamping and masking, in accordance to [2]. However, there are two reasons to critically reconsider this sub-sampling size parameter, ψ.

1. In the typical anomaly detection setting, the data set consists of extremely imbalanced classes. There is a large sample of normal instances, and only a minuscule percentage of the data that can be classified as an anomaly. Through sub-sampling, the sub-population used in the training of an isolation tree will frequently contain purely normal instances. This negatively impacts the ability of an isolation tree to isolate particular anomalies.

2. The height limit of an isolation tree is dependent on the sub-sampling size ψ. This means that with a smaller sub-sampling size, the isolation tree is pruned significantly earlier than the average computed path length given the number of input observations. The size of ψ, as shown in Subsection 4.5.1, can therefore significantly impact the performance of both the identification of anomalies and the feature importance results depending on the sub-sampling size.

To emphasize the downside to constructing an isolation tree using a sub-sample of the entire population, consider the performance of the AUC and AUPRC using a particular HiCS data set with a dimensionality of $d = 10$. The performance is evaluated while varying the sub-sample size ψ, from $\psi = 50$ up until $\psi = n = 1000$. From Figure 4.12 it is evident, within the context the HiCS data set, that for a binary Isolation Forest the results improve significantly as the sub-sampling size approaches $\psi = n$. This is clearly reflected in both the AUC and AUPRC results.

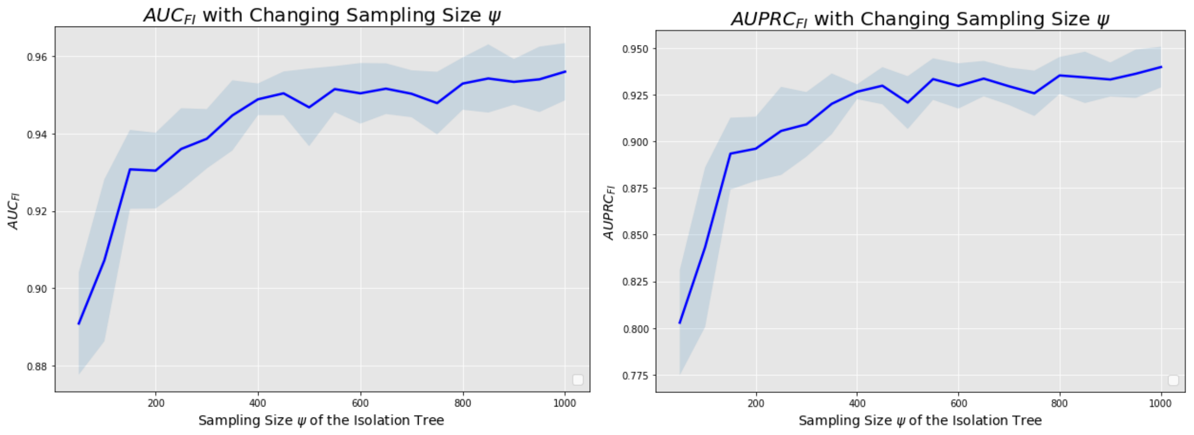


(a) AUC

(b) AUPRC

Figure 4.12: The performance of a binary Isolation Forest with underlying changes to the sub-sampling size parameter. Figure (a) represents the AUC results with increasing sub-sampling size ψ, while Figure (b) represents the AUPRC results with increasing sub-sampling size ψ.

The pattern of increased anomaly detection performance with increasing sub-sampling size ψ is also reflected in the performance of the feature importance scores. When using MI-Local-DIFFI to identify the most important features for isolating particular observations, the AUC_{FI} and $AUPRC_{FI}$ improve with an increase in the sub-sampling size as well, as shown in Figure 4.13.



(a) AUC

(b) AUPRC

Figure 4.13: The performance of MI-Local-DIFFI applied to a binary Isolation Forest with underlying changes to the sub-sampling size parameter. Figure (a) represents the AUC_{FI} results with increasing sub-sampling size ψ, while Figure (b) represents the $AUPRC_{FI}$ results with increasing sub-sampling size ψ.

Although Figure 4.12 and Figure 4.13 both demonstrate increased performance with an increase in the sub-sampling size ψ, be aware that this is not necessarily the case for all data sets. To counter the results of masking and swamping, an argument is made to consider a sample of the data for training purposes. However, in the context of money laundering detection and customer due diligence, it is argued that particularly the effect of masking reduces. With highly unbalanced classes and the resulting lack of large anomaly clusters, taking a sub-sample of the data for isolation tree training purposes is therefore assumed to negatively impact the effectiveness of the Isolation Forest model. Thus, in this thesis $\psi = n$ observations are used to construct every isolation tree.

4.6. Chapter Summary:

In this section, a summary of Chapter 4 is provided. As the chapter contains many experiments, the key findings have been summarized for additional clarity.

The iForest$_{CS}$ methodology proposed in this thesis demonstrates improved anomaly detection performance in the sensitivity analysis of Subsection 4.2.3 when compared to Isolation Forest with encoded nominal attributes. Furthermore, the pitfalls of encoding strategies and iForest$_{CAD}$ emphasized in Section 4.3 contribute to the argumentation of using iForest$_{CS}$ when working with real-life customer transaction data in Chapter 5. Note that the circumstances of performing anomaly detection over monthly transaction data is not time-sensitive. Thus, runtime complexity concerns are not weighed as heavily as performance results.

The analysis into the ternary Isolation Forest yielded interesting results. Originally, the hypothesis was held that a ternary Isolation Forest results in improved anomaly detection performance. However, using the newly computed average ternary path lengths from Subsection 3.2.2 as height limits, a ternary Isolation Forest does not outperform the binary Isolation Forest significantly on the HiCS data sets. In fact, when expanding the MI-Local-DIFFI methodology to ternary Isolation Forest, it was determined that the explanation performance deteriorates with the increased branching strategy. This results from the reduced probability of selecting (and hence detecting) feature subspaces containing anomalies. Therefore, in Chapter 5 a binary iForest$_{CS}$ is used to detect suspicious customer behaviour from real-life customer transaction data.

Finally, the Isolation Forest height limit and sub-sampling size as proposed by the original Isolation Forest paper [2], have been evaluated. It was determined that a height limit is considered that is not in line with the original intuition of the methodology. Instead, this thesis uses the following parameters:

- **Height limit:** $l = ceiling\Big(2H(n) - 2\Big)$
- **Sub-sampling size:** $\psi = n$

5
Applications and Results

Now that the methodology has been introduced and tested in the previous chapters, this chapter will focus on the applications of locally explainable anomaly detection. Through a fruitful collaboration with Triodos Bank, this thesis reaps the benefits from having the opportunity to experiment using real-world customer transaction data. This chapter will elaborate on the bank's data and procedures used to identify unusual customer behaviour, as well as present the results from the anomaly detection implementations.

Industry experts have been involved and provided useful guidance and expertise throughout this thesis. Particularly when utilizing sensitive, real-life customer transaction data, the experience and expertise in identifying money laundering modus operandi and suspicious customer behaviour has assisted in not only the construction of the anomaly detection data sets, but also the validation of results.

The chapter first discusses the impact of anomaly detection and the construction of the overall customer transaction data sets in Section 5.1. Following the construction of relevant data features, the results of the Isolation Forest methodologies are discussed in Section 5.2. This section will elaborate on the validation of detected anomalous customers, as well as provide a comparison between different implementations of Isolation Forest with mixed-attribute data.

5.1. Customer Transaction Data

Throughout this thesis, Isolation Forest models have been applied to data from the bank. Since banks possess the strong ambition to investigate the potential of AI applications in their current business practices, investigation into the success of using anomaly detection algorithms to detect fraudulent and suspicious customer behaviour is encouraged. Furthermore, this collaboration ensures that domain experts are involved throughout the entire process-chain; from assisting in the selection of features relevant for money laundering, up until the validation of the classified suspicious customers.

In this section, the process of constructing a relevant data set is discussed. First, the purpose and value of utilizing data-driven anomaly detection in business practices is reviewed in Subsection 5.1.1. Then, Subsection 5.1.2 will elaborate on the actual data and the corresponding features deemed relevant for the detection of money laundering. Finally, Subsection 5.1.3 discusses the process of classifying the detected anomalies using the expertise of alert handlers.

5.1.1. Value of Anomaly Detection

Banks strive to incorporate AI solutions in their current business practices. Particularly within the Customer Activity Monitoring (CAM) and CDD departments, the potential of including data-driven anomaly detection is investigated. Throughout the duration of this thesis research, there has been extensive collaboration into the application and value anomaly detection solutions can provide to the bank.

Before addressing the potential value that can be extracted from data-driven anomaly detection, it is important to stress that outcomes of the model are always revisited by domain-experts. First and foremost, the rule-based systems are optimized with money laundering and other financial fraudulent behaviour in mind. The incorporation of the anomaly detection models, with the associated local explanation methodologies, act as tools to improve the rule-based system and identify unseen customer behaviour. Finally, all results derived from the models used in this thesis are subjected to review by domain specialists. CDD specialists and alert handlers at the bank have both reviewed and validated the results before deciding on further actions.

Throughout the CDD process, there are numerous angles in which the application of anomaly detection can be proposed. The primary use-cases are expressed below. Note that not all of these applications have been addressed throughout this thesis, but are addressed to illustrate other potential use-cases. To some extent, these use-cases can be viewed as recommendations for further practical applications.

1. **Identifying suspicious customer behaviour not detected through the rule-based system:**
   Currently the rule-based system is responsible for generating alerts for suspicious customer activity. With a newly proposed methodology, one can generate a set of anomalies that potentially includes customers that were not detected through the current rules in place. Although there is a large overlap of the two sets of generated alerts, another perspective into identifying suspicious customer behaviour allows for more customers to be critically evaluated and therefore an increase in the potential identification of money laundering or financial fraud.

2. **Validation of transaction monitoring and introducing new rules:**
   When performing due diligence, there are well-known scenarios in money laundering. These are accounted for in a bank's rule-based system and customer behaviour is monitored according to a certain standard and combination of these scenarios. Through the use of data-driven anomaly detection, new features can be explored and incorporated into the input data sets. The resulting customer behaviour that is identified as anomalous can be locally examined to provide a validation of the existing transaction monitoring and whether these new features contribute to the detection. If the customer behaviour is then indeed suspicious and not yet known to the bank, an argument is made that additional rules can be constructed.

3. **Ranking of the rule-based alerts:**
   The rule-based system consists of various rules. These rules are all based on features and scenarios known to address risk. However, a large majority of these rules are static and an alert is generated once a threshold is met. Yet, there is no initial distinction between two customers that trigger an identical alert. With a coupling of anomaly scores to the rule-based generated alerts, an order can be introduced that may function as a prioritisation tool for alert handlers evaluating particular alerts or customers.

4. **Peer group evaluation:**
   Anomaly detection can be used on a wide variety of data. In this thesis, there is already a distinction made between private and business customers. There are however many peer group definitions that can prove insightful from an anomaly detection's effectiveness perspective. The more behaviour an underlying peer group is expected to share, the more distinct anomalies become. Furthermore, there is the consideration of the optimal size of these said peer groups.

In this thesis, focus is placed on the first two applications. The data set and its corresponding features are constructed with the most common money laundering scenarios in mind. However, some additional features are incorporated to evaluate whether new suspicious customer behaviour can be found. In line with the first research objective of this thesis, some of these new features are nominal. It is explored how the introduction of these additional nominal features affect the outcomes of the Isolation Forest model.

Furthermore, using domain-expert validation, customers that were detected through the Isolation Forest model and not through the rule-based system are evaluated for suspicious behaviour. When the use of data-driven anomaly detection becomes more embedded into the business practices of the bank, additional use-cases can be explored.

5.1.2. Feature Selection

The goal is to identify suspicious customer behaviour that is not detected through the usual channels as well as identify new risks factors that will potentially improve the rule-based system. In order to do this, the models are tested with real-life transaction data. This section will elaborate on the most important money laundering considerations and will give an idea of the features considered in the experimental data set. When discussing the features in a general sense, there are numerous measurement types applied to different scenarios. These are as follows:

- The total transaction volume over a specified time-period, either monthly or yearly.
- The total transaction frequency over a specified time-period, either monthly or yearly.
- Percentage of monthly transaction volume when compared to the yearly volume.

Note that when the total volume or frequency is stated, it indicates the summation of the transactions to and from a customer. Below, short descriptions are provided of particularly interesting risk factors that are considered for the customer transaction data sets throughout this thesis. The risk factors form the basis of particular rules and features, which will not be stated specifically.

International Transactions (with High Risk Areas and Fiscal Paradises)

It is important to consider all international transactions of a customer, as international transactions often are the cornerstone of layering illegal funding. With increasing international transactions, the origins of funds are easily concealed. Therefore, the international transactions of customers are monitored to determine suspicious or unexpected behaviour.

Particular countries have been identified by the Financial Action Taskforce (FATF) and the European Commission to show deficiencies in combating either money laundering or terrorist financing [51]. Both the FATF and the European Commission have an up-to-date list of countries on their website. Furthermore, the European Commission also maintains a list of countries that do not sufficiently combat tax evasion, fraud, or avoidance [51]. The consideration of these fiscal paradise countries is also considered throughout the customer due diligence processes and in the data set for anomaly detection purposes.

Cash and Money Transferring Services

There are three steps that are typically found in the money laundering process. The first, is the placement of illegitimate funds into the legitimate financial system. As illegitimate businesses often produce large amounts of cash, this step usually involves the placement of cash into the financial system. Therefore, financial institutions must remain aware of their customers cash transaction behaviour.

Money Transferring Services (MTS) refer to financial services involving the acceptance of cash, cheques, or other monetary instruments and the resulting payment to a beneficiary through a transfer or a clearing network [52]. MTS are attractive, lower cost options to send funds quickly to another individual when compared to wire transfers or more conventional banking services. These services are coupled with money laundering and terrorist financing risks and can therefore provide insight into suspicious customer behaviour.

Cryptocurrency

Cryptocurrencies are increasingly gaining popularity as a means of collecting, storing and cleaning of criminal proceedings. Particularly in cross-border money laundering, cryptocurrencies have become attractive to criminals, for a variety of reasons [53]:

- **Lack of regulations:** Financial institutions are heavily regulated and invest significantly in resources to detect fraud and protect the integrity of their financial products. Regulation of cryptocurrencies, however, are in comparison limited or negligible, allowing offenders to abuse cryptocurrencies for criminal purposes. Furthermore, there is no clear universal strategy to proactively counter misuse of a cryptocurrency platform.
- **Anonymity or pseudonymity:** Many wallet providers and crypto-exchanges offer transaction services with limited regulations. Particularly, limited KYC regulations are pertained in the crypto space, allowing for criminals to enjoy relative anonymity.
- **Criminal payment option:** Cryptocurrencies are already a common form of payment for criminal proceedings.

- **Ease of layering illegal funds:** Before integrating illegitimate funds into the economy, the origins of the funds can be concealed through a series of transactions. Using cryptocurrency channels, the ease of structuring these transactions makes it easier to conceal the trail.

Therefore, considering the rising trend of money laundering through cryptocurrency channels, transaction information regarding cryptocurrencies is incorporated into the data set.

Counter-Accounts

To understand the customer's transaction behaviour, it is essential to provide insight into the customer's relationships with other accounts. These other accounts to which a customer receives or transfers money from, are indicated with counter-accounts. The volumes, frequency, uniqueness, and geographical location of a customer's counter-accounts are all taken into consideration during the construction of the data sets.

Other

There are a couple of other scenarios that can indicate money laundering, terrorist financing, or other suspicious behaviour. In the data set, these features are also considered. These are:

- Transactions that are large in volume.
- Most common country where funds are transferred to and from (not the Netherlands). Here countries or regions can be categorized, for example.
- Business types. Additionally SBI codes can be interpreted as categorized data.

5.1.3. Validating Anomalies

With the features of the data set constructed, it is important to elaborate on the validation procedures of the model's detected anomalies. With real transaction data, there is no preconceived notion of the customers that can be classified as anomalies. Therefore, measuring detection performance of the Isolation Forest models is not as straightforward as with synthetic data sets. Instead, domain expertise is used to give an indication of the detected anomalies' effectiveness. Furthermore, generated anomalies have been introduced into the bank's official alert handling channels and have been evaluated by alert handlers. This feedback allowed for iterations to the features, as well as immediate validation of the model's outcomes.

When transaction behaviour is evaluated with the help of domain expertise, the customer behaviour is classified based on the relevance to detecting suspicious behaviour and observing interesting behaviour. This is done in the following section.

Alert Assessment

After a customer is classified as an anomaly, the respective customer is subjected to a rigorous evaluation process to determine follow-up procedures. The results from the models utilised throughout this thesis have been submitted to evaluation of both AML domain experts and alert handlers. The classifications are as follows:

1. **Not worthwhile:**
   All behaviour of a customer can be understood as not suspicious or explained.
2. **Worthwhile but no additional action required:**
   From the evaluation, certain components of customer behaviour are interesting, but do not indicate concerning suspicious behaviour with respect to money laundering or terrorist financing.
3. **Action required:**
   When customer behaviour is indicated as such, an additional follow-up procedure is required after the alert handler evaluation. This is, in the most suspicious cases, escalated and reported to third parties.

5.2. Results

This section will elaborate on the results of the Isolation Forest methodology to detect anomalies in the customer transaction data. Numerous different approaches have been used to compare the consistency of the detected anomalies and address the impact of mixed-attribute data from a practical context.

It is important to note that all alerts handled through the official channels are generated using a binary variant of $i\text{Forest}_{CS}$. This is done using the argumentation from Chapter 4. From experiments in Section 4.2, it is observed that the performance of detecting anomalies with 10% nominal attributes and 50 features is better when utilising $i\text{Forest}_{CS}$ than encoding the attributes. These parameters to the data correspond most with the customer transaction data to which Isolation Forest is applied. Additionally, the pitfalls of encoding strategies are considered in the application of $i\text{Forest}_{CS}$.

The binary $i\text{Forest}_{CS}$ is used over the ternary variant due to insight from Section 4.4. The improved explanation results of the binary implementation are preferred when trying to identify unknown suspicious behaviour.

The parameters of the Isolation Forest are aligned with the analysis of Section 4.5. Summarized, the parameters to the Isolation Forest experiments are as followed:

- **Number of isolation trees:**
  In the experiments, $T = 100$ is used. Although the performance slightly increases with an increase in the number of trees, the number of trees is taken in line with the original Isolation Forest paper.
- **Height limit of a isolation tree:**
  Using the insight into the depth of the isolation tree in Section 4.5, the depth of the isolation trees is set to the theoretically derived average path length of an isolation tree. For a binary isolation tree, the derivation is found in Subsection 2.3.2, while the ternary isolation tree is derived in Subsection 3.2.2.
- **Sub-sampling size**
  From the analysis on the performance effects of the isolation tree sub-sampling size in Section 4.5, the sub-sampling size ψ is set to the total number of observations. Hence, $\psi = n$.
- **Number of Isolation Forest runs**
  For all experiments and generated results, a total of 20 runs are performed. The results of these 20 runs are averaged to yield the ranking of the final anomalies.

There are numerous filtering steps performed over the customer transaction data. First, the customer-base is divided into business and private customers. As the behaviour of these two groups are distinct, and extra features are incorporated to business customers, the two groups are separated. Second, all customers with low monthly transaction volumes are omitted from the data sets. A threshold is introduced since anomalies with minimal transaction volumes are not subjected to further investigation from a money laundering perspective. Third, only active customers are considered. Fourth, only customers that have an account older than 3 months are considered. Otherwise, not enough information can be obtained to gain an understanding of the customer's typical behaviour. Finally, an upper transaction volume is introduced for business customers. Through evaluation of the results and the corresponding explanations, it was found that only “large" companies were isolated. However, particularly large companies are already placed under more frequent due diligence, resulting in a large overlap between anomalies and rule-based alerts.

When all relevant information is combined and filtered, the final data sets consists of approximately $25{,}000$ and $100{,}000$ business and private customers, respectively. The private customer data set has a dimensionality of 48, while the business customer data set incorporates extra information regarding business types, resulting in the consideration of 50 features. For the private customer data set, 2 features are nominal, whereas the business customer data set includes 4 nominal features.

This section is further divided into different sub-sections. First, the results of the alert handler validation are discussed in Subsection 5.2.1. Then, a comparison is made between the anomaly consistency of different Isolation Forest implementations and between anomaly detection results and alerts generated through the rule-based system in Subsection 5.2.2.

5.2.1. Domain Experts Validation

Spanning several months, a variety of data sets have been used for anomaly detection. In this section, the validation of the model's alerts by alert handlers is addressed using two months of data in particular. As there are no clear truth labels to the customers, the domain expertise of the alert handlers and AML experts is used to evaluate the outcomes of the anomaly detection model.

First, the approach of deciding which anomalies undergo validation is explained. As an Isolation Forest model omits an array of isolation scores that varies for every run due to the stochasticity of the forest's construction, deciding on the ordering of anomalies can be done using several subtly different methods. In this thesis, the average scores over all 20 runs are taken for all customers, after which the highest average isolation scores are considered anomalies. One might also consider introducing a threshold of the anomaly scores, yet in this thesis the constant size of the output anomaly array is preferred.

Then, these customer alerts are all compared to the set of existing rule-based alerts. The main purpose of using data-driven anomaly detection alongside a rule-based alert system is to identify suspicious transaction behaviour that was not originally detected through the rule-based system. Receiving validation over an overlapping set of customers will therefore reduce the insight of the Isolation Forest's detected anomalies and hinder the detection of new suspicious behaviour. Furthermore, the most important features needed to isolate “new" suspicious behaviour are analysed to address the identification of new risk factors.

Due to business and time constraints resulting from the constant rule-based alert-flow, approximately 20 customers resulting from the Isolation Forest model can additionally be submitted for review and validation per month. After validating earlier experiments with AML experts, anomalies in the private customer subspace were deemed most interesting. Therefore, a total of 14 business customers and 26 private customers have been submitted for extensive validation. The alert assessment assigned by the alert handler have been summarized in Table 5.1:

	Business Customers	**Private Customers**
1. Not worthwhile	8	8
2. Worthwhile but no additional action required	5	3
3. Action required	1	15

Table 5.1: The alert assessment of 14 business and 26 private customers as assigned by an alert handler. These alerts are generated using two distinct months of Triodos Bank transaction data

From Table 5.1, it is seen that the use of anomaly detection methods contributes to the detection of suspicious customer behaviour. Of the customers validated by alert handlers, 40% gained a classification of ‘Not worthwhile’. This indicates that the behaviour of particular customers can be justified after the alert handler's evaluation. From a data-induced perspective, customers may show unique patterns in their behaviour, however unique behaviour is not necessarily indicative of criminal intent.

The majority of customers submitted for validation, were determined to be ‘worthwhile but no additional action required’ or ‘action required’. This indicates that interesting information is obtained regarding customer behaviour and that additional follow-up procedures are initiated when a customer requires additional action. Through the application of the Isolation Forest model with mixed-attribute data, there are some observations that are detected and can be emphasized.

- Extrapolation of certain features created false-positive anomalies. These were revisited and corrected to some extent. This emphasizes again that Isolation Forest is data-driven, so inconsistencies in the data (generation) are brought to light.
- Customers are using accounts interchangeably for business and private purposes.
- There are inconsistencies between customer behaviour and the account information the bank possesses.

The bank is using particular insight provided by the Isolation Forest method to revisit some existing rules, as well as incorporate additional rules into their rule-based system. The most prominent behaviour and findings observed through validating the outcome of Isolation Forest with local explanations did not emanate directly from the incorporation of nominal features. Anomalies were more readily isolated in other subspaces of the data, and nominal features were rarely included in the top-three most important features. On the other hand, the incorporation of nominal attributes did impact the ordering and classification of particular customers as anomalies. In the following section, this will be emphasized further.

5.2.2. Comparison Different Isolation Forest Implementations

In this section, a comparison is made between different Isolation Forest implementations. These comparisons are conducted to provide insight into the contribution of nominal attributes, as well as the sensitivity to the underlying methodology of incorporating these attributes into an Isolation Forest.

To perform these comparisons, the transaction data of only one month is considered. With this data, multiple different implementations have been run to generate sets of outlying customers. Only the first 200 customers are considered in this comparison, as in practice it is only the top anomalies that will generate alerts and will be evaluated by alert handlers.

Consistency Comparison

First, the consistency of the top anomalies are compared. To make this comparison, consider the intersection of the lists of customer IDs ranked according to the anomaly scores. By taking increments of length 5, Figure 5.1 is constructed. This figure considers a particular partition of the ranked anomalies, and determines the intersection of the top x anomalies of different implementations. The figures show the comparison of the binary $i\text{Forest}_{CS}$ to numerous different implementations, for both the business customers and private customers.

First, notice that the utilization of binary $i\text{Forest}_{CS}$ indeed causes the intersection of the top anomalies to deviate when compared to omitting the nominal features altogether. With an intersection that is slightly less than 70% for the top 200 anomalies, there is an indication that the nominal features contribute to the isolation of customers.

When considering Figure 5.1, it is apparent that there is a relatively consistent overlap between the binary implementation of $i\text{Forest}_{CS}$ and the ternary $i\text{Forest}_{CS}$ and *Scikit-learn* Isolation Forest with frequency and label encoded data. For both the business and private customers, the binary and ternary implementations of $i\text{Forest}_{CS}$ share approximately 75% of its detected anomalies when considering the top 200 anomalies. When the data is encoded, however, the intersection of the anomalous customers tends to deviate. For the business customers, considering the top 200 customers, there is approximately a 70% overlap between the binary $i\text{Forest}_{CS}$ and the *Scikit-learn* implementation with frequency and label encoded data. When comparing the results for the private customers, however, the intersection increases to above the 80%. The data sets for private customers contain less nominal features than the business customers. This demonstrates the impact encoding of categorized data to a numerical scale has on the the resulting anomalies.

Unsurprisingly, the one-hot encoded data shares the least similarity when comparing the top identified anomalies. The resulting increase in dimensionality impacts the Isolation Forest's ability to detect anomalies as it impacts the probability of selecting a particular subset of features. Additionally, many sparse, binary indicators are introduced with the one-hot encoding of the data. Using the frequency encoding strategy, the pitfall of the encoding strategy also becomes clear when examining the nominal attributes. In the most extreme scenario, a particular nominal feature reduces in cardinality to approximately 33% of its original cardinality. This is the result in overlapping frequencies, that are all assigned a similar value after encoding. This particularly occurs with relatively unique categories in a nominal feature, resulting in loss of information.

Furthermore, the nominal features are examined more closely for the top 200 anomalies. When considering the top-3 local explanations for every particular anomaly, it was observed that anomalies are more readily isolated in subspaces of the data not dependent on nominal attributes. However, when observing the consistency amongst the top 200 anomalies of different methodologies, it is shown that with increasing

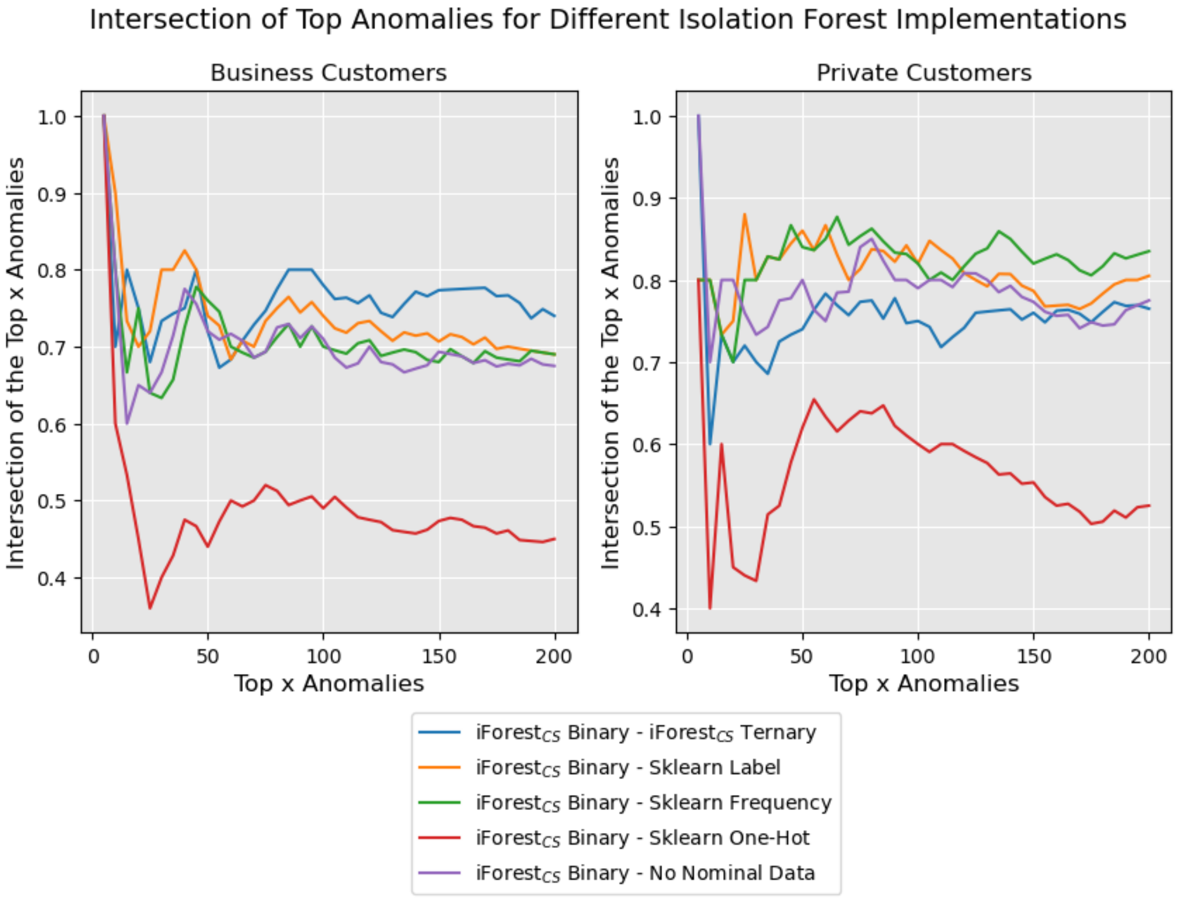


Figure 5.1: Comparison between the *i*Forest$_{CS}$ binary implementation and multiple other Isolation Forest implementations. The comparison is made against the ternary implementation of *i*Forest$_{CS}$ and three different encoding strategies, namely label encoding, frequency encoding, and one-hot encoding. Finally, Isolation Forest is also performed on the data set without nominal features. The plot visualizes the intersection of the top x outlying customers detected compared to anomalies detected with the *i*Forest$_{CS}$ binary implementation. So, for example, when considering the top 100 anomalies, the overlap between the binary and ternary *i*Forest$_{CS}$ implementation is approximately 0.79, meaning 79 identical customers are detected within the first 100 anomalies. The figure demonstrates the anomaly intersection trend across the 200 customers receiving the highest anomaly scores. The intersection values are computed at intervals of length 5. Figure (a) depicts the business customers, while Figure (b) depicts the private customers.

number of nominal attributes, the intersection between different anomaly sets reduces. This gives an indication that the nominal features indeed impact the identification of particular customer behaviour. In Figure 5.2 and Figure 5.3, the histograms of the feature importance rankings of the nominal features for the business and private customers is addressed, respectively. From these histograms, insight into the importance of these nominal features can be derived.

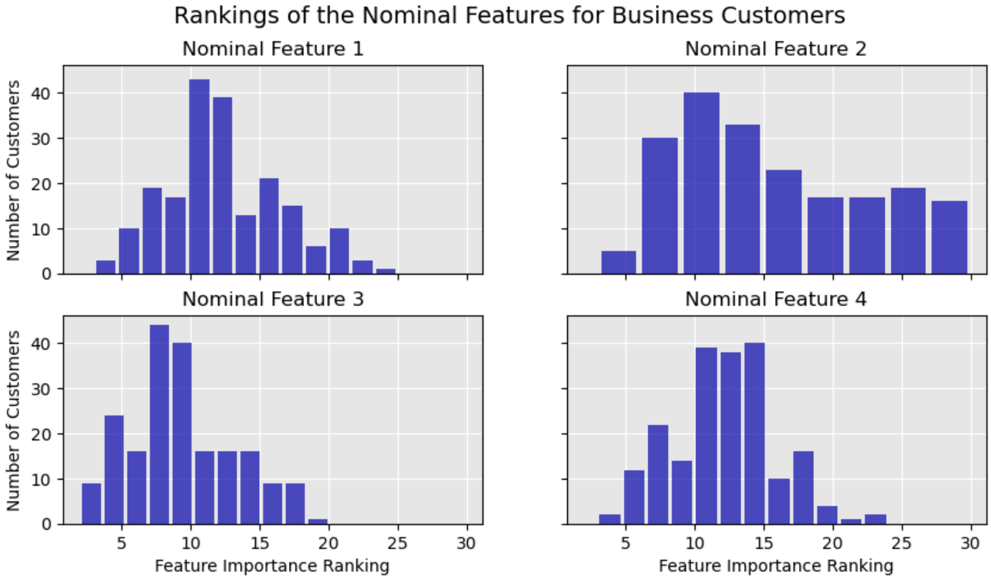


Figure 5.2: The histogram portraying the feature importance rankings of all 4 nominal attributes in the business customers data subset. Considering the top 200 anomalies, the feature importance of all data features (50 total) can be computed locally and ranked. The histograms demonstrate that the nominal features consistently obtain relatively high feature importance rankings, but are not often contained in the absolute top features that cause isolation. The feature importance rankings are determined using MI-Local-DIFFI on the binary $i\text{Forest}_{CS}$ implementation.

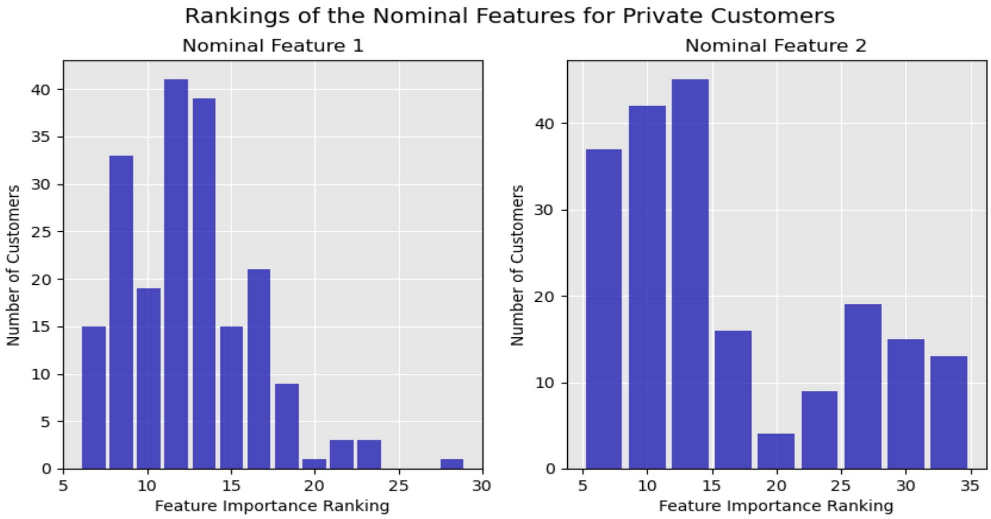


Figure 5.3: The histogram portraying the feature importance rankings of the 2 nominal attributes in the private customers data subset. Considering the top 200 anomalies, the feature importance of all data features (48 total) can be computed locally and ranked. The histograms demonstrate that the nominal features consistently obtain relatively high feature importance rankings, but are not contained in the absolute top features that cause isolation. The feature importance rankings are determined using MI-Local-DIFFI on the binary $i\text{Forest}_{CS}$ implementation.

As observed in Figure 5.2 and Figure 5.3, the distribution of the feature importance rankings indicate that the nominal attributes indeed contribute to the isolation of customers with Isolation Forest. There are, however, limited instances in which the nominal features belong to the most important features to isolate a customer. Particularly for the private customers, the nominal features considered never belong to the top-5 important features. This indicates that any anomaly admitted to the alert handler from this set of 200 private customers would never include information regarding a nominal attribute.

When constructing the results of Table 5.2, all 'Action required' customers identified in the analysis month did not contain nominal attributes in the top feature importance rankings. Since no nominal attributes were important in identifying these customers, the different implementations of Isolation Forest would have isolated these particular customers as well. Only the one-hot encoding and Ternary iForest$_{CS}$ implementations failed to identify one of these customers within the first 200 anomalies. All other implementations identified the customers that required additional action. This emphasizes that regardless of the encoding of the nominal attributes, anomalies contained in subspaces of other features are still consistently identified.

Comparison to the Rule-Based System

Since banks continuously utilize a rule-based system, a comparison can be made between the anomalies detected and the scores assigned to previously alerted customers. Table 5.2 compares the top 200 anomalies to the alerts that the bank had generated. The 'Other' alert assessment indicates the alerts that yielded no additional actions, as well as the alerts that are still being processed. The 'Action required' alert assessment is in line with Subsection 5.1.3. Furthermore, the table depicts the number of anomalous customers that have not received an alert through the rule-based system, indicated with 'No Alerts'.

	Alert Assessment	**Business Customers**	**Private Customers**
Binary iForest$_{CS}$	Other	60.5%	57%
	Action required	8%	17%
	No Alerts	31.5%	26%
Ternary iForest$_{CS}$	Other	63.5%	55%
	Action required	7%	19.5%
	No Alerts	29.5%	25.5%
Frequency Encoding	Other	70%	56%
	Action required	5%	19%
	No Alerts	25%	25%
Label Encoding	Other	69%	57%
	Action required	5%	18.5%
	No Alerts	26%	24.5%
One-Hot Encoding	Other	66%	44.5%
	Action required	7%	13%
	No Alerts	27%	42.5%
No Nominal Attributes	Other	69.5%	57%
	Action required	4.5%	18%
	No Alerts	26%	25%

Table 5.2: Comparison between different Isolation Forest implementations and the rule-based system generated alerts. The 'Other' alert assessment indicates the alerts that require additional action, as well as the the alerts that are still being processed. The 'Action required' alert assessment is in line with Subsection 5.1.3. Furthermore, the table depicts the number of anomalous customers that have not received an alert through the rule-based system, indicated with 'No Alerts'. It is important to note that customers that have not generated alerts are the most interesting from a practical perspective, as a goal of using data-driven anomaly detection is to gain additional insight into customer behaviour unnoticed beforehand. The percentages indicate how many customers of the top 200 anomalies have received the respective alert assessment.

From Table 5.2, it is observed that a large percentage of the top anomalies are indeed false positives. This is as expected. When considering the skewed class distributions, there is only a small fraction of the overall customer population that is actually suspicious or guilty of laundering money or financing terrorism. On the other hand, however, when comparing the results to the alerts of the rule-based system, all methods would have returned a significant portion of customers requiring additional action. This is an excellent confirmation of the potential of anomaly detection in detecting suspicious customer behaviour in the context of AML.

The customers that had not generated any alerts through the rule-based system, but are amongst the top 200 detected anomalies, are labeled as “No Alerts” in Table 5.2. As can be seen, a significant percentage of the top anomalies had not generated prior alerts. These are interesting customers, as potentially new insight can be gained regarding behaviour not reflected through the rule-based system. Additionally, using the alerts from Table 5.1 as argumentation, there is value and information contained in this customer subset. However, between the different implementations, the similarity between the “No Alerts” customers were was small. This does indicate that the nominal attributes contribute to isolation, even when the most suspicious behaviour did not emanate directly from these features.

5.3. Chapter Summary

In this section, a summary is provided of the chapter. This lists the key take-aways and findings of this chapter.

1. Over the course of this thesis, a total of 16 customers have been directly classified as customers that require additional action. This entails that an additional follow-up procedure is required after the alert handler has validated the customer alerts.

2. New suspicious behaviour has been identified through the inclusion of new features into the data. This will result in the improvement and generation of new rules.

3. The most suspicious behaviour did not emanate from nominal attributes. On average, these attributes demonstrated some indication of assisting in the isolation of customer behaviour. However, through the local feature importance explanations, nominal attributes were rarely identified as the top important features.

6
Conclusion & Recommendations

This chapter will draw conclusions from all experiments conducted throughout this thesis and provide recommendations for future research. In Section 6.1 the research objectives stated in Chapter 1 are revisited and answered. Then, Section 6.2 will briefly state the contributions of this thesis. Finally, Section 6.3 will discuss potential future research topics.

6.1. Conclusion

In this section, the research objectives stated in Chapter 1 will be answered and argued using results derived from this thesis.

RO1: *How should mixed-attribute data be incorporated into locally explained Isolation Forest?*

First, it is important to state that nominal features provide additional insight into the detection of anomalies. As demonstrated in Section 4.3, without the consideration of nominal features, certain anomalies will never be detected.

When working with mixed-attribute data, typically all nominal attributes are encoded to a numerical scale. Although in numerous practical applications this is proven to be sufficient, translating a non-ordered categorical variable to a numerical scale intuitively raises questions. To name some examples, encoding strategies can result in the ordering of unordered categories, the assigning of similar encoded values when categories appear with similar frequency, and the increase in the data set's overall dimensionality.

In this thesis, a new methodology is proposed, $i\text{Forest}_{CS}$, that directly samples amongst the categories of a nominal feature when selected in a given node of the isolation tree. This method does not require additional encoding of nominal features and demonstrates improved performance when evaluating it to other encoding strategies using independently sampled, mixed-attribute synthetic data. Furthermore, the downsides that can accompany an encoding strategy are not relevant when directly sampling from the categories in a nominal feature.

$i\text{Forest}_{CS}$ is also used with real-life, mixed-attribute customer transaction data. Although the nominal attributes in the data only rarely belonged to the top-3 most important features for isolating a customer, the inclusion of only a couple of nominal features can cause differences in the top anomalies detected.

Thus, to conclude, preference is placed on using $i\text{Forest}_{CS}$ to detect anomalies in mixed-attribute data.

RO2: *Does a ternary tree structure improve an Isolation Forest's ability to detect anomalies and provide local explanations?*

Throughout this thesis, the ternary Isolation Forest has been explored in more detail. Through an analysis performed in earlier research, the original hypothesis stated that performance can be improved using a ternary isolation tree structure over a binary structure.

To justify this hypothesis, derivations of average path lengths of ternary isolation trees were revisited. As there are no built-in Python libraries to construct a ternary Isolation Forest, the Isolation Forest algorithm is constructed and altered to incorporate the ternary branching strategies. Using this implementation, the ternary Isolation Forest could be extended to incorporate mixed-attribute data. Furthermore, MI-Local-DIFFI is adapted to dissect the ternary isolation tree nodes for feature importance information as well.

There are numerous findings derived from Chapter 4 over ternary Isolation Forest. First, the probability of selecting a certain subspace in a data set reduces when using ternary isolation trees. This is the direct result of the smaller (theoretical) average path lengths for a ternary isolation tree. As a binary tree experiences relatively less fragmentation in every node, the path length of an anomaly will be longer and contains more information about anomaly containing sub-spaces. This negatively impacts the ability to detect anomalies contained in more complex subsets of the data, which was reflected when considering experiments with the HiCS data sets. Furthermore, the split interval indicator of MI-Local-DIFFI carries less information when increasing from a binary to a ternary splitting strategy.

When applying Isolation Forest to mixed-attribute customer transaction data, the binary implementation of $i\text{Forest}_{CS}$ was chosen for submitting anomalies to alert handlers. This was done since improved explanation results are preferred when attempting to identify unknown suspicious behaviour.

Thus, a ternary tree structure can for certain data sets provide improved results regarding the ability to detect anomalies. However, the decrease in probability of selecting a particular subspace with a ternary isolation tree impacts the local explanation capabilities. In the practical setting of this thesis, the reduced explanation performance outweighed the other considerations. Thus, the original hypothesis that ternary Isolation Forest yields improved results, is not confirmed.

6.2. Contributions

The contributions made throughout this thesis are as follows:

1. A new approach to incorporating nominal features within Isolation Forest is introduced, namely $i\text{Forest}_{CS}$. This approach respects the essence of the Isolation Forest; the random, data-induced splits are maintained in a nominal attribute by randomly selecting categories. This method generates improved results when compared to common encoding strategies.

2. A more detailed analysis of the comparison between results of a binary and ternary Isolation Forest is performed. Additionally, the MI-Local-DIFFI method is extended to incorporate ternary isolation trees, allowing for an evaluation of the local explanation performance.

3. An evaluation on real customer transaction data involving nominal attributes has been conducted. Using data-driven anomaly detection, new suspicious customers and additional risk factors have been identified.

4. Parameters originally proposed for the Isolation Forest algorithm have been evaluated, resulting in the proposal of new height limits and sub-sampling sizes when using Isolation Forest.

6.3. Recommendations

In this section, recommendations for future research are discussed. Every paragraph below indicates a new recommendation:

Throughout this thesis, comparisons have been drawn to various encoding strategies. These encoding strategies share certain similarities; the categorical encoding is performed independent from other attributes in the data and they are used within the context of anomaly detection with Isolation Forest. This is done to accommodate for the fact that the nominal attributes in the customer transaction data share dependencies to numerous other features. A particular form of encoding that has not been experimented with, is the encoding based on a target feature. One can encode a nominal feature on the basis of the mean, variance, or higher moment of a target feature, for example. There are a couple of side-notes to consider with this approach. The first, is determining the relevant feature that acts as a target feature. In customer transaction data, a nominal attribute can impact the expected international, cash, or transaction volume behaviour of a customer, for example. The question that therefore arises is one of choosing the correct target feature.

A common local explanation method that is readily available in Python libraries, is Tree SHAP. In this thesis, the newly proposed iForest$_{CS}$ has not been constructed to accommodate analysis with Tree SHAP. This holds true for the ternary Isolation Forest as well. To improve further research into these Isolation Forest implementations, the algorithm's should be constructed to resemble the $Scikit-learn$ implementation of Isolation Forest more.

In Section 4.4 it was concluded that the split interval indicator of MI-Local-DIFFI loses significance with an increasing n-ary isolation tree. This occurs due to the decreasing expected anomaly split interval when the branching strategy of an isolation tree is expanded. For further research, it is interesting to modify the split interval indicator to account for an increasing n-ary isolation tree and observe whether the local explanation performance improves.

Furthermore, the split interval indicator of MI-Local-DIFFI depends on the underlying feature distribution and a notion of distance. With a nominal attribute, this indicator does not translate. It is therefore interesting to explore whether the indicator can be adapted to incorporate nominal features directly, or if other information of isolation trees can be exploited to improve MI-Local-DIFFI. In Appendix C, an initial idea is proposed that has been explored briefly throughout this thesis. It is recommended to continue research in this direction to derive more robust conclusions.

From a practical point-of-view, it is interesting to consider anomaly detection applied to particular peer-group definitions. When considering the real-life customer transaction data, using peer-group definitions to identify smaller subsets of the data can improve the detection and explanation of anomalies. The more behaviour a peer group is expected to share, the more distinct anomalies within these peer-groups become. Therefore, the process of defining peer-group definitions is from a practical perspective an interesting future endeavor.

Bibliography

[1] *Transactie Monitoring Nederland unieke stap in strijd tegen witwassen en terrorismefinanciering.* https://www.nvb.nl/nieuws/transactie-monitoring-nederland-unieke-stap-in-strijd-tegen-witwassen-en-terrorismefinanciering/. July 2020.

[2] F.T Liu, K. Ting, and Z.H Zhou. "Isolation Forest". In: *2008 Eighth IEEE International Conference on Data Mining* (2008), pp. 413–422. DOI: 10.1109/ICDM.2008.17.

[3] M. Carletti et al. "Explainable Machine Learning in Industry 4.0: Evaluating Feature Importance in Anomaly Detection to Enable Root Cause Analysis". In: *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)* (2019), pp. 21–26. DOI: 10.1109/SMC.2019.8913901.

[4] K.B Bergþórsdóttir. "Local Explanation Methods for Isolation Forest: Explainable Outlier Detection in Anti-Money Laundering". MA thesis. Delft University of Technology, Aug. 2020.

[5] UNODC. *Money Laundering.* https://www.unodc.org/unodc/en/money-laundering/overview.html. Accessed: 02-02-2021.

[6] FATF. *Money Laundering.* https://www.fatf-gafi.org/faq/moneylaundering/#d.en.11223. Accessed: 02-02-2021. 2020.

[7] *Wet ter voorkoming van witwassen en financieren van terrorisme.* https://wetten.overheid.nl/BWBR0024282/2021-07-01. July 2008.

[8] FATF (2012-2020). "International Standards on Combating Money Laundering and the Financing of Terrorism Proliferation". In: (). http://www.fatf-gafi.org/publications/fatfrecommendations/documents/fatf-recommendations.html.

[9] Netherlands Public Prosecution Service. "Investigation Houston Criminal investigation into ING Bank N.V." In: (). https://www.om.nl/documenten/publicaties/fp-hoge-transacties/feitenrelaas/map/feitenrelaas-ing.

[10] Netherlands Public Prosecution Service. "Investigation Guardian Criminal investigation into ABN AMRO Bank N.V." In: (). https://www.om.nl/documenten/publicaties/fp-hoge-transacties/feitenrelaas/map/guardian-feitenrelaas-en-beoordeling-door-het-om.

[11] Joost Van der Burgt. "General principles for the use of Artificial Intelligence in the financial sector". In: (2019).

[12] Charu C. Aggarwal. *Outlier Analysis.* Springer International Publishing, 2017. ISBN: 978-3-319-47577-6. DOI: 10.1007/978-3-319-47578-3.

[13] F.T Liu, K. Ting, and Z.H Zhou. "Isolation-Based Anomaly Detection". In: *ACM Transactions on Knowledge Discovery From Data - TKDD* 6 (Mar. 2012), pp. 1–39. DOI: 10.1145/2133360.2133363.

[14] Remi Domingues, Maurizio Filippone, and Jihane Zouaoui. "A comparative evaluation of outlier detection algorithms: Experiments and analyses". In: *Pattern Recognition* 74 (Sept. 2017). DOI: 10.1016/j.patcog.2017.09.037.

[15] Bernhard Schölkopf et al. "Support Vector Method for Novelty Detection". In: vol. 12. Jan. 1999, pp. 582–588.

[16] S. P. Lloyd. "Least squares quantization in PCM". In: *IEEE Trans. Inf. Theory* 28 (1982), pp. 129–136.

[17] Martin Ester et al. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: KDD'96. Portland, Oregon: AAAI Press, 1996, pp. 226–231.

[18] R. Sibson. "SLINK: An Optimally Efficient Algorithm for the Single-Link Cluster Method". In: *Comput. J.* 16 (1973), pp. 30–34.

[19] Stephen D. Bay and Mark Schwabacher. "Mining Distance-Based Outliers in near Linear Time with Randomization and a Simple Pruning Rule". In: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '03. Washington, D.C.: Association for Computing Machinery, 2003, pp. 29–38. ISBN: 1581137370. DOI: 10.1145/956750.956758. URL: https://doi.org/10.1145/956750.956758.

[20] Markus Breunig et al. "LOF: Identifying Density-Based Local Outliers." In: vol. 29. June 2000, pp. 93–104. DOI: 10.1145/342009.335388.

[21] S. Papadimitriou et al. "LOCI: fast outlier detection using the local correlation integral". In: *Proceedings 19th International Conference on Data Engineering (Cat. No.03CH37405)*. 2003, pp. 315–326. DOI: 10.1109/ICDE.2003.1260802.

[22] Markus Goldstein and Andreas Dengel. "Histogram-based Outlier Score (HBOS): A fast Unsupervised Anomaly Detection Algorithm". In: Sept. 2012.

[23] JooSeuk Kim and Clayton D. Scott. "Robust Kernel Density Estimation". In: *Journal of Machine Learning Research* 13.82 (2012), pp. 2529–2565. URL: http://jmlr.org/papers/v13/kim12b.html.

[24] Kevin Beyer et al. "When Is "Nearest Neighbor" Meaningful?" In: *ICDT 1999. LNCS* 1540 (Dec. 1997).

[25] Alexander Hinneburg, Charu Aggarwal, and Daniel Keim. "What is the Nearest Neighbor in High Dimensional Spaces?" In: *First publ. in: Proc. of the 26th Internat. Conference on Very Large Databases, Cairo, Egypt, 2000, pp. 506-515* 671675 (Oct. 2000).

[26] S. Hariri, M. Kind, and R. Brunner. "Extended Isolation Forest with Randomly Oriented Hyperplanes". In: *IEEE Transactions on Knowledge and Data Engineering* PP (Oct. 2019), pp. 1–1. DOI: 10.1109/TKDE.2019.2947676.

[27] M. Carletti, M. Terzi, and G.A Susto. "Interpretable Anomaly Detection with DIFFI: Depth-based Feature Importance for the Isolation Forest". In: (July 2020).

[28] Scott M. Lundberg, G. Erion, and Su-In Lee. "Consistent Individualized Feature Attribution for Tree Ensembles". In: *ArXiv* abs/1802.03888 (2018).

[29] Scott M. Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 4768–4777. ISBN: 9781510860964.

[30] A. Taha and A.S. Hadi. "Anomaly Detection Methods for Categorical Data: A Review". In: *ACM Computing Surveys* (Jan. 2019). DOI: https://doi.org/10.1145/1122445.1122456.

[31] Yun Wang. "Statistical Techniques for Network Security: Modern Statistically-Based Intrusion Detection and Protection". In: *Statistical Techniques for Network Security: Modern Statistically-Based Intrusion Detection and Protection* (Jan. 2008), pp. 1–461. DOI: 10.4018/978-1-59904-708-9.

[32] D. Maier. "The Theory of Relational Databases". In: 1983.

[33] Patricio Cerda, Gaël Varoquaux, and Balazs Kegl. "Similarity encoding for learning with dirty categorical variables". In: *Machine Learning* 107 (Sept. 2018). DOI: 10.1007/s10994-018-5724-2.

[34] J. Cohen and P. Cohen. *Applied Multiple Regression/correlation Analysis for the Behavioral Sciences*. Taylor & Francis Group, 1983. ISBN: 9780898592689.

[35] Yibo Wang and Wei Xu. "Leveraging deep learning with LDA-based text analytics to detect automobile insurance fraud". In: *Decision Support Systems* 105 (2018), pp. 87–95. ISSN: 0167-9236. DOI: https://doi.org/10.1016/j.dss.2017.11.001.

[36] E. Stripling et al. "Isolation-based conditional anomaly detection on mixed-attribute data to uncover workers' compensation fraud". In: *Decision Support Systems* 111 (2018), pp. 13–26. ISSN: 0167-9236. DOI: https://doi.org/10.1016/j.dss.2018.04.001.

[37] Meghanath M Y, Deepak Pai, and Leman Akoglu. "ConOut: Contextual Outlier Detection with Multiple Contexts: Application to Ad Fraud". In: Jan. 2019, pp. 139–156. ISBN: 978-981-13-6048-0. DOI: 10.1007/978-3-030-10925-7_9.

[38] Dechang Chen et al. "On Detecting Spatial Outliers". In: *GeoInformatica* 12 (Dec. 2008), pp. 455–475. DOI: 10.1007/s10707-007-0038-8.

[39] Li Sun et al. “Detecting Anomalous User Behavior Using an Extended Isolation Forest Algorithm: An Enterprise Case Study”. In: (Sept. 2016).

[40] Sunil Aryal, Kai Ming Ting, and Gholamreza Haffari. “Revisiting Attribute Independence Assumption in Probabilistic Unsupervised Anomaly Detection”. In: *Intelligence and Security Informatics*. Ed. by Michael Chau, G. Alan Wang, and Hsinchun Chen. Cham: Springer International Publishing, 2016, pp. 73–86. ISBN: 978-3-319-31863-9.

[41] Fei Tony Liu. *Anomaly detection using isolation*. Jan. 2017. DOI: 10.4225/03/58901b400c59b. URL: https://bridges.monash.edu/articles/thesis/Anomaly_detection_using_isolation/4597651/1.

[42] Mathieu Garchery and Michael Granitzer. “On the influence of categorical features in ranking anomalies using mixed data”. In: *Procedia Computer Science* 126 (2018). Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 22nd International Conference, KES-2018, Belgrade, Serbia, pp. 77–86. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2018.07.211. URL: https://www.sciencedirect.com/science/article/pii/S1877050918311852.

[43] F. Keller, E. Müller, and Klemens Böhm. “HiCS: High Contrast Subspaces for Density-Based Outlier Ranking”. In: *2012 IEEE 28th International Conference on Data Engineering* (2012), pp. 1037–1048.

[44] Leo Breiman. “Machine Learning, Volume 45, Number 1 - SpringerLink”. In: *Machine Learning* 45 (Oct. 2001), pp. 5–32. DOI: 10.1023/A:1010933404324.

[45] STUDENT. “THE PROBABLE ERROR OF A MEAN”. In: *Biometrika* 6.1 (Mar. 1908), pp. 1–25. ISSN: 0006-3444. DOI: 10.1093/biomet/6.1.1. eprint: https://academic.oup.com/biomet/article-pdf/6/1/1/605641/6-1-1.pdf. URL: https://doi.org/10.1093/biomet/6.1.1.

[46] S. S. SHAPIRO and M. B. WILK. “An analysis of variance test for normality (complete samples)†”. In: *Biometrika* 52.3-4 (Dec. 1965), pp. 591–611. ISSN: 0006-3444. DOI: 10.1093/biomet/52.3-4.591. eprint: https://academic.oup.com/biomet/article-pdf/52/3-4/591/962907/52-3-4-591.pdf. URL: https://doi.org/10.1093/biomet/52.3-4.591.

[47] J. Brainard and D. E. Burmaster. “Machine Learning, Volume 45, Number 1 - SpringerLink”. In: *Risk Analysis* 12 (June 1992), pp. 267–275. DOI: https://doi.org/10.1111/j.1539-6924.1992.tb00674.x.

[48] WorldData. *Average sizes of men and women.* https://www.worlddata.info/average-bodyheight.php. Accessed: 01-08-2021.

[49] F. Keller, E. Müller, and Klemens Böhm. “HiCS: High Contrast Subspaces for Density-Based Outlier Ranking”. In: *2012 IEEE 28th International Conference on Data Engineering* (2012), pp. 1037–1048.

[50] Donald E. Knuth. *The Art of Computer Programming, Volume 3: (2nd Ed.) Sorting and Searching*. USA: Addison Wesley Longman Publishing Co., Inc., 1998. ISBN: 0201896850.

[51] De Nederlandsche Bank. “Guideline on the Anti-Money Laundering and Anti-Terrorist Financing Act and the Sanctions Act”. In: (Dec. 2019).

[52] FATF. “Guidance for a Risk-Based Approach for Money or Value Transfer Services”. In: (2016).

[53] Tookitaki. *The Rise in Cryptocurrency Money Laundering Cases in 2021*. https://www.tookitaki.ai/news-views/the-rise-in-cryptocurrency-money-laundering-cases-in-2021/. Accessed: 08-09-2021.

[54] Davide Ceolin et al. “Uncertainty Estimation and Analysis of Categorical Web Data”. In: Jan. 2014, pp. 265–288. ISBN: 978-3-319-13412-3. DOI: 10.1007/978-3-319-13413-0_14.

[55] K. W. Ng, G.-L. Tian, and M.-L. Tang. *Dirichlet and related distributions: Theory, methods and applications*. Wiley, 2011. ISBN: 047068819X.

A
Multinomial-Dirichlet Distribution

Often when handling nominal features, the Binomial or Multinomial distribution are natural modelling considerations, depending on the number of categories the data is divided into [54]. Both distributions allow for the modeling of n draws from a feature data set, often with the presumption that category frequencies are known. When generating categorical data, category frequencies can be established beforehand. However, the parameters of the Binomial or Multinomial distributions can also be generated from the Beta or Dirichlet distributions respectively, considering these are its conjugate priors.

The Dirichlet-Multinomial modelling of nominal features is parametric, and can be classified as an empirical Bayesian model [54]. To further elaborate on the distribution, some definitions are introduced for formality:

Definition A.0.1. *The* ***n-dimensional closed simplex*** *is defined as*

$$\mathbb{T}_n(c) = \left\{ (x_1, \ldots, x_n)^T : x_i > 0, 1 \leq i \leq n, \sum_{i=1}^{n} x_i = c \right\},$$

where c is a positive number. We define $\mathbb{T}_n = \mathbb{T}_n(1)$.

Using this notation for the n-dimensional closed simplex, the Dirichlet distribution can be defined.

Definition A.0.2. *Let* $\boldsymbol{x} = (x_1, \ldots, x_n)^T \in \mathbb{T}_n$ *be a random vector.* $\boldsymbol{x}$ *is said to have a* ***Dirichlet distribution*** *if its probability density distribution is as follows:*

$$f(x) = \frac{\Gamma(\alpha_0)}{\prod_{i=1}^{n} \Gamma(\alpha_i)} \prod_{i=1}^{n} x_i^{\alpha_i - 1},$$

where $\alpha^n = (\alpha_1, \ldots, \alpha_n)^T$ *is a positive parameter vector,* $\alpha_0 = \sum_{i=1}^{n} \alpha_i$*, and* $\Gamma(c) = \int_0^\infty t^{c-1} e^{-t} dt$ *is the Gamma function. We denote the Dirichlet distribution as* $Dir(\alpha^n)$ *[55].*

It can be observed that when $n = 2$, the probability density function of the Dirichlet distribution corresponds to that of the Beta distribution with α_1 and α_2 as parameters. It is indeed a multivariate generalization of the Beta distribution, often being referred to as such. Figure A.1 visualizes the probability simplex, and how the parameter vector α^n impacts the distribution over the simplex.

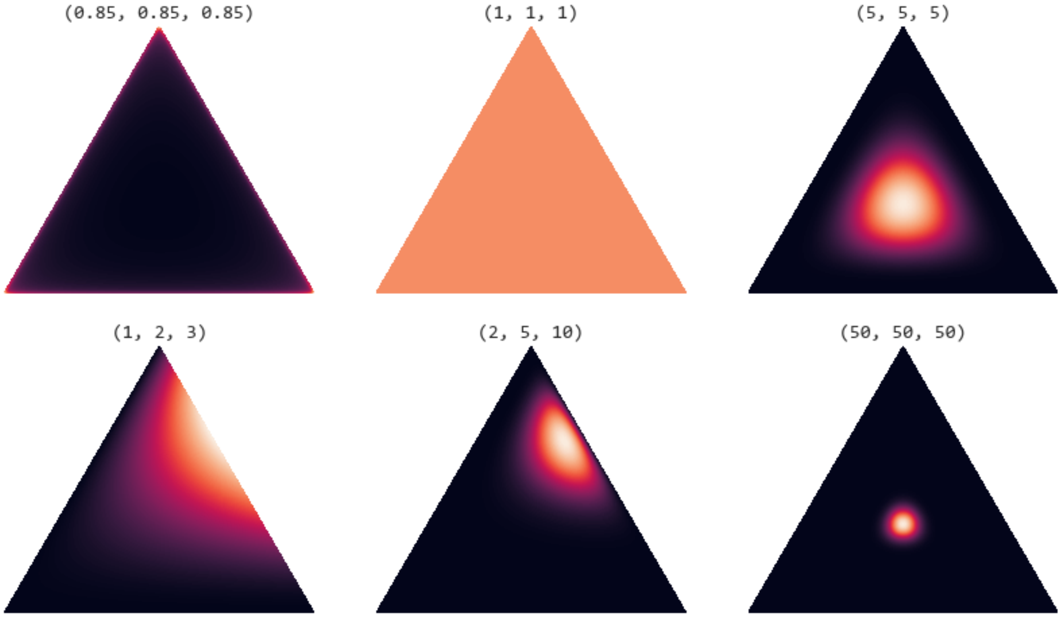


Figure A.1: The probability density functions of a Dirichlet distribution with three random variables, for different α^3 vectors. In general, the density function is a n-1 dimensional simplex that exists in the n-dimensional space. The α^3 parameter vector controls the probability congestion over the simplex, with every vertex representing a random variable. Code taken from `https://github.com/yusueliu/medium/blob/master/scripts/plot_dirichlet.py`.

From Figure A.1 there are some noteworthy observations. First, when every parameter α_i is equal to one another, the distribution over the n-dimensional closed simplex is symmetrically distributed.

Second, the magnitude of the parameters α_i impact the probability density shape. When $0 < \alpha_i < 1$, the probability density congests at the edges of the simplex. Increasing the value of α_i such that $\alpha_i = 1$, causes the density to become uniformly distributed over the simplex. Finally, when $\alpha_i > 1$, the probability density accumulates in the centre of the simplex.

B

Statistical Significance

B.1. Figure 4.7

	Assumptions			
	Homoscedasticity	**Normality**		
Data set		Binary	Ternary	**Student's *t*-test**
1	0.717	0.295	0.420	**7.26e-10**
2	6.34e-2	0.184	0.473	**7.61e-18**
3	0.124	0.772	0.111	**4.35e-24**
4	0.212	0.984	0.253	**5.72e-14**
5	0.423	0.670	6.72e-2	**1.69e-20**

Table B.1: The Student's *t*-test and the verification of its assumptions applied on the AUC outcomes of Figure 4.7. All reported values indicate the p-values.

	Assumptions			
	Homoscedasticity	**Normality**		
Data set		Binary	Ternary	**Student's *t*-test**
1	0.954	0.270	0.842	**7.74e-10**
2	**1.78e-2**	**4.40e-2**	0.895	**4.72e-14**
3	**4.73e-4**	0.769	**2.68e-2**	**2.44e-18**
4	0.236	0.176	0.905	**1.47e-15**
5	8.79e-2	0.937	0.556	**2.07e-17**

Table B.2: The Student's *t*-test and the verification of its assumptions applied on the AUPRC outcomes of Figure 4.7. All reported values indicate the p-values.

B.2. Figure 4.8

	Assumptions			
	Homoscedasticity	Normality		
Data set		Binary	Ternary	Student's t-test
1	0.326	**3.13e-2**	**4.29e-2**	0.611
2	0.666	0.198	0.460	**2.14e-4**
3	0.114	0.631	0.545	**1.13e-5**
4	0.912	0.837	0.402	**1.5e-6**
5	0.334	**3.40e-2**	0.817	**6.1e-9**

Table B.3: The Student's t-test and the verification of its assumptions applied on the AUC outcomes of Figure 4.8. All reported values indicate the p-values.

	Assumptions			
	Homoscedasticity	Normality		
Data set		Binary	Ternary	Student's t-test
1	**2.94e-3**	0.337	0.610	0.294
2	0.552	0.532	0.766	0.956
3	6.57e-2	0.889	0.136	0.649
4	**4.31e-2**	0.824	0.148	0.115
5	5.08e-2	0.858	0.499	5.46e-2

Table B.4: The Student's t-test and the verification of its assumptions applied on the AUPRC outcomes of Figure 4.8. All reported values indicate the p-values.

B.3. Figure 4.9

	Assumptions			
	Homoscedasticity	Normality		
Data set		Binary	Ternary	Student's t-test
1	0.178	0.541	0.183	**1.55e-9**
2	0.789	**1.68e-2**	0.281	**2.19e-9**
3	0.205	0.868	0.909	**1.77e-8**
4	0.574	0.555	0.109	**5.51e-8**
5	0.472	0.753	0.169	**1.47e-11**

Table B.5: The Student's t-test and the verification of its assumptions applied on the AUC outcomes of Figure 4.9. All reported values indicate the p-values.

	Assumptions			
	Homoscedasticity	Normality		
Data set		Binary	Ternary	Student's t-test
1	0.489	0.283	0.645	**5.78e-10**
2	0.974	6.94e-2	0.662	**7.08e-11**
3	0.635	0.641	0.829	**2.43e-9**
4	0.334	0.697	**1.72e-2**	**2.60e-12**
5	0.811	0.506	0.688	**4.94e-18**

Table B.6: The Student's t-test and the verification of its assumptions applied on the AUPRC outcomes of Figure 4.9. All reported values indicate the p-values.

B.4. Figure C.1

	Assumptions			
	Homoscedasticity	Normality		
Data set		Binary	Ternary	Student's t-test
1	0.588	0.960	0.968	0.607
2	0.801	0.747	0.952	0.571
3	0.185	0.243	0.827	**3.65e-2**
4	0.224	0.309	0.118	7.10e-2
5	0.944	0.450	0.629	0.538

Table B.7: The Student's t-test and the verification of its assumptions applied on the AUC outcomes of Figure C.1. All reported values indicate the p-values.

	Assumptions			
	Homoscedasticity	Normality		
Data set		Binary	Ternary	Student's t-test
1	0.402	0.413	0.880	0.946
2	0.817	0.841	0.256	0.497
3	0.449	0.984	0.748	0.113
4	0.149	0.488	0.679	7.96e-2
5	0.409	3.91e-2	0.882	0.435

Table B.8: The Student's t-test and the verification of its assumptions applied on the AUPRC outcomes of Figure C.1. All reported values indicate the p-values.

C

MI-Local-DIFFI Split Interval Length Indicator for Mixed-Attribute data

C.1. Split Interval Length Indicator

Recall from Section 3.3 that the split interval of a node v_{ij} in the path of anomaly o in tree t_i is defined as:

$$si\left(o, v_{ij}\right) := \frac{\left|\text{Anomaly split interval of } v_{ij} \text{ w.r.t } o\right|}{\left|\text{feature interval of } v_{ij}\right|}$$

The split interval weight is then defined as the vector $\overrightarrow{w}^{SI}(o,i) = \left(w_1^{SI}(o,i), \ldots w_{PL}^{SI}(o,i)\right)$, where PL represents $PL(o,i)$ defined above, where

$$w_j^{SI}(o,i) = 1.5 - \frac{1}{si\left(o, v_{ij}\right) + 1}$$

This split interval weight is dependent on the underlying feature distribution. Definitions of the anomaly split interval and the feature interval are all dependent on a notion of distance. The calculation of such a distance metric is not as straightforward when considering nominal attributes. However, only taking this weight into consideration for the numerical attributes, would induce a bias in the feature importance scores. Without considering the third weight only for nominal attributes, the numerical attributes are penalized more severely. Thus, in this thesis, it was first opted to discard the split interval weight when considering iForest$_{CS}$ entirely.

However, when testing the different indicators of MI-Local-DIFFI, it was determined that the split interval weight improved the explanation results of purely numerical data [4]. To strive for this improvement in results, a hybrid adaptation is proposed. By introducing an additional weight when considering the random sampling in a nominal attribute, the split interval weight can potentially be used again in the MI-Local-DIFFI approach applied to iForest$_{CS}$.

When considering nominal feature, it can be interesting to evaluate the cardinality of the nominal feature. Especially using a random sampling approach to construct the left and right child nodes, the number of categories that are sampled can provide skewed splits. For example, if one selects a single category for the left branch and assigns all remaining categories to the right branch, it is likely that the observations in the left branch are isolated more readily as a result. This will be reflected in the first two indicators of MI-Local-DIFFI already, so similarly to the original split interval weight a penalty is introduced to indicate the “luckiness" of the split in this particular nominal node.

Hence, it is proposed that for the nominal split interval of a node v_{ij} in the path of anomaly o in tree t_i is defined as:

$$si_{nom}\left(o, v_{ij}\right) := \frac{c}{k_j},$$

where $c \in [1, \lfloor \frac{k_j}{2} \rfloor]$ represents the number of categories sampled from the nominal feature selected in node v_{ij}, and k_j represents the total cardinality of the nominal feature selected in node v_{ij}.

C.2. Experiment with Synthetic Data

In this section, the data set presented in Subsection 4.3.2 is used to address the MI-Local-DIFFI method applied to mixed-attribute data. With the underlying assumption that the three attributes containing information regarding Height, Weight and Country all contribute to the detection of anomalies, these features are classified as the feature subspace containing anomalies. The data set is expanded with 10 additional features, all containing noise sampled independently from a Uniform distribution on the interval $[0,1]$. This is done to justify the anomalies located in the Height/Weight/Country subspace using the probabilistic anomaly measure. Furthermore, the size of the overall data set is increased from $n = 300$ to $n = 1000$, in line with the data sets used in Subsection 4.4.1. Finally, the cardinality of the nominal attribute is increased by incorporating country data from Germany and Indonesia. A Dirichlet prior is again used to automate the parameters of the Multinomial distribution.

This experiment specifically addresses the newly proposed split interval length indicator of MI-Local-DIFFI for nominal attributes, described in Section C.1. In the original proposal of MI-Local-DIFFI, a split interval length indicator was found to improve the explanation results compared to only utilising the path length and split proportion indicator. This split interval length indicator is not compatible with a nominal attribute. Thus, an adaptation is proposed that considers the cardinality of a nominal attribute instead, ensuring that the split interval indicator can be incorporated into MI-Local-DIFFI when using iForest$_{CS}$. The indicator therefore becomes a hybrid weight that adapts to the feature's attribute typology. In Figure C.1, MI-Local-DIFFI is applied to different data sets, with and without the hybrid split indicator weight:

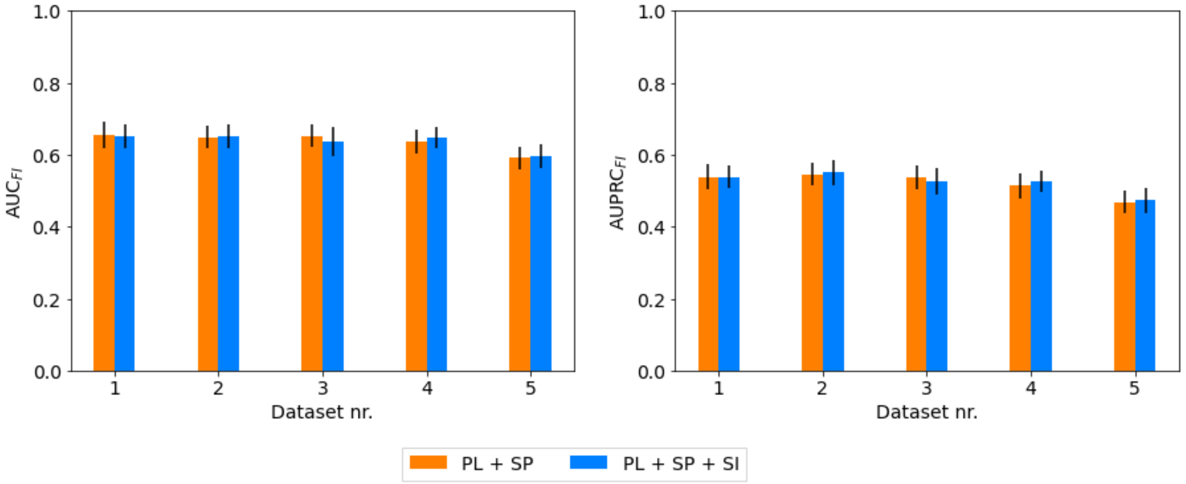


Figure C.1: The results of MI-Local-DIFFI applied to 5 different data sets with height and weight data conditioned on the nominal country feature. First, MI-Local-DIFFI is applied using only the Path Length (PL) and Split Proportion (SP) indicators. Then, the hybrid variant of the Split Interval Length Indicator (SI), is taken into consideration. Per data set, 50 runs have been conducted.

From Figure C.1, it is apparent that with these data sets there is no consistent statistically significant difference between MI-Local-DIFFI with and without the hybrid split indicator weight. This is also confirmed in Appendix B, where only the AUC_{FI} results of data set number 3 are found to have significant different means.

However, the intuition behind incorporating an indicator that penalizes overly fortunate splits is understood and valued. Therefore, it is believed that data sets can be constructed to improve the testing of this indicator. The assumption stated earlier that the height, weight and country all contribute to the detection of anomalies is true, yet does not incorporate the fact that anomalies can also be determined through the weight and height features only as well. Thus, this analysis can be improved by using data sets that incorporate nominal attributes and have clear anomaly containing sub-spaces of the data similar to the HiCS data sets. Hence, further research on this indicator is suggested.