Unsupervised Anomaly Detection of Healthcare Providers using Generative Adversarial Network

by

Krishnan Naidoo

Submitted in partial fulfilment of the requirements for the degree
Masters of Science (Computer Science)
in the Faculty of Engineering, Built Environment and Information Technology
University of Pretoria, Pretoria

January, 2021

Publication data:

Krishnan Naidoo. Unsupervised Anomaly Detection of Healthcare Providers using Generative Adversarial Networks. Masters thesis, University of Pretoria, Department of Computer Science, Pretoria, South Africa, December 2019.

Electronic, hyperlinked versions of this thesis are available online, as Adobe PDF files, at:

https://repository.up.ac.za

Unsupervised Anomaly Detection of Healthcare Providers using Generative Adversarial Networks

by

Krishnan Naidoo
E-mail: marionaidoo@gmail.com

Abstract

Healthcare fraud is considered a challenge for many societies. Healthcare funding that could be spent on medicine, care for the elderly or emergency room visits is instead lost to fraudulent activities by medical practitioners or patients. With rising healthcare costs, healthcare fraud is a major factor in increasing healthcare costs. This study evaluates previous anomaly detection machine learning models and proposes an unsupervised framework to identify anomalies using a Generative Adversarial Network (GAN) model. The GAN anomaly detection model was applied to two different healthcare provider data sets. The anomalous healthcare providers were further analysed through the application of classification models with the logistic regression and extreme gradient boosting models showing acceptable performances. Results from the SHapley Additive exPlanation also shows the predictors used to explain the anomalous healthcare providers.

Keywords: Generative Adversarial Network (GAN), Anomaly Detection, Healthcare Providers, Machine Learning, Deep Learning, SHapley Additive exPlanation.

Supervisors **:** Dr. V. Mariate
Department **:** Department of Computer Science
Degree **:** Master of Science

Acknowledgements

I wish to extend my gratitude to the following organisations and individuals:

- Dr Vukosi Mariate, my supervisor, who supported and guided my research, was always generous with constructive criticism, provided a platform to collaborate with other researchers, and always pushed me to explore the unknown.
- Krystal Naidoo, my wife, who embarked on this roller-coaster journey with me, witnessed all the ups and downs, proofread my research many times and always pushed me to see my research through to the end.
- Members of the Data Science for Social Impact group, whose monthly discussions and interactions were intellectually stimulating, and the members very supportive over the past months.
- Lastly, to ABSA for their financial support through the ABSA UP Data Science Chair. This has aided me in completing my thesis without having to carry the financial burden.

It has been a long, exciting and knowledge-acquisitive journey. I am glad I did not have to go through this it alone.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background Information

In 2016, the global spend on health was US$ 7.5 trillion, representing close to 10% of global GDP [82]. Studies across Europe estimate around 30% has been lost to wasteful spending [50]. A study in 2016 by [78] confirms the financial value of fraud cases in Europe. France leading the way with a loss of (€ 46.3M) relating to fraud committed by 37% was healthcare practitioners. 27% was committed by health facilities, and approximately 20% by insured persons. The Netherlands' fraudulent activity accounted for (€ 18.7) million and was mostly related to wrongful billings. The UK followed closely at (€ 11.9) million related to fraud, bribery and corruption in this industry. Furthermore, healthcare fraud in the United States ranged from US$80 billion to US$200 billion [58] relating to improper coding, phantom billing, kickback schemes and wrong diagnoses being some of the reasons.

In Africa, there is a lack of systems and strong financial processes which contribute to its healthcare fraud [61]. Reports done in South Africa suggest that approximately 3-4% of the R160 billion medical industry relates to fraudulent claims and abusive or wasteful healthcare costs in South Africa [25, 51]. Despite numerous attempts to solve the problem, the detection of these fraudulent activities within the healthcare sector remains a challenge due to poor data quality or lack of data [56, 74].

Healthcare fraud falls within the domain of anomaly detection or outlier detection

[43, 23, 27] which is the identification of data that deviates from normal behaviour or trends [87, 45, 91]. Some of the common, unsupervised anomaly detection in the healthcare sector relates to differing methods of detection: distance-based methods [42, 29], isolation methods [29, 42, 70], domain-based methods [57, 29, 89, 10] and neighbour-based methods [29, 43, 36].

In the domain of unsupervised anomaly detection using deep learning algorithms, the research by Schlegl [68] first introduced the use of a generative adversarial networks (GANs) to solve the anomaly detection problem (AnoGAN) on medical imaging data [68]. The authors proposed an anomaly score which measures the difference between the actual image and the reconstruction image. Subsequently, the authors proposed f-AnoGAN (Fast Anomaly Detection GAN) and improved the initial AnoGAN research by substituting the Deep Convolutional GAN architecture (DCGAN) [63] with a Wasserstein GAN (WGAN) [8], and furthermore introducing an encoder during the training [67]. Research by Deecke [26] proposed an anomaly detection GANs (ADGAN) that showed improved results when compared to AnoGAN and other unsupervised anomaly detection models. The ADGAN algorithm searches the latent space from multiple different areas for the closest possible match [26]. The ADGAN also discards the discriminator after training which allows the ADGAN to be coupled with other generative networks such as variational autoencoders (VAEs) [26].

Previous studies used unsupervised machine learning [29, 74, 10, 70] and deep learning approaches [73, 1, 13, 84] to solve anomaly/fraud detection problems. However, these researches have predominately focused on image problems, with limited application within the health care domain.Furthermore, these studies used a black-box approach to detect anomalies, i.e., there is little or no understanding on how the prediction was made or the features that is contributing to the prediction [14]. With this shortcoming, it is challenging to convince domain experts to trust and adopt deep learning approaches and algorithms [14]. Deep learning models need to provide an explanation for each instance as opposed to a model explanation [6]. Providing context to classifiers will improve the trust domain expert's have in the algorithm's output [6].

GANs is a deep learning model used in an unsupervised and semi-supervised learning environment for problems where there is an imbalance of anomalous labels [68, 8, 26, 84].

Not only does the GANs algorithm able to detect both fraudulent activities [47] and malicious users [84], they have been used to augment minority classifiers that solve the classification between fraudulent and normal samples [81, 68, 8, 26, 84]. This this study suggests how to identify anomalies without labels and explains the anomalous healthcare providers (HCPs).

1.2 Problem Statement

The essential components of a successful healthcare system revolves around two main pillars, competent HCPs and a stable financial infrastructure [10]. These pillars can be susceptible to medical cost fraud, waste and abuse [10, 43]. By analysing information collected from these systems, it is possible to detect anomalies through the use of predictive models [43]. Furthermore, data collected that are related to behavioural interactions with the system, it is possible to identify outlying users with malicious intent [24, 91, 34]. As a result, anomaly detection has become a pertinent subject in domains such as fraud detection [43, 29, 74, 10, 70] and intrusion detection [29, 45].Previous research suggest that using various machine learning techniques can solve the fraud detection problem with some of the best results being achieved by supervised learning [29, 74, 10, 70]. However, supervised anomaly detection requires labelled data representing both normal and unusual behavior to train a model [29]; obtaining labelled data is often time consuming and costly [56, 74]. In some studies, labels would have to be manually produced or augmented from other data sources [10] before the training process begins. This approach to obtaining labels can be a time-consuming and costly exercise [55] if there are huge volumes of records.

In this thesis, we present an approach by combining unsupervised and supervised learning to solve the problem of not having labelled data and the lack of interpretability with deep learning models. The approach is to use unsupervised GANs model to label a HCP fraudulent, or not. Based on this label, a new labelled dataset was created to train supervised classification models to explain the features contributing to the anomalies.

1.3 Research Objectives

The first objective is to examine and evaluate the fraudulent activities and cost abuses by HCPs. The second objective is to build a model across the various HCP types to predict if a provider is fraudulent or not. It is vital to detect fraudulent activities and cost abuse before the transaction is registered because fraud can incur significant costs [40, 82, 78]. However, building a predictive fraud model can be challenging due to the lack of labelled data [34]. This study uses a public data set from *Medicare Provider Utilisation and Payment Data: Physician and Other Supplier* [25], and another private dataset from a *South African claims administrator organisation.*

To achieve these objectives, the following research guidelines were used:

- To understand the factors that contribute to healthcare fraud and cost abuse.
- To establish a model for predicting healthcare fraud and resultant cost abuse without labels.
- To quantify these contributing factors that influence healthcare fraud and cost abuse.

The implementation and outcome of this study may assist organisations to overcome the fraud detection challenge highlighted in Section 1.1. Analysing HCP's payment and claim behavioural patterns may enable insurance companies to detect fraudulent transactions in real-time with a high degree of accuracy. Furthermore, the implementation of this anomaly detection model will allow the ability to stop payments to anomalous transactions and thereby significantly reduce costs.

1.4 Research Questions

The research questions listed below were addressed in this study:

- Can deep learning algorithms detect HCPs relating to fraud and cost abuse without having labels?
- Can machine learning models interpret the reasons for anomaly detection?

- Which features explain the reasons for the anomalous HCPs?

1.5 Contributions

Motivated by recent deep learning research, this study [1] proposes an anomaly detection approach using a generative adversarial training framework [73, 1, 10, 40, 84, 68]. Furthermore, similar to Zenati's [84], we use non-image unlabelled HCP data as the inputs for training examples [84].

However, in addition to [73, 1, 10, 40, 84], our two-step approach is both the use of a GANs algorithm to identify labels and use of classification and SHAP algorithms to provider reasons to anomalous HCPs. The contributions of this thesis are summarised as follows:

- a unsupervised GANs model was designed to detect anomalies across the various HCPs;
- an anomaly score calculated by the discriminator and the generator losses are used to detect anomalies.

1.6 Limitations

The focus of this study is to create an approach on how to detect and give insights into the anomalous transactions received by HCPs. The following limitations are set for this thesis:

- The classification algorithms used in this thesis only classified one of two classes as a target variable. To answer the current research questions, these classes were identified as 1 representing anomaly and 0 representing normal.

[1] The contents of this research have been published as an article in the 19th IFIP WG 6.11 Conference on e-Business, e-Services, and e-Society [53]

- A Generative Adversarial Network (GAN) algorithm was used for unsupervised learning for anomaly detection. This limitations was partly set due to time constraints and related work compared with other GANs methods.
- The configuration of the hyperparameters for the GAN model was guided by a previous study by Zenati [84] to minimise the time spent on model optimisation.

1.7 Thesis Outline

The rest of the thesis is structured as follows:

- **Chapter 1** provides the background information, highlighting the context of healthcare fraud and machine learning, details of the problem statement, research objectives and questions, and the relevance and contributions of this thesis.
- **Chapter 2** introduces the previous literature on HCP cost abuse, discusses the various anomaly detection techniques and anomaly score.
- **Chapter 3** presents our proposed methodology highlighting the GANs architecture, anomaly score function, algorithms, and evaluation metrics.
- **Chapter 4** presents the data sets used in the experiments, data pre-processing, and preliminary data analysis.
- **Chapter 5** discusses the results and implications based on the research questions.
- **Chapter 6** summarises the paper and proposes possible future work.

1.8 Summary

This chapter added context to the healthcare economic landscape and a backdrop to the research problem. It highlighted the purpose and the research problem. It also showed some of the benefits for the organisations. Furthermore, it explored how deep learning models could serve as levers for to overcome these challenges. To this end, it was suggested that the model to be built to identify fraudulent HCPs.

Chapter 2

Literature Review

2.1 Introduction

This chapter focuses on the literature review of healthcare abuse and costs incurred wastage that is relevant to the current research study. Takeaways from previous academic studies are covered and discuss the different supervised and unsupervised models that underpin the study. Thereafter, we discuss in detail the GAN algorithm, challenges, as well as the SHAP algorithm which aims to give context to the anomalous HCP.

2.2 Healthcare Fraud and Cost Abuse

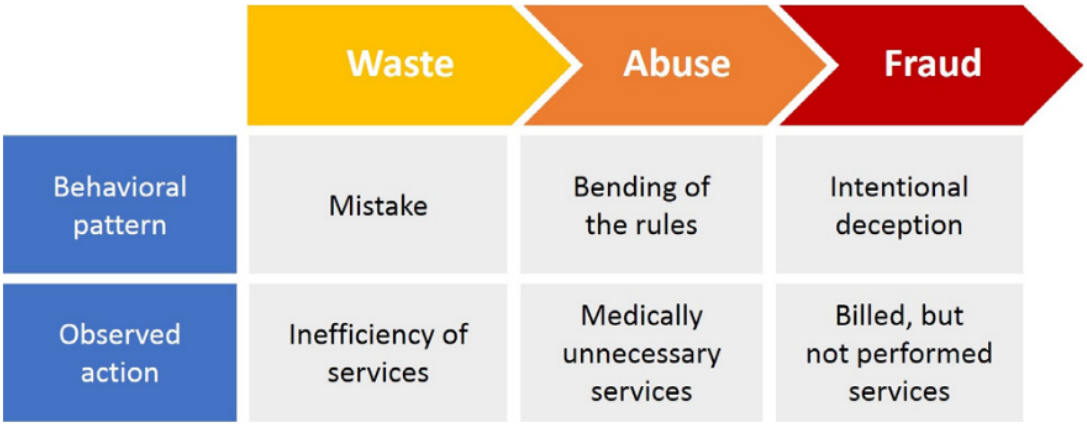
The definitions of medical cost abuse and fraud are heterogeneous in various previous literature, and depend on market and regulatory environments [11]. The definition of fraud was used from [74] describing fraud as ”the intentional deception or misrepresentation that an individual knows to be false and knowing that the deception could result in some unauthorized benefit to himself/herself or some other person”. Inappropriate payments by insurance organisations or HCPs occur as a result of fraud, abuse or error. The study by [43] describes HCP abuse as either directly or indirectly providing a service, resulting in unnecessary costs to the insurance organisation or patient. Abuse also relates to any HCP that is not consistent in providing patients with necessary medical services, does not meet professionally-recognised standards, and is not fairly priced [74, 25].

According to Bayerstadler [11] shown in Figure 2.1, fraud and abuse can be classified into three behavioral categories:

1. Services not performed (fraud): Medical services which are not rendered to the patient, but documented and charged for. For example, a healthcare provider invoicing a patient for stitching of a finger which was never carried out on the patient.

2. Services not required (abuse): Medical services which deviate from medical best practice, and performed without medical justification. For example, the prescription of bandages for a mild nasal congestion.

3. Other billing issues (fraud/abuse): Additional to services not performed or required, are intentional misbehaviour by HCPs and/or policyholders. For example, the unbundling of procedure codes. Thornton refers to unbundling in this context as ”the billing of each stage of a procedure as if it were a separate treatment” [74].



	Waste	Abuse	Fraud
Behavioral pattern	Mistake	Bending of the rules	Intentional deception
Observed action	Inefficiency of services	Medically unnecessary services	Billed, but not performed services

Figure 2.1: Behavioral patterns of fraud and abuse in healthcare

Fraud and abuse within healthcare systems are attributed to the involvement of three major stakeholders. First, healthcare service providers which includes doctors, hospitals, ambulance services, and medical laboratories across public and private health departments [74]. Secondly, insurance policyholders including patients and patients’ employers

[74]. Finally, insurance organisations which receive premiums from policyholders and are responsible for payment of costs to HCP [74, 11].

2.3 Types of healthcare provider fraud and cost abuse

A recent review of studies that discussed the major types of fraud committed by healthcare service providers [46, 74]. Improper coding, unbundling, double billing, billing for services not rendered, providing unnecessary care, using incorrect diagnosis code or billing for services delivered by unqualified medical workers [46, 74, 44] were the most commonly investigated types of fraud among HCPs.

Improper coding, sometimes referred to as upcoding, is a common occurring fraud type among HCPs [74]. Thornton [74] describes upcoding as ”billing for a service or procedure that is more expensive than the one performed”. Improper coding is often explained as an administrative error as opposed to the malicious attempts to increase revenue [74, 44]. Further to improper coding, is a concept of unbundling whereby creating separate billing line items for supplies, treatments or services instead of grouping them together [46]. In Orgunbanjo [59] study, unbundling is categorised as improper coding [59] whereas other studies views unbundling as a different type of fraud [74, 44].

Double billing is another key theme evident in previous studies [74, 59]. This typically involves the HCP submitting the same invoice multiple times for payment. Often, the separate invoices submitted relate to the same procedure performed. Thornton [74] further defines ”double-billing as billing multiple times for the same service”. The majority of insurance organisations implement automatic processes to accept and pay claims to improve processing speed and customer service [48]. Implementation of these process efficiencies could possibly miss ”double-billed” claims, however, researchers like Banarescu [17] suggest that the true efficiency of claim-processing system depends on speed and legitimacy. In cases relating to double-billing, the relevant care or service is provided to a patient. Additionally, there are instances where HCP invoices for medical services not rendered, for medication not received or medical devices that the patient has no knowledge of receiving [74]. Some authors also refer to this concept as ”phantom billing” [17, 74, 44].

Examples mentioned by Thornton [74] and Kirlidog [44] note that an HCP submitting a high volume of claims on a single day is suspicious. They say it is not normal for a HCP to see unrealistically excessive numbers of patients on a single day. Furthermore, Thornton [74] describes a new trend that use ghost employees. This type of excessive billing relates to employees who are no longer employed with the healthcare provider but are billing for services rendered. The researcher introduced multi-dimensional data models centred around the HCP and their groups, respectively [74]. These models were used to highlight excessive billing.

In organisations where the majority of the claims are automatically processed [17], gives rise to another type of excessive billing approach introduced by Banarescu [17]. This type is the submission of false claims by HCPs by exploiting known loopholes within the claims payment process that go undetected [17]. A recent study also highlighted other examples of false claims, such as a submission of a claims for a patients who are deceased [74]. Additionally, Thornton's research [74] indicated that providers submit claims for medication after the patient had died [74]. These claims are submitted for payment based on falsifying the the patients diagnosis [17]. One of the contributing factors why this type of fraud is carried out, is to falsely prescribe certain medications for a patient [17].

2.4 Anomaly Detection

Anomalies are patterns in data that do not conform to expected or normal behaviour [45]. Depending on the application domains, these deviating patterns are often referred to as "outliers, anomalies, containments, exceptions, aberrations, peculiarities, or discordant observations" [22]. In the context of anomaly detection, anomalies and outliers are the two most common terms and are often used interchangeably [22, 87, 91, 45]. Anomaly detection techniques are used across many different areas such as early fault detection in production systems, fraudulent transactions on credit cards, security screening for border security, and intrusion detection against cyber attackers [22, 44, 18, 55].

In the remaining parts of this study, we will use anomaly detection as a term that relates to both fraud and medical cost abuse constructs [87, 91]. Usually, if labelled data is available, a supervised anomaly detection approach may be used [87]. Datasets are considered as labelled if both the normal and anomalous data points have been recorded [91, 87]. When labels are not recorded or available, the alternative option is an unsupervised anomaly detection approach [91].

Anomaly detection is divided into three main approaches:

- **Supervised Anomaly Detection:** This approach involves training a machine learning model, using data instances containing both normal and anomalous labels [57]. For example, supervised classification models created as a binary classifier assists in detecting normal and fraudulent healthcare transactions [57, 43].

- **Semi-supervised Anomaly Detection:** The training set only includes only examples from normal patterns. In many healthcare applications, it is possible to collect data reflecting the normal pattern of HCP payments. However, access to anomalous observations is often difficult, not recorded or labelled. Semi-supervised models build a normal profile from the available data and the model reject data points that deviate from the normal profile [60].

- **Unsupervised Anomaly Detection:** Compared to supervised anomaly detection, the training data does not contain any anomalous labels [29]. Recent unsupervised anomaly detection research suggested using kernel-based [29], clustering-based [20], distance-based [29], density-based [12] and reconstruction error based methods [20].

Due to the lack of labelled data, semi-supervised or unsupervised methods are the preferred options as opposed to supervised anomaly detection methods [13]. Furthermore, the performance of supervised anomaly detection models is sub-optimal due to the imbalance of classes [15]. Generally, the imbalance of classes is referred to when the total number of normal instances is greater than the total number of anomalous instances [55].

To address these challenges, unsupervised anomaly detection approaches are used to create normal and anomalous labels for each data sample since labelled data is either unavailable or difficult to obtain [20]. In areas such as medical image and clinical record analysis [20, 22], variants of unsupervised deep learning algorithms have shown to have better performance than traditional machine learning methods such as support vector machine (SVM) [55], principal component analysis (PCA) [4], and Isolation Forest (iForest) [29].

Below, we will discuss the different unsupervised anomaly detection models: kernel-based, clustering-based, reconstruction error based, distance-based, density based and isolation based.

2.4.1 Distance-Based Methods

This anomaly detection method use distance measures to identify anomalies. One commonly applied example is the Mahalanobis distance for anomaly detection problems across a multivariate dataset [22]. Similar to the one component Gaussian mixture model (GMM) [29], the Mahalanobis distance parameters are the inverse covariance matrix and mean from a given dataset [29].

Another distance-based example is a technique called the Local Distance Based Outlier Factor (LDOF) measure that attempts to find local anomalies [22]. The LDOF measure is the ratio of the average distance between each data point and its K-nearest neighbours. The main objective of the LDOF algorithm is to measure the extent of deviation in each instance from its K nearest neighbours instead of the whole dataset [22]. The choice of the K parameter is an important challenge especially for the detection of cluster anomalies [22].

2.4.2 Kernel-Based Methods

Kernel-based methods computes a hyperplane in a high-dimensional space separating points from the input space [29]. One such example of kernel-based method is One-class support vector machines (OCSVM). OCSVM are one of the common applied semi-supervised anomaly detection techniques due to their strong theoretical foundations [29]. OCSVM is an outlier detection that constructs a linear boundary separating the input data from the rest of the high dimensional space [29]. Normally, outliers identified by OCSVM is any data point found outside the linear boundary. OCSVM utilises memory efficiently on small subsets of the training dataset. However, some of the OCSVM limitations relate to [34]:

- The non-linear training complexity, which limits its application on large datasets
- Calibration is challenging due to its sensitivity to the parameter and kernel bandwidth.

2.4.3 Clustering-Based Methods

Anomaly detection using clustering algorithms can be performed by calculating the distance between every instance and its nearest cluster centroid [62]. The distance is often used to calculate its anomaly score [29]. Additionally, a threshold on the density of the detected clusters can also be defined by labelling sparse clusters as anomalies [90, 62]. Since the main aim of clustering is different from anomaly detection, many cluster based anomaly detections like the K-Means are not optimised for anomaly detection, especially for large and high-dimensional datasets [22].

A study by Zhu [90] reviewed two studies where K-Means was combined with other models to solve anomaly detection problems. First, the use of the K-Means clustering model to identify outliers, followed by classification of diabetic patients using the K-nearest neighbor (KNN) [90]. Secondly, an ensemble classifier model that combined the K-Means and decision tree algorithms that predicts diabetes [90]. The details of the K-Means algorithm concerning its anomaly detection application in Section 3.3.2 will be discussed.

2.4.4 Density-Based Methods

One of the common density based method used to solve anomaly detection problems is the Local Outlier Factor (LOF) [36, 29, 43, 55]. LOF generates a score for each observation in a dataset which is often referred to as the anomaly score [22]. The anomaly score is the variance between each data sample and its local neighbours [22].

For normal samples, the expectation is to have LOF scores close to one another, because of their similarity to their neighbouring normal samples. Anomalies, however, will have larger values, as their local reachability distance is much higher than their neighbouring normal samples [22]. The performance of the LOF model depends on the selection of a K parameter and an appropriate threshold for the anomaly score [29].

There are three main limitations of the LOF model [33]. First, LOF does not generate an explicit model that can be applied to future test data. Secondly, anomalies that have a similar density to their neighbouring points may not be detected. Lastly, the central points in a cluster of anomalies may not be detected.

2.4.5 Isolation-Based Methods

Anomalies can be identified by a technique called isolation, which assumes that they can be isolated from normal data if the majority of instances in a training set are normal [33]. A well-known algorithm used for this specific purpose is the iForest [29]. The iForest algorithm repeatedly split feature values randomly, thus resulting in isolating each data point from the remaining observations[29]. As part of the iForest algorithm, an isolation score is calculated using random forests. Domingues defines the isolation score for each data point as “the average path distance between the root of the tree and the node containing the specific data point” [29]. Generally, these anomalies are rare and different, hence they would be isolated sooner than normal instances. This means anomalous data points will have a shorter depth from the root of an iForest compared with normal data points. The iForest algorithm demonstrates significantly better outlier detection performance and linear time complexity when compared to the LOF algorithm [33].

2.4.6 Reconstruction Error Methods

Reconstruction-based methods commonly use neural networks to model data so that the data can be reconstructed from the model [29]. Initial deep learning methods either use traditional outlier detection methods on the learned encoding [2, 86, 54, 15], or directly use the reconstruction error of test points [7, 28, 54]. More recent deep learning studies have additional losses but are still fundamentally based on reconstruction methods [20, 80, 68, 84]. The reconstruction error measures the deviation between the test data and the dimensional reconstructed output [87, 21]. In scenarios where the training data only contains normal behaviour data points, the model will fail to reconstruct any test data that deviates from its training set. Thus, the reconstruction error for anomalies should be greater when compared with normal data points. Table 2.1 shows some anomaly detection studies that used reconstruction error to solve anomaly detection problems.

Generative Adversarial Networks (GANs) [37] is another deep learning model that makes use of the reconstruction-based approach. GANs have received much attention due to its better performance and fewer restrictions for generator functions when compared to deep generative models such as Boltzmann machines and autoencoders [26].

2.5 Generative Adversarial Network

In Goodfellow's [37] original formulation, the Generative Adversarial Networks (GANs) framework consists of two neural networks. The first neural network is a generator that generates samples from latent distribution and the second neural network is a discriminator that aims to differentiate between real and generated samples [37]. The GANs framework [37] have been successfully applied to high-dimensional and complex data distributions [68, 63, 84, 1]. The GANs characteristic suggests they can be used successfully for anomaly detection, although their application has only been recently explored. A common GANs unsupervised anomaly detection approach is using only normal behavior samples in the training process and an anomaly score for the detection of anomalies [68].

The GANs framework presented in Figure 2.2 consists of two neural networks that are trained simultaneously and each network competes against each other in a zero-sum game [37]. The first neural network is the generative model G, that gets vectors of noise (z) as input and maps z to a latent space from the input data distribution. The second neural network is the discriminator model D, that determines if the sample comes from G or the original dataset. The objective of the generative network is to deceive D into believing that the generated data comes from the actual input data distribution (x) [37].

In a review of recent GANs studies, the majority of the GAN-based anomaly detection approaches are built upon on the Adversarial Feature Learning idea [30] or using convolutional generative adversarial networks (DCGANs) [63]. The Bidirectional GAN (BiGAN) architecture extended the original GAN architecture to incorporate an autoencoder (AE), adding to the learning by inverse mapping [30]. The new autoencoder learned function maps the input data to a new dimensional data space, together with

Table 2.1: Examples of studies using reconstruction error for anomaly detection

Study	**Overview**	**Reference**
Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection (DAGMM)	DAGMM uses an autoencoder with a reconstruction error to detect anomalies. This approach uses an autoencoder to generate a latent space, reconstruct features, and thereafter fed into a Gaussian Mixture Model (GMM).	[13]
Anomaly Detection using Autoencoders in High Performance Computing Systems	The research proposes an approach for anomaly detection using an autoencoder. The key proposition is to use a set of autoencoders in training phase to learn the normal behaviour of the supercomputer nodes. During testing on normal and abnormal data, the model identifies abnormal conditions through a higher reconstruction error.	[15]
adVAE: A self-adversarial variational autoencoder with Gaussian anomaly prior knowledge for anomaly detection	The model makes use of a Gaussian prior assumption and an anomaly score for the detection of anomalies.	[80]
A Comparison Study of Credit Card Fraud Detection: Supervised versus Unsupervised	The study applied deep learning models such as AE and GANs on data without labels. The authors used reconstruction errors to detect anomalies.	[57]

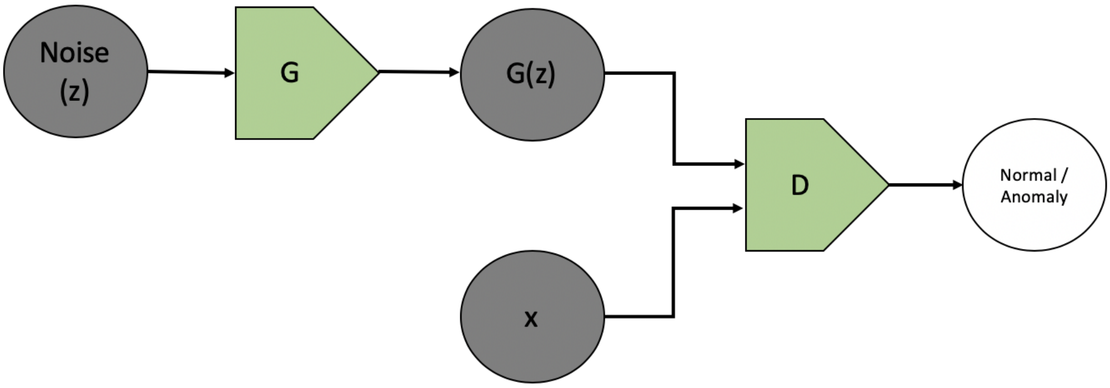


Figure 2.2: Structure of GANs

the output from generator function, is the basis of the anomaly detection by the discriminator. Table 2.2 describes and summarises some of the GAN-based approaches used to solve anomaly detection problems.

2.5.1 GAN Activation Functions

An activation function is part a neural network that assists in learning complex patterns. The objective of an activation function is to determine if a neuron should be activated or not by calculating the weighted sum and further adding bias to it [75]. Furthermore, the activation function introduces non-linearity to a neutron output [16]. Activation functions need to be partially differentiable, therefore, a smooth approximation such as the sigmoid function [16] is one of the activation functions used to solve deep learning anomaly detection problems [84, 68]. The sigmoid activation function is defined as:

$$Sigmoid(x) = \frac{1}{1 + e^{-net}}. \tag{2.1}$$

The output of the sigmoid function is within a continuous range between 0 and 1. The left panel of Figure 2.3 shows the graph of the sigmoid function and illustrates that the sigmoid exhibits linear behaviour around the origin, and saturates for large positive and negative input values.

Table 2.2: GAN-based approaches for anomaly detection

Algorithm	**Description**	**Overview**	**References**
AnoGAN	Anomaly GAN	AnoGAN uses a standard GAN, trained only on positive samples which learns a mapping from the latent space representation from input data and uses this learned representation to map new, unseen, samples back to the latent space.	[68, 67, 72, 84, 27]
FenceGAN	Fence GAN	In the work of FenceGAN both generator and discriminator loss functions are modified using the Encirclement Loss and Dispersion Loss respectively to solve anomlay detection problems.	[12, 72]
OCGAN	One Class GAN	OCGAN is a one-class novelty detection algorithm that learnt latent representations by applying a denoising autoencoder. The main contributions is to explicitly constrain the latent data space to represent a given class.	[76, 72]
BiGAN	Bidirectional GAN	BiGAN extends the GANs framework to include an encoder that learns the inverse of the generator. During training, the BiGAN model allows learning by simultaneously mapping data to the latent space and vice versa.	[30, 72, 27]
GANomaly	GAN Anomaly	GANomaly trains a generator network on normal samples to learn their manifold while at the same time an autoencoder is trained to learn how to encode input images in their latent representation efficiently. The GANomaly approach intends to improve ideas from previous GAN approaches such as AnoGAN, BiGAN and EGBAD	[1, 72, 27]
WGAN	Wasserstein GAN	Anomaly detection research using a WGAN because firstly WGAN does not collapse contrarily and secondly the use of a loss function to evaluate convergence.	[39, 8, 38]

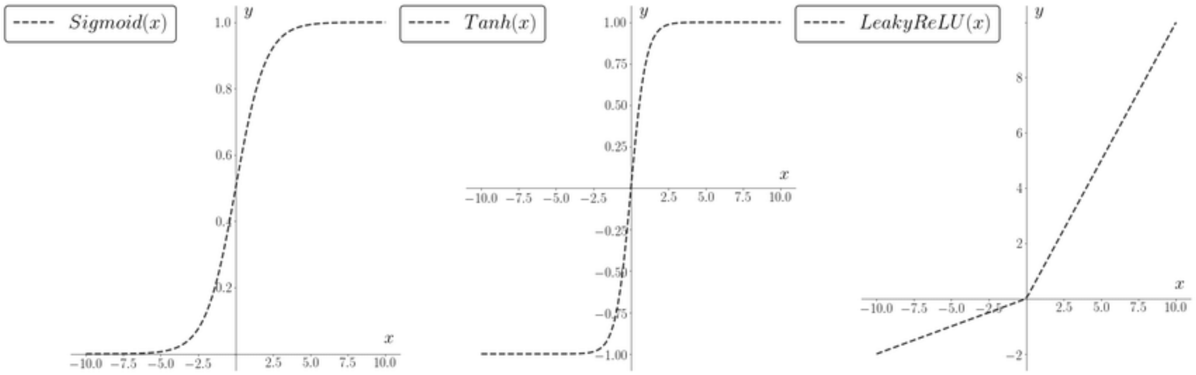


Figure 2.3: Activation functions

The sigmoid function is constrained to the positive range of (0;1) and consistent positive signals are likely to cause saturation within the hidden neurons [16]. To reduce the saturation problem, the hyperbolic tangent function (TanH), can be used as a substitute for the sigmoid. The TanH activation function is defined as:

$$TanH(x) = \frac{e^{net} - e^{-net}}{e^{net} + e^{-net}}. \tag{2.2}$$

The TanH output is in the range (-1; 1) and is thus centred at zero. Figure 2.3 shows that TanH has a similar s-like shape to sigmoid, and approaches asymptotes at -1 and 1. However, since the function is bounded, the possibility of saturation still exists, but the zero-centred range makes saturation less likely [16].

Figure 2.4 illustrates that both the sigmoid and the TanH activations exhibit small derivatives due to their asymptotic behaviour. Generally in neural networks, the backpropogation is defined recursively which introduces the vanishing gradient problem [75]. In deep neural networks, the vanishing gradient problem results in difficultly during training and ineffective [16]. To overcome the vanishing gradients issue, the rectified linear activation (Relu) function, was proposed as an alternative to the sigmoid and TanH activations [16]. The ReLU activation function is defined as

$$ReLU(x) = \begin{cases} net, & \text{if} \quad net > 0 \\ 0, & \text{otherwise} \end{cases} . \tag{2.3}$$

Figure 2.3 illustrates that the ReLU activation is simply the identity function for all

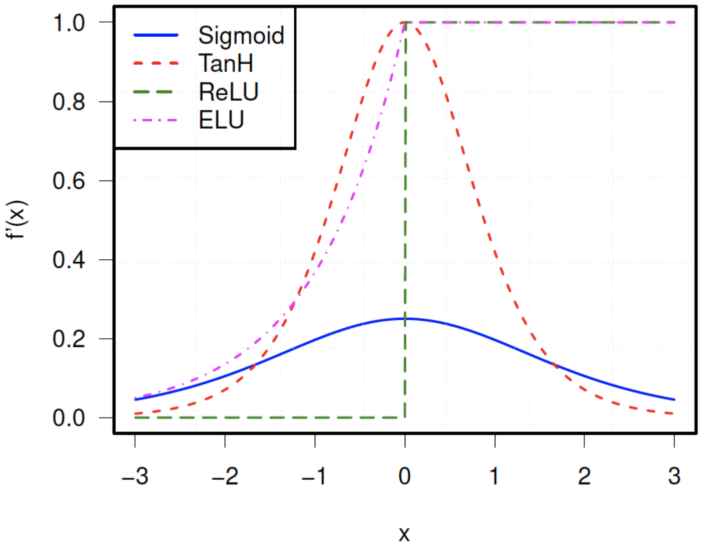


Figure 2.4: Activation functions

positive input signal values and saturates only for negative input signals. Recent studies suggests that ReLu is a preferred activation function in deep learning networks [16].

GANs architectures in previous studies suggests using the ReLU activation function in the generators hidden layers and using the TanH function for the output layer[63, 68, 84]. Radford suggested that using an ReLu activation function allowed the model to learn to saturate the training distribution much faster [63]. Furthermore, Radford also indicated that the leaky rectified activation together with the discriminator worked well, especially on images that require higher resolution modelling. Research by Schlegl [68] and Zenati [84] applied Radford's GAN activation function configuration to solve the anomaly detection problem on X-ray images and a similar activation setup to detect anomalies across non-image data.

2.5.2 GANs Challenges

Recent studies of GANs have shown that they can be used in various domains solving problems such as image generation, improving the resolutions of images, facial attribute manipulation, object detection and many more [63, 68, 2]. One of the main objective when training GANs is finding a Nash equilibrium especially with continuous, high-dimensional parameters [65]. Goodfellow defines the Nash equilibrium “as a state in a non-cooperative game whereby no player can improve its score except by changing their own strategy” [65]. With reference to the structure of GANs highlighted in Figure 2.2, the Nash equilibrium is the state between the generator and the discriminator. The function of the discriminator is to determine between data sampled from x and data sampled from $G(z)$. Typically, GANs are trained using gradient-descent techniques to find the lowest value of a cost function [63, 65, 72], instead of finding the Nash equilibrium. Thus, seeking for a Nash equilibrium results in these algorithms failing to converge [65].

Similar to neural networks, GANs models are considered black-box models [71], meaning it is challenging to understand how the model made a regression or classification decision [14]. Using these models in business applications, it is pivotal to understand and give context to anomalies [14]. Giving context to anomalies creates user trust, provides insight into future model improvements, and provides context of the features that are contributing to the anomalies [49]. In some applications, linear models are often preferred due to their ease of interpretation, and may be less accurate when compared to complex models [49]. However, due to the rise and availability of big data, there have been an increase in the application of deep learning models [5]. Therefore, deep learning models such as GANs are bringing the trade-off between model accuracy and interpretability to the forefront. Interpretability methods such as SHAP (SHapley Additive exPlanation) [49] and DeepLift [71] have been recently proposed to address this issue in deep learning models.

2.6 SHapley Additive exPlanation

Lundberg and Lee [49] introduced Shapley Additive exPlanations (SHAP), which is used to interpret model outputs. The SHAP framework is based on local explanations, game theory and estimates the contribution each feature have on the prediction [49]. SHAP values is a conditional expectation function that measures feature importance [49]. SHAP values relates to the change of each feature in the expected model prediction when conditioning on that feature. SHAP explains how to get from the base value

$E[f(z)]$ that would be predicted in the absence of any features, to the current output $f(x)$. Figure 2.5 shows a single ordering which represents the blue arrows as increasing the prediction and the red arrow decreasing the prediction [49]. For non-linear models or input features that are not independent, the sequence of how features are added to the expectation is important cause the SHAP values are calculated from averaging the ϕ_i values across all possible orderings.

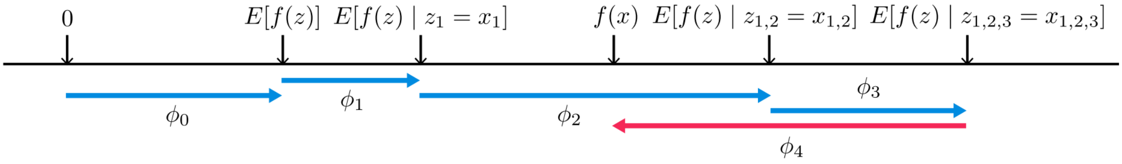


Figure 2.5: SHAP values

SHAP values also provide the feature importance measure that incorporate properties such as local accuracy, missingness, and consistency through the use of conditional expectations [49]. Lundberg categorised SHAP into 2 broad methods, namely the two model-agnostic approximation methods. Model-agnostic approximation methods refer to either the Shapley sampling values or the kernel SHAP. These two methods feature independence and model linearity is assumed when approximating conditional expectations [49]. Additional to SHAP are the model-specific methods which relate to Max SHAP, Linear SHAP, and Deep SHAP. Deep SHAP is a combination of the DeepLift [71] and SHAP [49] that applies to deep learning models [49]. For model-specific methods, computation of the expected values is simplified by using assumptions like model linearity and feature independence [49].

2.7 Related Work

Research by Bauder [10] looked at supervised machine learning methods to detect fraudulent medicare providers across various states in America. The study evaluated three machine learning models indicating that decision tree and logistic regression are good performing models. The lack of fraud labels contributed to the imbalance of the data and a random under-sampling strategy was employed to create the different class distributions. The sparsity of medical claim data and the availability of labelled fraudulent cases highlighted in [10, 55, 70, 74, 43] is a common challenge when solving for anomaly detection problems.

Canadian researchers [55] experimented on detecting anomalies using an unsupervised Spectral Ranking Approach (SRA). The problem used an unsupervised learning approach in ranking anomalies using SRA. The study focused on detecting anomalies using similarity kernels [55]. Outcomes from the research also highlighted the most important features to classify fraudulent claims that are policy features, car types, and cause of accident features.

The work of [29] includes a comparative survey over the last 20 years of outlier detection relating to fraud detection machine learning algorithms. The study indicates that iForest is a suitable model for efficiently identifying anomalies with good potential on scalability along with optimised memory utilisation when using large data sets. In contrast, OCSVM, although considered to be another good model for anomaly detection, does not perform well on large data sets and also can be challenging in tuning the input parameters [55].

Besides traditional methods, deep learning based unsupervised anomaly detection algorithms [28, 35, 79, 68, 84, 13] have garnered a lot of attention recently. For instance, Deep Autoencoding Gaussian Mixture Model (DAGMM) [13] combines an autoencoder and Gaussian model to highlight the density distribution. Adversarially Learned Anomaly Detection (ALAD) [85] is an anomaly detection approach that uses reconstruction error to determine how far is the input sample from its reconstruction data. Additionally, the study by Zheng [88] on the one-class adversarial nets (OCAN) enhanced the autoencoder to include a Long Short Term Memory (LSTM) and GANs discriminator for fraud detection using only benign users' as training data [88]. During training, the OCAN uses a LSTM-Autoencoder to learn the online activities of benign users. The data from the LSTM-Autoencoder is then used in discriminator of a complementary GANs model to detect malicious users [88].

2.8 Literature Limitations and Gaps

From the review of the literature presented in Chapter 2, the following gaps have been identified which the research objectives aim to achieve:

- Popular research across the healthcare and machine learning domains are either based on supervised learning [10], predefined medical rules [40], application of anomaly detection on medical images [68], or non healthcare data [84]. Furthermore, research like the ones by Carvalho [18] and Bauder [10] also highlighted challenges like the availability of HCP data; even if it is available, there is not enough reliable data since the providers generate it [18]. However, not much literature notes the application of unsupervised deep learning models for predicting healthcare fraud and cost abuse.

- Although there is a vast amount of research solving anomaly detection problems, the majority of it is limited to the identification of anomalies [4, 10, 40, 84]. Most research largely assumes that the results can be generalised across the healthcare industry and used to explain the factors contributing to anomalous HCP [36]. However, recent literature shows little evidence of quantifying the contributing factors that influence healthcare fraud and cost abuse.

Given the above, the current research explores a two-step approach for anomaly detection. First, a GANs based approach was applied to identify the *anomaly* or *normal* labels. Second, the results from the deep learning model serve as labels for identifying the features that contribute to the anomalous data points.

2.9 Chapter Summary

This chapter discussed some of the factors relating to healthcare cost abuse, fraud and anomaly detection. Then there was a review of unsupervised methods and their ability to solve anomaly detection problems. Finally, GANs and related properties like activation functions, and training challenges relevant to this study were described. The next chapter provides an overview of the methodology, and discusses the various algorithms used in the experiment.

Chapter 3

Methodology

3.1 Introduction

The previous chapter gave an overview of HCP fraud, cost abuse, and highlighted the various types of cost abuse. Also, supervised and unsupervised anomaly detection methods, and how these algorithms can aid in the generation of labels and interpretability were discussed. Anomaly detection still comes with numerous challenges about the availability of labels and HCP data. The suggested research framework plans to address these challenges discussed in the literature limitations and gaps.

To answer the research questions highlighted in Section 1.4, a quantitative research method has been adopted as the research design. Saunders Lewis [66] defines the research design as a general approach to answering the research questions. This chapter provides a high-level framework which serves as a basic outline of the various algorithms used to answer the research question. The different steps of the research framework will be described in detail in the rest of this chapter.

3.2 Research Framework

The current research aims to create an anomaly detection model in the absence of labels that can be used to detect anomalous medical transactions from HCPs. Furthermore, this research conducted several experiments to first identify labels and then provide insights into the features contributing to the anomalies. Figure 3.1 illustrates the proposed framework used in this research, specifically highlighting specifically the datasets used and the algorithms used in the two-step modelling approach to detect anomalous medical transactions from HCPs. The first modelling step is a GANs model designed to identify the anomalous HCPs based on the reconstruction error. The second modelling step uses the anomaly labels in the supervised classification models and SHAP to explain the features contributing to the anomaly.

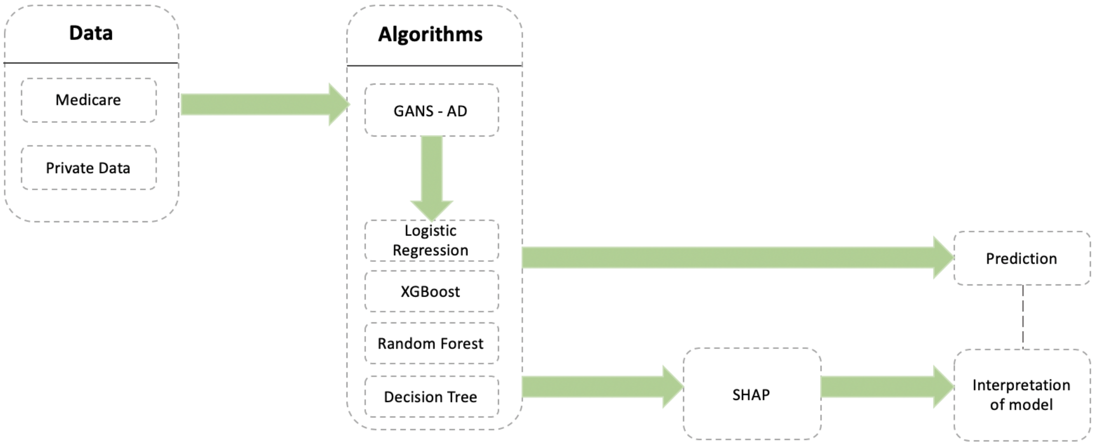


Figure 3.1: Proposed methodology approach

Chapter 4 covers the collection of the two datasets used in the experiment, as well as the pre-processing and selected features across the two HCP datasets.

3.3 Algorithms

Both datasets described in Section 4.2 do not contain any labels. Unlike other machine learning algorithms, that require a vast amount of labelled data in order to generalise well, GANs can be trained with missing data [68, 1] and can also improve the performance of classifiers when limited data is available. The labels are defined by the application of a GANs with a 'feature-matching' anomaly score. Thereafter, several classification models (Random Forest, Decision Tree, Logistic Regression, and Extreme Gradient Boosting) were applied to get a deeper understanding of how the features contribute to the anomalous labels.

3.3.1 Generative Adversarial Network

GANs algorithm consists of two adversarial networks, a generator G and a discriminator D [37]. The first neural network is the generator G that generates samples (p_g) from a noise vector (p_z) and maps the vector to a data space $G(\mathbf{z})$. The data space $G(\mathbf{z})$ is a one dimensional vector from latent space $\mathcal{Z}$. The main purpose G is to generate samples that can represent a similar distribution of x to fool D.

A second neural network called the discriminator D outputs a single vector $D(x)$ representing the probability that x came from the distribution function of the original data represented by $p_{data}(x)$ rather than the generator p_g. The purpose of the GANs is to train the discriminator D to maximise the probability to correctly classify both the real input data and data from the generator G. Furthermore, a binary cross-entropy loss function was applied to optimise the D and G through a min-max game:

$$\min_G \max_D V(D.G) = \mathbb{E}_{\mathbf{X}\sim p_{data}(x)}\left[\log D\left(\mathbf{X}\right)\right] + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}(}z)}\left[\log\left(1 - D\left(G\left(\mathbf{z}\right)\right)\right)\right], \tag{3.1}$$

where ($[\log D\left(\mathbf{X}\right)]$) is the real distribution of data that passes through the discriminator (normal data). The discriminator tries to maximise these data samples to 1. The term ($[\log\left(1 - D\left(G\left(\mathbf{z}\right)\right)\right)]$) represents data from random input that passes through the generator, which then generates fake samples which are then passed through the discriminator to identify the anomaly. In this term, the discriminator tries to maximise it to 0. So overall, the discriminator tries to maximize the function V. Similarly, the task of the generator is exactly the opposite, it tries to minimise the function V so that the differentiation between normal and anomalous data is at a minimum. The second term in Equation 3.1 ($[\log\left(1 - D\left(G\left(\mathbf{z}\right)\right)\right)]$) represents data from random input that passes through the generator and then generates fake samples which are then passed through the discriminator to identify the anomaly. During this term, the discriminator tries to maximise it to 0. So overall, the discriminator tries to maximise the function V. Similarly, the task of the generator is exactly the opposite, it tries to minimise the function

V so that the differentiation between normal and anomalous data is at a minimum.

Feature Matching

Feature matching aims to solve the instability issue in GANs by defining a generator function that prevents the discriminator from over training [65]. With feature matching, the main objective of the generator is to produce samples that matches the distribution of the real input samples instead of optimizing the performance of the discriminator [65]. During training, the generator network have to fit the expected values of the discriminator's intermediate layer [65]. During the training process, a discriminator feature loss calculates the difference in characteristics between the real and generated data [65]. Results from previous studies indicate that feature matching can be used in situations where GAN stability is a challenge [84, 65, 86].

Exponential Moving Average

The convergence issue mentioned in Section 2.5.2 can be overcome by the application of feature matching and Exponential Moving Average (EMA) during the training process [65, 27]. Averaging methods such as EMA are typically more robust, simple to implement and have minimal computation overheads [83]. As they operate outside of the GANs training loop, they do not influence optimal points between the generator G and discriminator D [83]. Exponential Moving Average (EMA) is an option to solve the convergence issue [83]. EMA is a smoothing filter that is frequently used to solve time series problems [83]. Bosman defines EMA as "the moving average for each step by assigning exponentially-decaying weights to the previous steps. The weight for each of the previous step decreases exponentially but never reaching zero" [16]. EMA is defined by the equation below:

$$\theta^{(t)}_{EMA} = \beta\theta^{(t-1)}_{EMA} + (1-\beta)\theta^{(t)}, \tag{3.2}$$

where $\theta^{(t)}_{EMA}$ represents the moving average at the t^{th} data point for the sample $\theta^{(t)}$ while $\theta^{(t-1)}_{EMA}$ represents the moving average at the $(t-1)^{th}$ data point for the sample $\theta^{(t-1)}$. Note that β is the decay coefficient and can be approximated as $\beta = 1 - 1/n$ with

n the number of samples to average. For large n, β converges to 1, and for small n, β converges to 0. As n increases β increases towards 1, it accelerates decay and weaken smoothing. Conversely, when β decreases closer to 0, it result in a slower decay with stronger smoothing. Therefore, to balance decay and smoothing, it is essential to find the appropriate β.

Activation Function

The various activation functions used in GAN architectures were discussed in Section 2.5.1. For the current classification problem, binary output is typically expected for the target variable, thus the output value should be within the binary range of 0 and 1. Therefore, the sigmoid activation is used for the discriminator output layer, due to the output range of (0; 1). For the hidden layers, the Relu activation function was applied across the generator and the discriminator. The detailed activation functions applied for the two datasets for the generator and the discriminator networks can be found in Tables 3.1 and 3.2.

Dropout

Dropout is a regularisation method that prevents overfitting in GANs by randomly dropping units from the neural network during training [64]. Effectively during training, samples dropout from the number of different networks prevented units from co-adapting too much. Research by [52] suggests that using a dropout value between 0.2 and 0.5 often led to better results. For our study, there was no dropout applied to the generator network, however, a dropout rate of 0.2 was applied to each layer on the discriminator network across the two datasets.

Anomaly Detection

The GANs algorithm labels data as normal or anomalous through the use of a loss function called the *anomaly score*. The *anomaly score* was calculated for every evaluation between normal and generated samples in the training process [68, 84]. The *anomaly score* is expressed mathematically as:

$$A(x) = (1 - \lambda)\mathcal{L}_G + \lambda\mathcal{L}_D. \tag{3.3}$$

Here, $\mathcal{L}_G$ and $\mathcal{L}_D$ are the generator and discriminator losses respectively. The discriminator loss $\mathcal{L}_D$ applied a feature-matching approach [85]. Feature-matching defined as:

$$\mathcal{L}_D(x) = \parallel f_D(x, D(x)) - f_D(G(x), D(x)) \parallel, \tag{3.4}$$

with f_D returning the discriminator intermediate layer for the given input sample x. $f_D(x, D(x))$ evaluates the original sample against the reconstructed data features in the discriminator. Similarly, $f_D(G(x), D(x))$ evaluates the generator samples against the reconstructed data features in the discriminator. For a given data sample x, a high anomaly score of $A(x)$ indicates possible anomalies within the sample. The evaluation criteria for this is to a threshold (ϕ) the score, where $A(x) > \phi$ indicates an anomaly.

GAN Architecture

For the experiments setup in this thesis, we used a neural network for the generator with three layers and different hidden layers for each layer. The neural network for the discriminator has four layers and different hidden layers. Inspired by Li's research [47], regarding the latent space dimension, a higher latent space dimensions were used. This improved the sample generation [47], especially in the context of anomaly detection [47, 84]. Thus, the latent space of the generator is the same as the number of features in the train set. To demonstrate our model's ability, we train both the Medicare and private datasets on a GANs architecture and hyperparameters identified in Table 3.1 and 3.2.

3.3.2 K-Means Algorithm

K-Means is a clustering analysis algorithm that groups observations based on their feature values into K clusters [91, 9]. Observations that are classified into the same cluster having similar feature values. The K-Means algorithm is applied using the following steps:

Table 3.1: Medicare GAN Architecture and hyperparameters

Operation	Units	Activation	Dropout
$G(z)$			
Dense Layer	64	ReLU	0.0
Dense Layer	128	ReLU	0.0
Dense Layer	70	Linear	0.0
$D(x)$			
Dense Layer	256	Leaky ReLU	0.2
Dense Layer	128	Leaky ReLU	0.2
Dense Layer	128	Leaky ReLU	0.2
Dense Layer	1	Sigmoid	0.0
Optimiser	Adam ($\alpha = 10^{-5}$,$\beta_1 = 0.4$)		
Batch size	50		
Latent dimension	70		
Epochs	200		
Leaky ReLU slope	0.1		
Weight, bias initialisation	Xavier Initialiser, Constant(0)		

1. Define the number of clusters K which is a positive integer. One of the commonly used methods to define K, is using the Elbow technique.

2. For each cluster, the centroids are calculated. This is usually done by randomly splitting all observations into K clusters, and verifying that each cluster centroid is different.

3. Calculate the distances to the centroids for each cluster by iterating over all the observations. Thereafter, assign each observation to a cluster based on the shortest distance to the nearest centroid.

4. Repeat step 2 and 3 until the centroids for each cluster do not change.

Since K-Means is a distance-based clustering algorithm, for anomaly detection problems, the distance is used as a measure of normal or anomalous observation [90]. For the

Table 3.2: Private GAN Architecture and hyperparameters

Operation	Units	Activation	Dropout
$G(z)$			
Dense Layer	64	ReLU	0.0
Dense Layer	128	ReLU	0.0
Dense Layer	54	Linear	0.0
$D(x)$			
Dense Layer	256	Leaky ReLU	0.2
Dense Layer	128	Leaky ReLU	0.2
Dense Layer	128	Leaky ReLU	0.2
Dense Layer	1	Sigmoid	0.0
Optimiser	Adam ($\alpha = 10^{-4}$,$\beta_1 = 0.4$)		
Batch size	50		
Latent dimension	54		
Epochs	500		
Leaky ReLU slope	0.1		
Weight,bias initialisation	Xavier Initialiser, Constant(0)		

current study, the distance of each observation from the centroids is calculated using the Euclidean distance metric. Equation 3.5 shows the Euclidean mathematical definition as:

$$d(x,y) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2}, \tag{3.5}$$

where $x = (x_1, ..., x_m)$ and $y = (y_1, ..., y_m)$ are input vectors with m representing the number of features. The smaller the distance between objects the greater the similarity [90]. The observations whose Euclidean distance is greater than the 90th percentile are labelled as anomalies.

3.3.3 Classification Algorithms:

The four binary classification algorithms were applied in the second part of the study to give interpretability to the anomalies. The algorithms applied include the logistic regression (LR), extreme gradient boosting (XGB), random forest (RF), and decision tree (DT) [69].

The LR, XGB, RF, and DT algorithms are successful in detecting anomalies [31, 69, 57] however, their main use in the current study is their ability to generalise, feature selection, interpretability [31, 69, 57, 89] and further explain how the features contribute to the anomalous HCP. These algorithms are summarised highlighting their core capability.

Tables 3.3, 3.5, 3.4, and 3.6 highlight the key parameters and setting of the classification models applied to the training data through cross-validation.

Logistic Regression

Logistic Regression (LR) is a classification algorithm that is used to predict the probability of a binary dependent variable. In the current context of anomaly detection, the target variable contains data coded as 1 (anomaly) or 0 (normal). The objective of our LR algorithm is to find the best fit that describes the relationship between the target variable (anomaly or normal) and the predictor variables [90]. The logistic regression algorithm is given in Equation 3.6 below:

$$Y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + ... + \beta_j x_{i,j}, \tag{3.6}$$

where Y_i is the target variable output of i^{th} observation, β is a vector of regression coefficients, and $x_{i,j}$ is the j^{th} predictor variable for the i^{th} observation. Similarly to linear models, the regression coefficients β is a logit rather than a mean. Thus, holding all other predictors constant, β_j is the change in the logit of the probability in the j^{th} predictor.

LR has been widely used in credit scoring and fraud detection domain due to its extensive interpretability and generalisation abilities [89]. LR is a linear model and not suitable to solve complex non-linear problems [90, 57]. Therefore, the use of a LR model to solve a non-linear problem is dependant on a strong feature engineering process [90].

Alternatively, Zhu suggests the use of other non-linear models [90].

The binary LR algorithm was applied to both datasets using the model settings listed in Table 3.3. For each dataset, the data was split into training and test datasets. The test dataset was used to evaluate the LR model fit. The LR calculates probability across the two classes (normal and anomaly) that was associated with HCP which was previously discussed in the literature review.

Table 3.3: Modelling procedure used for logistic regression

Category	Setting
Modelling Method	Binary Logistic Regression
Sampling Type	K-Fold Cross Validation
Iterations	10
Sampling Method	Simple Random Sampling
Model Selection Criteria	AUC / Sensitivity / Specificity
Software Version	Python 3.7.1
Software Packages	sklearn.LogisticRegression

Decision Tree

Decision Tree (DT) is a simple and intuitive algorithm that utilises a top-down approach in which the root node creates binary splits until a certain criteria are met [31]. DT structures separate the data into groups that are mutually-exclusive[57]. These groups are created by recursively separating all observations either using a breadth-first or depth-first greedy approach until all observations are allocated to a group [89]. In the current context of anomaly classification, the DT model outputs a predicted target class (anomaly or normal) for each terminal node produced. Decision Trees automatically reduce complexity, useful for the selection of features, and therefore, the predictive analysis structure is understandable and interpretable [69].

However, one of the challenges with DTs is the dependence on certain features. For example, if anomalies occur on a set of features, then the DT model tends to generalise decisions based on those set of features [89]. In comparison to RF, the RF model avoids this problem by applying the bagging principle [57, 69]. For such cases, when features are not very reliable, an alternative is to use a RF algorithm [89].

A DT model was applied to the training data. The DT was trained using the Decision Tree Classifier algorithm and no pruning was done. The Gini index was the standard index used to determine the quality of each node split.

Table 3.4: Modelling procedure used for Decision Tree

Category	Setting
Modelling Method	Decision Tree
Sampling Type	K-Fold Cross Validation
Iterations	10
Sampling Method	Simple Random Sampling
Model Selection Criteria	AUC / Sensitivity / Specificity
Software Version	Python 3.7.1
Software Packages	sklearn.tree.DecisionTreeClassifier

Random Forest

Random Forest (RF) is a tree constructed algorithm from a set of possible trees with random features at each node [57]. A random forest can be efficiently generated and the combination of large sets of random trees generally leads to accurate models to detect anomalies [69]. Additionally, the random forest algorithm has been used in this study due to its versatility in being applied to large datasets and feature importance [31]. In the current classification context, the random forest classifier is defined mathematically by:

$$m(x : \Theta_1, ..., \Theta_K) = \begin{cases} 1 & \text{if} \quad \frac{1}{K}\sum_{j=1}^{K} m(x; \Theta_j) > \frac{1}{2}, \\ 0 & \text{otherwise} \end{cases} \tag{3.7}$$

where m is the random forest classifier obtained via a majority vote among K classification trees with input x and Θ is the parameter set. The predicted model $m(x : \Theta_1, ..., \Theta_K)$ is the aggregation of the majority votes from the classification prediction of individual trees $(m(x; \Theta_j))$ on input training data(x).

The RF algorithm was applied to both Medicare and private training datasets using the model settings listed in Table 3.5. The Gini index was the standard index used to determine the quality of each node split and thus contributing to the feature importance calculation method.

Table 3.5: Modelling procedure used for Random Forest

Category	Setting
Modelling Method	Random Forest
Sampling Type	K-Fold Cross Validation
Iterations	10
Sampling Method	Simple Random Sampling
Model Selection Criteria	AUC / Sensitivity / Specificity
Software Version	Python 3.7.1
Software Packages	sklearn.ensemble.RandomForestClassifier

Extreme Gradient Boosting

Extreme Gradient Boosting (XGB) is a powerful machine learning technique for classification, regression, and ranking problems that produces a prediction model in the form of an ensemble DT [57]. The XGB model is built in a multi-step approach where each step, introduces a new weak learner to compensate for the shortcomings of the existing weak learners [57].

The XGB model has a unique approach that uses a regularised model formalisation to achieve better performance, controls complexity, and reduces the risk of overfitting. Gradient boosting relies on regression trees, where the optimisation step's purpose is to reduce mean square error, and for binary classification is to optimise the standard logarithmic loss. For multi-class classification problems, the objective function is to optimise the cross-entropy loss. Combining the loss function with a regularisation term arrives at the objective function. Furthermore, XGB uses gradient descent for optimisation to improve the predictive accuracy at each step by following the negative of the gradient during the process of finding the sink within the dimensional plane. The set of functions used in the XGB model minimises the following regularised objective

$$L(\Theta) = \sum_i l(y_i, \hat{y}_i) + \Omega(\Theta), \tag{3.8}$$

where Θ is the learned parameter set, l is a differentiable convex loss function that measures the difference between the predictions $\hat{y}_i$, the target y_i and Ω is the regularisation term.

The XGB model applied the settings listed in Table 3.6, a wrapper package for the XGB model was fit on the training data. The following hyperparameters are applied as part of the training step:

- Objective function set to 'Binary:Logistic' due to the target class is 1 and 0 (anomaly and normal)
- A learning rate of 0.1
- Tree-specific tuning parameters maximum depth = 7, the minimum child weight is equal to 1, gamma to 0.3, and subsample to 0.9.

Table 3.6: Modelling procedure used for XGB

Category	Setting
Modelling Method	Extreme Gradient Boosting
Sampling Type	K-Fold Cross Validation
Iterations	10
Sampling Method	Simple Random Sampling
Model Selection Criteria	AUC / Sensitivity / Specificity
Software Version	Python 3.7.1
Software Packages	xgboost.XGBClassifier

3.3.4 SHapley Additive exPlanation

In the context of the current study, SHAP provided a unified approach for the interpretability of the features in detecting anomalies across HCP. As discussed in Section

2.6 the SHAP framework was applied to assist in explaining the feature importance and the features that contribute to the anomalies. Table 3.7 shows the SHAP package used for the explanation of the features contributing to the anomaly [49].

Table 3.7: Modelling procedure used for SHAP

Category	Setting
Software	Python 3.7.1
Software Packages	SHAP

3.4 Model Training and Testing

Training and testing of the model is imperative to ensure the rigidity of the model, as well as to ensure the model functions as expected. The current study implemented a similar anomaly GANs (AnoGAN) architecture [84] highlighted in the research study named ”Efficient GAN-based Anomaly Detection” [84, 68]. All experiments are performed using the TensorFlow library in Python 3.7.1.

Across the two different datasets, we trained the GANs model using the architecture and hyperparameter settings listed in Section 3.3.1 and the results are presented in Chapter 5. Additional to the training process, EMA is applied, with a decay of 0.999 for both the private and Medicare datasets. Furthermore, the training process of the second step research approach for the classification models was through crossing validation. The process used a 10-fold cross-validation. For each fold in the cross-validation process, a randomised copy of the test dataset is used. The results was then averaged to produce a value for the accuracy and sensitivity evaluation metrics. Some of the advantages of using cross-validation are that it improves the validation of the models and aids in the reduction of biases [57].

Having trained the GANs model to yield results for the generator (G) and discriminator D, the test dataset was batched and scored by the anomaly score function $A(x)$ defined in Section 3.3.1. The anomaly detection accuracy of the GANs model was evaluated on the private dataset in the form of a case study by investigating the anomalies identified through the anomaly score $A(x)$ with business feedback. The anomaly detection of HCP was based on the anomaly score $A(x)$ using $\lambda = 0.1$ stated in Equation 3.3 and using a threshold of 90%.

3.5 Model Evaluation

We evaluated the performance of the anomaly detection algorithms by using quantitative and qualitative evaluation approaches. This section discusses the approaches used to evaluate the model results. The first part of this thesis discussed in Section 3.2 used an unsupervised GANs algorithm to determine normal and anomalous labels. Specifically, the private data results will be evaluated qualitatively by domain experts to confirm the labels. In the second part of the thesis, quantitative metrics were used to evaluate the accuracy of the models. The following subsections will discuss the qualitative and quantitative methods in detail.

3.5.1 Qualitative Evaluation

As part of a qualitative evaluation of the results, the GANs output results were shared with domain experts. Research by [32] suggests that machine model evaluation by domain experts plays a critical role in an unsupervised machine learning context. Evaluation methods are useful not only for informing the user about the relative or absolute quality of a given model, but also for informing the business users how their actions can improve the models [32]. Furthermore, the research by [32] states that domain experts can perform a hands-on evaluation to assess model performance. For the private dataset business domain, experts have been requested to evaluate generated data and labels from the GANs model. As part of the suggested process, investigators analysed the anomalies from the GANs model.

3.5.2 Quantitative Evaluation

The two criteria's used to evaluate the model results in this research are sensitivity and specificity, also known as the true positive rate (TPR) and the true negative rate (TNR) respectively. The sensitivity (TPR) metric indicates the model's ability to correctly identify anomalous data points, and the specificity (TNR) metric indicates the model's ability to correctly identify normal data points. The sensitivity and specificity are expressed using the Equations 3.9 and 3.10.

$$Sensitivity = TPR = TP/P \tag{3.9}$$

$$Specificity = TNR = TN/N \tag{3.10}$$

In Equation 3.9, true positive (TP) represent the number of HCPs that are correctly identified as anomalous, and P represent the total number of anomalous HCPs in the actual dataset. In Equation 3.10, true negative (TN) represents the number of HCPs that are correctly identified as normal, and N representing the total number of normal HCPs instances in the actual dataset. Additional to the specificity and sensitivity metrics, the area under the curve (AUC) measurement was also applied. Araya states that "the AUC is an effective measure of accuracy which determines the overall inherent capacity of an anomaly classifier to differentiate between normal and anomalous data instances" [7]. Generally, for anomaly detection, an AUC score closer to 1 indicates the models classification ability to correctly identify normal and anomalous HCPs [7]. The second classification modelling step uses these metrics to evaluate the classification performance across the different supervised models.

3.6 Chapter Summary

In summary, this chapter described the research approach which guided this study. This was followed by the explanation of the data collection and the pre-processing steps carried out resulting in the final features used in the experiments. Then the GANs, classification models and SHAP model were discussed highlighting the architecture, hyperparameters, and settings used to conduct the experiments. Finally, the training, validation, and evaluation of the models were discussed.

Chapter 4

Data

4.1 Introduction

This section covers the sources of data used in the study, as well as the data pre-processing. The pre-processing step discusses in detail the data cleaning, feature generation, feature encoding, feature scaling, and feature selection. Lastly, this section provides an analysis of the various features relating to HCP across the two datasets.

4.2 Data Collection

The datasets used in this study relates to HCP from 2 different sources and both datasets did not contain any labels. The sources of the 2 datasets are described further below:

1. The Medicare dataset is a public dataset from the Centers for Medicare and Medicaid Services (CMS) for the 2016 calendar year only [25]. The *Medicare Provider Utilisation and Payment Data: Physician and Other Supplier* contains aggregated payment and claims data with information on services and procedures provided to claimants and beneficiaries over the age of 65.

2. The Private dataset is from a organisation from which access to carry the research has been granted. The required data was obtained from a South African company that performs the administration of claims. The data received was aggregated at

an HCP level with features representing financial and injury information for the 2018 calendar year.

4.3 Data Pre-processing

As part of the data pre-processing phase, a data cleaning process is applied, encoding of categorical attributes takes place, standardised scaling is applied, and highly correlated features are removed. Certain attributes in the data are modified to mask the HCP details in the results, due to ethical and privacy issues. The final output from the pre-processing step that was used in the GANs model contained features with a combination of categorical features. These represented the practitioner type, type of injury, and continuous features representing payment across beneficiaries and injury types. The Medicare dataset is discussed in the following sections

4.3.1 Data Cleaning

One of the major steps was the data cleaning in the data collection process. Data cleaning described by Topcu ”as a set of operations carried out for detecting, removing errors and inconsistencies from data” [77]. The data cleaning step identified attributes in the dataset that had missing or incorrect values. The continuous attributes for the Medicare dataset in Appendix 1 were set to zero if they contained either nulls or blanks. For categorical attributes, a default value of ’9999’ was used for blanks or nulls. This cleaning step was also applied to the private dataset, and the records that contained negative values in the dataset were removed.

4.3.2 Feature Generation

Health insurance is described as “a complex phenomenon governed by multiple features” [31] because there is no standard formulae of set of features that can be used to identify fraudulent HCPs [31]. One of the objectives of the thesis was to determine the factors that have an impact on the detection of fraudulent HCPs. This step involves creating features out of the original attributes to maximise the discriminatory abilities of the model in separating normal and fraudulent HCPs [31]. For the Medicare dataset, new features like average payment, the number of claimants serviced, and unique claimants serviced per day across similar injuries and HCP types were created. For the private dataset, new contextual features like reporting lag, average payment, weekday and weekend treatment indicators, and counts of claims were created.

4.3.3 Feature Encoding

The feature encoding step applied a one encoding or dummy coding method [3]. The one-hot encoding method represents each permutation of a categorical feature as a new dummy variable that takes on the value of 0 or 1 [3]. In other words, the new dummy variable takes on the value of one if the value of the associated categorical feature belongs to a particular group, otherwise zero if the value does not belong to that group [3].

Here, we aimed to encode string categorical entries for statistical analysis and enable efficient neural architecture search [19]. For example, Table 4.1 and 4.2 indicate the representation of the new dummy variables after one-hot encoding has been applied to the ”Grouping” categorical feature in the private dataset.

Groupings
Hospital
Radiology
Allied
Specialists
General Practitioners
Pharmacy

Table 4.1: Groupings Variable - values before one-hot encoding step

Hospital	Radiology	Allied	Specialists	General Practitioners	Pharmacy
0	1	1	0	0	1

Table 4.2: Groupings Variable - values after one-hot encoding step

4.3.4 Feature Scaling

Often in a dataset, the features contain values of varying scales, resulting in a greater influence on the model outcome [7]. Feature scaling such as standardisation is a method used to rescale the values in the data within a particular range. Feature scaling of features is a common step in the data cleaning and experiment process. Furthermore, feature scaling techniques like standardisation has proven to accelerate the calculations in deep learning algorithms [7, 1, 2].

The two datasets used in this experiment have features that contain values of varying ranges. For the current experiments, data standardisation was applied. For each transformed feature, the resulting distribution of its mean value is zero and the standard deviation is one [1]. The data standardisation transformation is given by:

$$z = \frac{x - \mu}{\sigma}, \tag{4.1}$$

where x is the value of each feature in the Medicare and private dataset used in our experiment, μ and σ are the mean and standard deviation of that feature respectively.

4.3.5 Feature Selection

The features required and the necessary pre-processing steps were discussed in the previous three subsections. After the data was cleaned, additional features were generated and thereafter feature selection was carried out to determine which features capture the behaviour of an HCP.

Part of the analysis was to explore the distributions of payments, HCPs, HCP types and injury types, and the severity of injuries through the use of visualisation tools such as bar charts, histograms, and box plots. Thereafter, a correlation matrix was applied to Medicare and private dataset to determine the features that can be used in the modelling steps. Through this step variables that was highly correlated was removed resulting in the final feature across the two datasets. Sections 4.3.6 and 4.3.7 discuss the features used in the modelling section in detail.

4.3.6 Data Analysis : Private data

This subsection covers the analysis of selected features from the private data. The analysis was carried out across the HCP groups, practitioner types, injury, and the severity of these injuries.

Healthcare Provider Groups

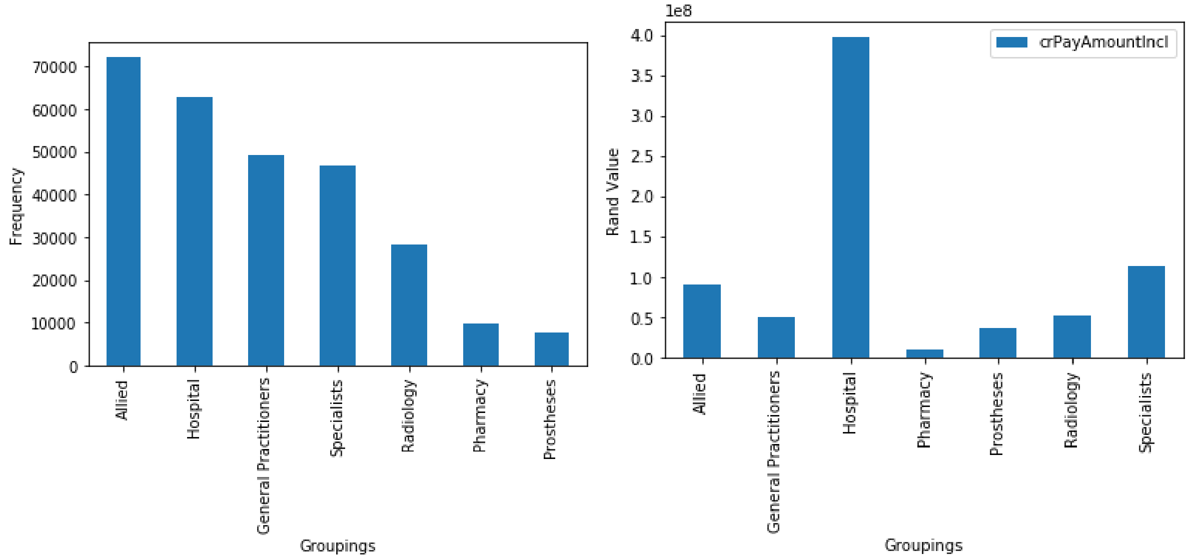
Healthcare provider groups consist of medical professionals, services, and workers who provide medical care and support to those in need. The provider groups consist of seven predefined groupings, of which every HCP can be classified into at least one of these groups. To show the HCP group coverage, the groups distribution diagrams of the sample points obtained by each of the HCP groupings were plotted. The resulting histograms are shown in Figure 5.21. Figure 4.1a shows that Allied, Hospitals, and Practitioners are the top three groups with the most frequent transactions among the highlighted seven groups. In contrast, Figure 4.1b shows that the highest payments are made to Hospitals followed by Specialists and Allied. This is an indication that there are many, smaller valued payments across the Allied groups.

Healthcare Practitioner Types

Healthcare practitioner types indicate health services that are dedicated to certain injuries or health issues. The practitioner types consist of numerous different types, of which every HCP is classified into these types. These practitioner types are highlighted in Figure 4.2 and the descriptions have been masked.

Injury and Severity

Figure 4.3 show stacked bar chart of the number of claims with the samples obtained by the various DRG codes. Figure 4.3 represents the type of injuries and the severity of these injuries. The type of injuries is represented by the Diagnosis Related Group (DRG) instead of an ICD10 code. According to Figure 4.3, DRG code 13 (injuries to the wrist and hand) has the highest number of claims being treated followed by DRG code 15 (injuries to the knee and lower leg). Severe cases were distributed across DRG's



(a) Frequency distribution

(b) Payment distribution

Figure 4.1: Distribution of frequency and payments with the samples obtained across the various HCP groups

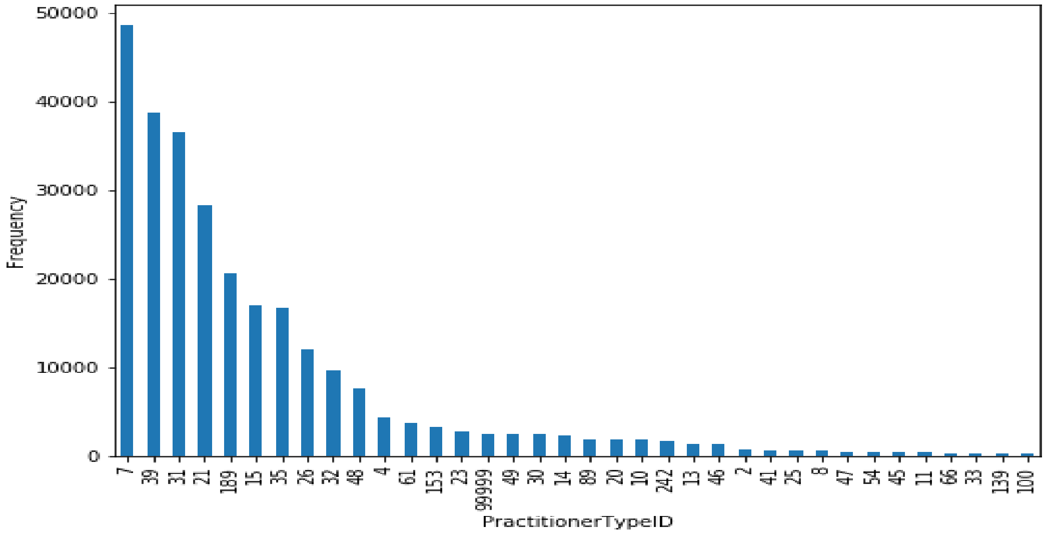


Figure 4.2: Distribution of frequency with the samples obtained across the various healthcare practitioner types

0 to 20, with the highest in DRG 1 (spinal cord injuries) and DRG 13 (injuries to the wrist and hand).

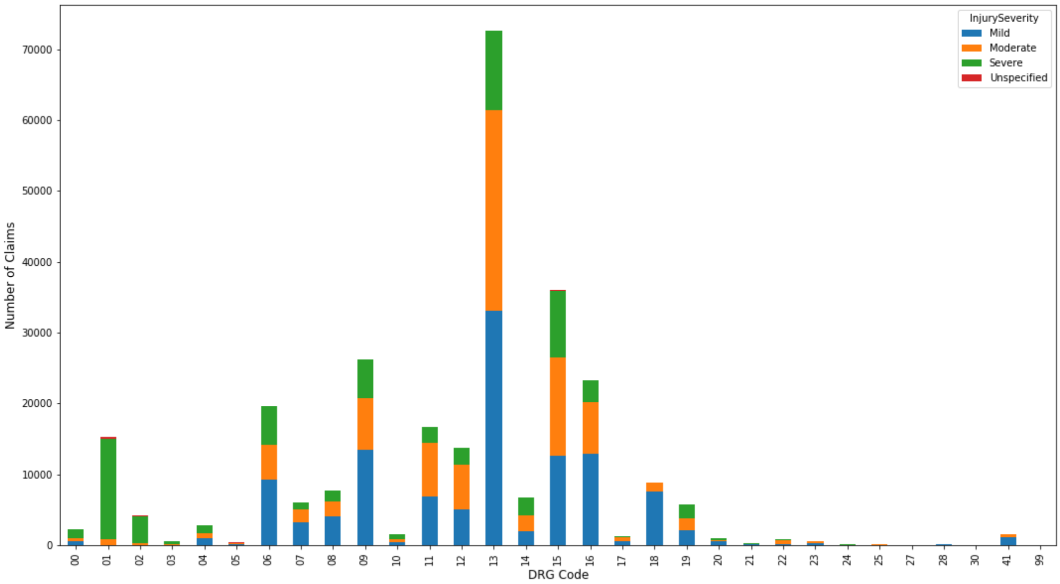


Figure 4.3: The number of claims across injury type and severity

The table A.1 in Annexure A contains the entire list of variables from the data received. Table 4.3, is the list of final variables used in training the models. After the data pre-processing steps applied to the private dataset, the final dataset contained 3 567 transactions with 54 features.

Table 4.3: Private variables used in training the model

Variable Name	**Description**
DRGCode	The group which the ICD10 belongs to
InjurySeverity	The severity of the injury (Mild, Moderate or Severe)
Groupings	Represent the group an HCP belong to (i.e Physiotherapist, Hospitals, Radiologist, etc)
crPayAmountIncl	Amount paid including VAT

PracTypeIdClean	Identifier representing the practitioner type the HCP belongs to. Due to the numerous combinations of practitioner types, certain types had to be grouped together
HospitalGroupIDClean	Identifier representing the hospital group the HCP belongs to. Due to the numerous combinations of hospital groups, certain groups had to be grouped together
AvgMSPPaid	Average payment across HCP with the same injury, injury severity, healthcare type and healthcare group
AvgMSPReportingLag	Average reporting variance between service and reporting date across HCP with the same injury, injury severity, healthcare type and healthcare group
MSPDistinctcountClaim	Count of unique claims across HCP with the same injury, injury severity, healthcare type and healthcare group
TotalMSPClaim	Count of claims across HCP with the same injury, injury severity, healthcare type and healthcare group
WeekendTreatmentCount	Number of claimants the HCP has treated during Saturday and Sunday
WeekendTreatmentSum	Total number claimants the HCP has treated during Saturday and Sunday
WeekendTreatmentUniqueCount	Total unique number of claimants the HCP have treated during Saturday and Sunday
WeekendTreatmentAvg	Average number of claimants the HCP have treated during Saturday and Sunday
WeekendTreatmentUniqueAvg	Average number of unique claimants the HCP have treated during Saturday and Sunday
WeekdayTreatmentCount	Number of claimants the HCP has treated from Monday to Friday
WeekdayTreatmentSum	Total number of claimants the HCP has treated from Monday to Friday

WeekdayTreatmentUniqueCount	Total number of unique claimants the HCP has treated from Monday to Friday
WeekdayTreatmentAvg	Average number of claimants the HCP has treated from Monday to Friday
WeekdayTreatmentUniqueAvg	Average number of unique claimants the HCP has treated from Monday to Friday

4.3.7 Data Analysis : Medicare

This subsection covers the analysis of selected features from the Medicare dataset.

Healthcare Practitioner Types

Figure 4.4 show the top 10 results obtained for different healthcare practitioner in the Medicare dataset. The graph shows radiology, internal medicine and family practices are the predominant medical services.

Medical services

Figure 4.5 show the distribution of claims received obtained for different types of injury. The top 10 medical services offered across the Medicare dataset are listed in Figure 4.5. The results show medical service codes 99213 and 99214 which is the treatment to outpatients are the top two dominant services rendered based on the number of times billed.

Similarly, listed in Table 4.4 is the list of Medicare variables used in training the models. The full set of the features is listed in A.2 in Annexure A. After the data pre-processing steps applied to the private dataset, the final dataset contained 199 567 transactions with 71 features.

Table 4.4: Medicare variables used for training the model

Variable Name	**Description**

NPI	National Provider Identifier (NPI) is a unique identifier for each HCP.
ProviderType	HCP type
HCPCSCode	Is a unique code used to identify the specific medical service rendered by the HCP.
HCPCSDrugIndicator	Identifies whether the HCPCS code for the specific service furnished by the provider is a HCPCS list
LineSrvcCount	Count of the number of treatments/services provided
BeneUniqueCnt	Number of unique beneficiaries receiving the treatment/service.
BeneDaySrvcCnt	Count the number of unique beneficiary/per day treatment/service.
AverageMedicarePaymentAmt	Average amount paid after deductible and coinsurance amounts
AveragePaymentAmountPerTypeCode	Average amount paid after deductible and coinsurance amounts across same healthcare type and medical service
AverageSRVCCountPerCodeType	Average number of services provided across the same HCP and medical service
AverageUniqueCountPerCodeType	Average unique beneficiaries receiving the service across the same HCP and medical service
AverageDaySrvcCountPerCodeType	Average distinct Medicare beneficiary/per day services across the same HCP and medical service
AverageUniqueServicesCountPerCodeType	Average number of service received by each distinct Medicare beneficiaries

AverageSrvcByAverageUnique	Number of unique beneficiaries serviced across the same HCP and medical service

4.3.8 Data Sampling

Generally in the model building process, the data is divided into 3 partitions. These three partitions are commonly referred to as train, test and validation datasets. There are various techniques to split the data into these partitions, however one of the widely used technique is the random sampling, that splits the partitions proportionally [41]. The training dataset will always contain the highest number of observations cause this is the dataset used to build the model. The validation dataset is used to optimise the hyper-parameters values while the test dataset is used to test the model for robustness and generalisation [66]. Due to the nature of the current study, the data was divided into 2 partitions. A simple random sample applying 70% of the data was partitioned for training the models and the remaining 30% used for testing the model.

4.4 Chapter Summary

In summary, this chapter described the sources of the two datasets. This was followed by the pre-processing steps carried out resulting in the final features used in the experiments. The chapter was concluded with the preliminary analysis across the HCP and the medical services carried out.

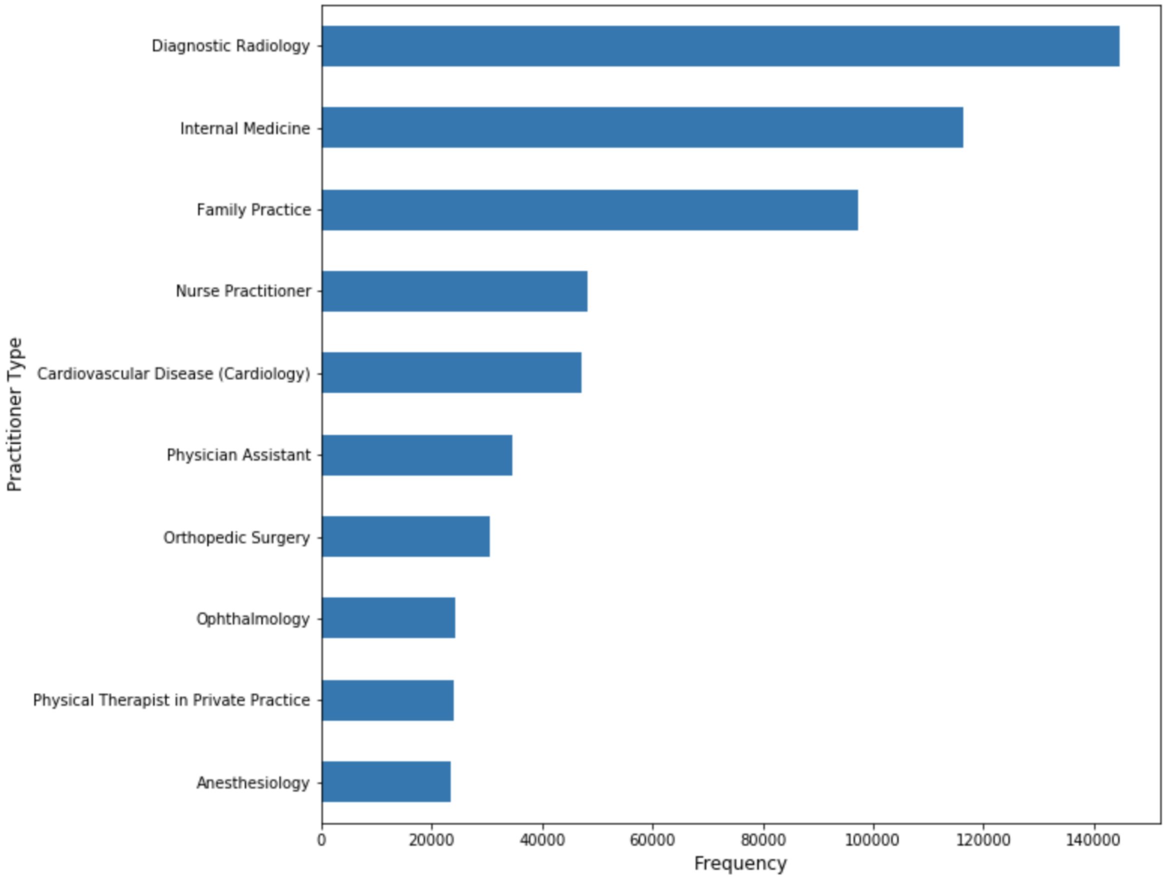


Figure 4.4: Distribution of the number of claims received across HCP Types

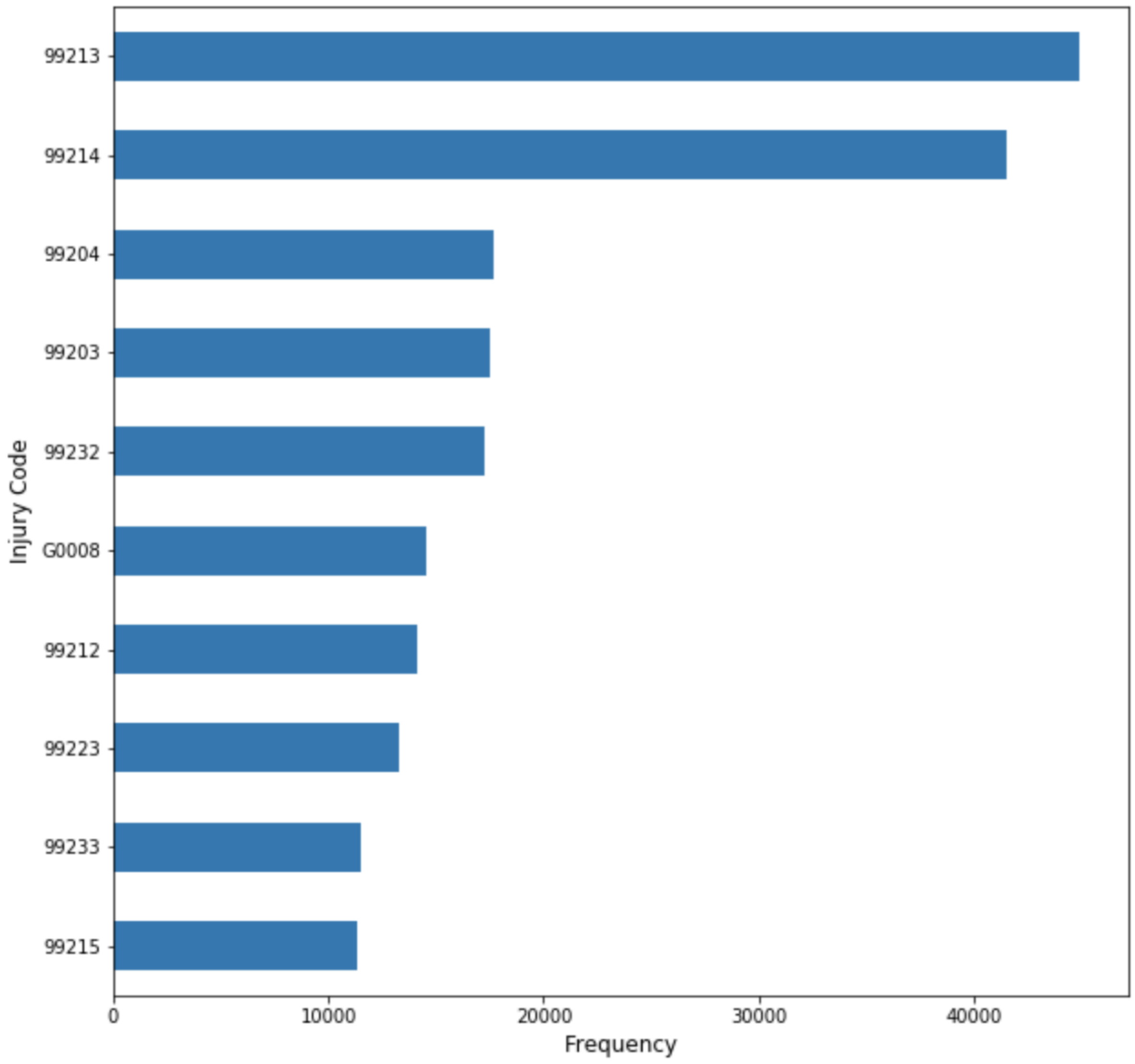


Figure 4.5: Distribution of the frequency of medical services

Chapter 5

Results and Discussions

5.1 Introduction

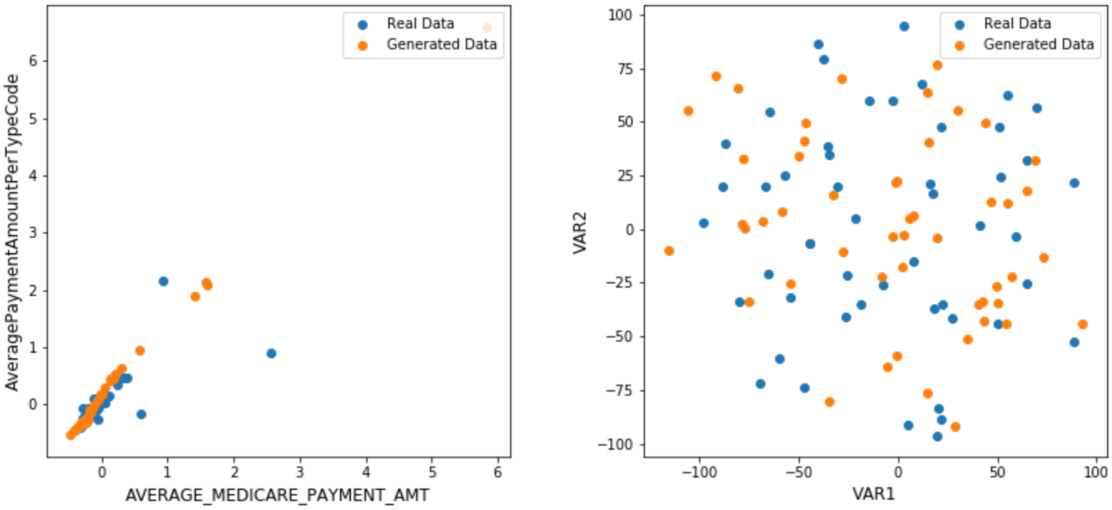
The results of this study are discussed in this chapter. Section 5.2 discusses the results for the Medicare dataset while Section 5.3 discusses the results for private data. For each of these sections, a GANs model was created to identify anomalies and create labels. Thereafter, the results from the supervised classification models were discussed followed by the results from the SHAP model.

5.2 Medicare results

5.2.1 Generation of realistic data

The challenge in the study is the lack of fraud labels across the datasets which plays an important role in measuring model performance and accuracy in machine learning models. Therefore, the current study to adopt a GANs approach to generate fraud labels that are used as the ground truth. The trained generator in the GANs model generates realistic data across the different features. The generated data is conditioned by sampling from latent representations z as discussed in Section 3.3.1. The purpose of analysing the generated data is to understand if the generator G is learning to create realistic data in order for the discriminator D to correctly identify anomalous or normal labels.

Figure 5.1 highlights the data distribution of the generated data and real data. The generated data for a single batch across the 2 variables $AverageMedicarePaymentAmt$ and $AveragePaymentAmoutPerTypeCode$ show a similar distribution when compared to the real data. The data points are distributed between 0 and 8 across the 2 variables (see Figure 5.2a). In this study we also used T-distributed Stochastic Neighbour Embedding (TSNE) to reduce the high dimensional space to 2 dimensions. On the TSNE data Figure 5.2b shows the generated data having a similar distribution to the real data.



(a) Comparison of generation and real data across 2 variables without using TSNE

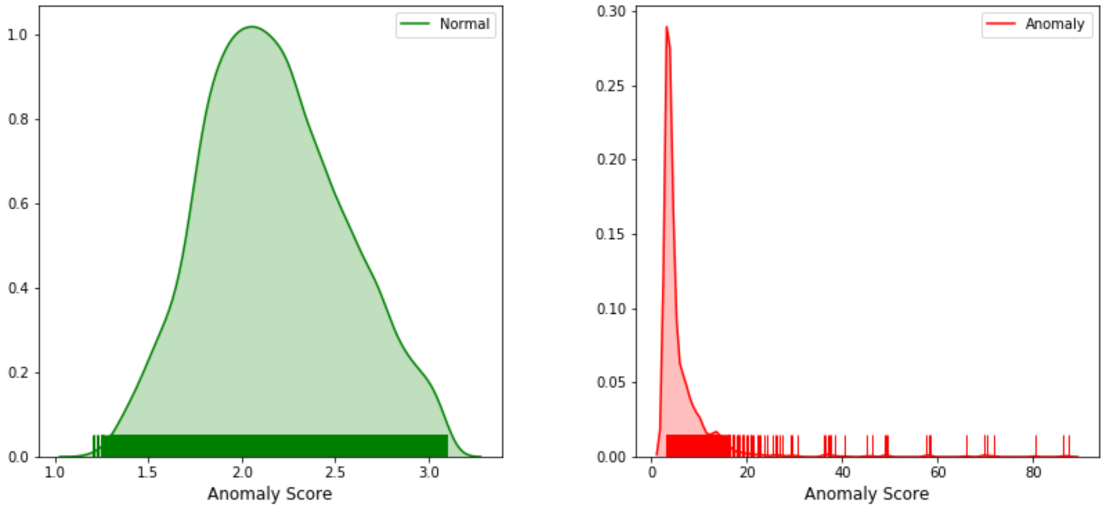
(b) Comparison of generation and real data across variables using TSNE

Figure 5.1: Comparison of generation and real data obtained for different values across 2 variables

5.2.2 Anomaly detection

Figure 5.3 show the distribution of the labels created by the GANS model using the anomaly score from the GANs algorithm (Equation 3.3). Both Figures 5.3 and 5.2 use a 90% threshold to identify between anomalous and normal HCPs. Additionally, Figure

5.2 shows the anomaly detection based on the anomaly score. The distributions of the anomaly score in Figure 5.2 show that both the generator and discriminator loss of the proposed anomaly score is suitable for the classification of normal and anomalous samples. The anomaly scores for the normal HCP consists of values between 1 and 3 while the anomalous points are greater than 3. Results show anomalous HCP has higher anomaly scores due to the feature matching between the normal data and latent space.



(a) Distribution of the anomaly score for normal HCP

(b) Distribution of the anomaly score for anomalous HCP

Figure 5.2: Distribution of the anomaly score for normal and anomalous HCP

5.2.3 Validation of GANs results

The main purpose of this section is to validate and discuss the results, specifically the labels generated from the GANs model against the unsupervised K-Means model.

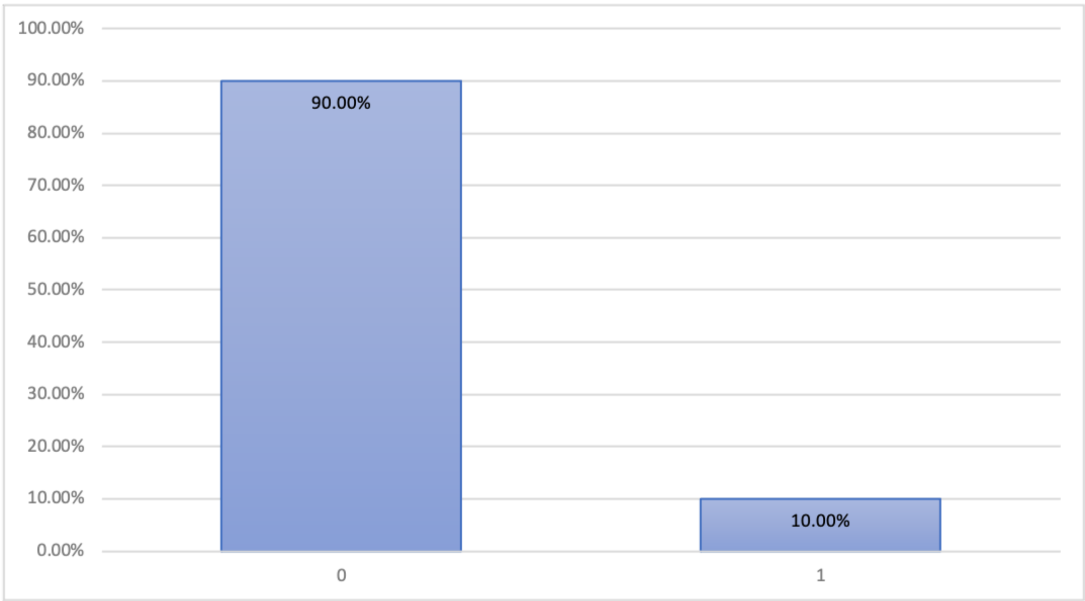


Figure 5.3: Labels created in the GANs model using the anomaly score and a 90% threshold

GANs versus K-Means

The accuracy and efficiency of our GANs model are of importance because the results from this step are used as the ground truth in further steps in achieving the research objective. Figure 5.4 shows the optimal number of clusters in the private data was 6 clusters. We analysed and evaluated our model output against the K-Means algorithm using the confusion matrix and the evaluation metrics illustrated in Section 3.5.

The confusion matrix in Table 5.1 summarises the normal and anomalous HCP between the K-Means and the GANs algorithms. Regarding our experimental results, we clearly see that the proposed GANs algorithm shows similar results when compared to the K-Means algorithm. The results from the confusion matrix show a true positive rate (TPR) and a true negative rate (TNR) of 78.5% and 97.6% respectively. Compared with the GANs model, 78.5% of the HCP were also identified as anomalous by the K-Means model. However, the GANs algorithm has labelled an additional 1 070 HCP labelled as anomalous.

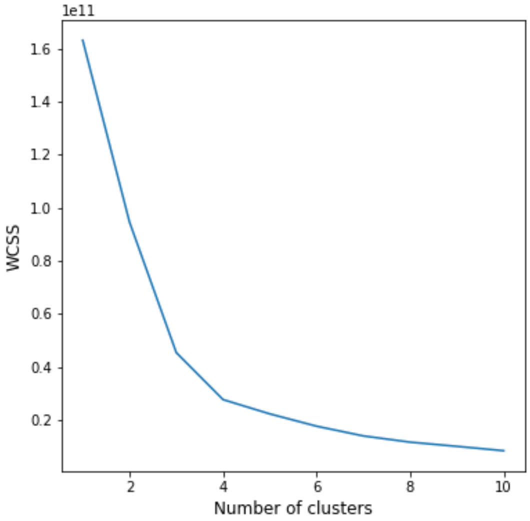


Figure 5.4: K-Means number of clusters on the Medicare data

Table 5.1: Confusion Matrix - GANs versus K-Means

		K-Means	
		Normal	**Anomaly**
GANs	**Normal**	43 832	1 070
	Anomaly	1 070	3 920

5.2.4 Classification Model Interpretation

In the following subsection, the results are based on the model performance from the LR, RF, DT and XGB algorithms. The supervised modelling part of this thesis used 60% of instances for training and 40% used to test the model. Highlighted in Table 5.2, the anomaly class is under-sampled and imbalanced. This means only 10% of the data labelled as anomalous while 90% labelled as normal. To overcome this imbalance of anomaly labels, a minority over-sampling technique was applied to the training data with a sampling strategy of 30%. The minority over sampling automatically increases the anomalous labels from 10% to 23% of the overall training data as shown in Table 5.3.

Figure 5.5 show the results obtained for each K-Fold iteration of FPR and TPR

Table 5.2: Class Distribution - Pre-Sampling

Class	**Count**
Normal	21 530
Anomaly	2 419

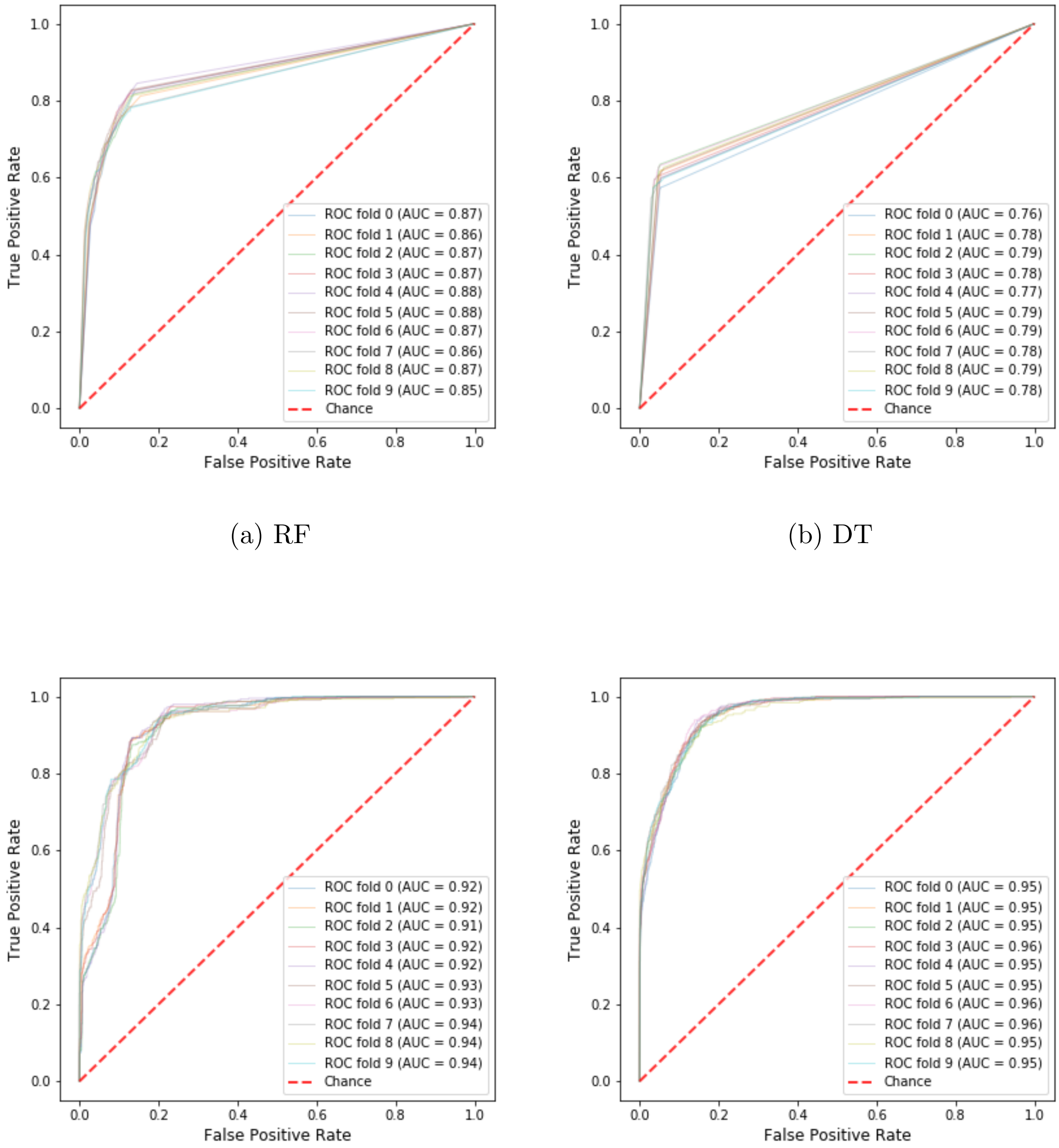
Table 5.3: Class Distribution - Post-Sampling

Class	**Count**
Normal	21 530
Anomaly	6 459

between 0 and 1 for the different classification models. Figure 5.5 highlights the performance across all the models on the Medicare data based on AUC. As previously indicated in Section 3.4, the training and prediction for each model went through a KFold 10 iteration process. The RF model, Figure 5.5a shows the AUC outputs ranging between 85% to 87% per iteration. Similarly DT model, Figure 5.5b shows similar outputs for each iteration AUC ranging from 76% to 79%. The remaining models, LR and XGB show AUC results ranging from 91% to 94% and 95% to 96% respectively.

Across the four models illustrated in Figure 5.6, the accuracy, specificity, sensitivity and AUC were averaged. The performing model based on AUC was the XGB model with 95% followed by the LR model with 93%. Table 5.4 shows the sensitivity rate of the XGB model (69.4%) is lower than the sensitivity rate of the LR model (80.0%). The high sensitivity rate indicates the classification models do a good job in classifying the anomalous labels identified by the GANs model. The LR model has a higher sensitivity rate but the specificity of the XGB model (95.1%) outperforms the LR model (90.0%). The RF outperforms the rest of the models in terms of specificity. The specificity rate reflects the model's ability to correctly classify false negatives.

In the context of the current classification problem, a high sensitivity value is preferred as the sensitivity will be identifying anomalous HCP. The next section will evaluate the key features contributing to the anomaly.



(a) RF

(b) DT

(c) LR

(d) XGB

Figure 5.5: K-Fold AUC results obtained across each of the classification models

Table 5.4: Model results on the Medicare data

Model	Accuracy	Sensitivity	Specificity	AUC
CART	0.9189 ±0.0124	0.592 ±0.021	0.954 ±0.007	0.781 ±0.009
LR	0.892 ±0.021	**0.799 ± 0.074**	0.901 ±0.030	0.926 ±0.008
RF	0.920 ±0.012	0.602 ±0.014	**0.955 ± 0.008**	0.868 ±0.008
XGB	**0.927 ± 0.016**	0.694 ±0.052	0.951 ±0.017	**0.954 ± 0.002**

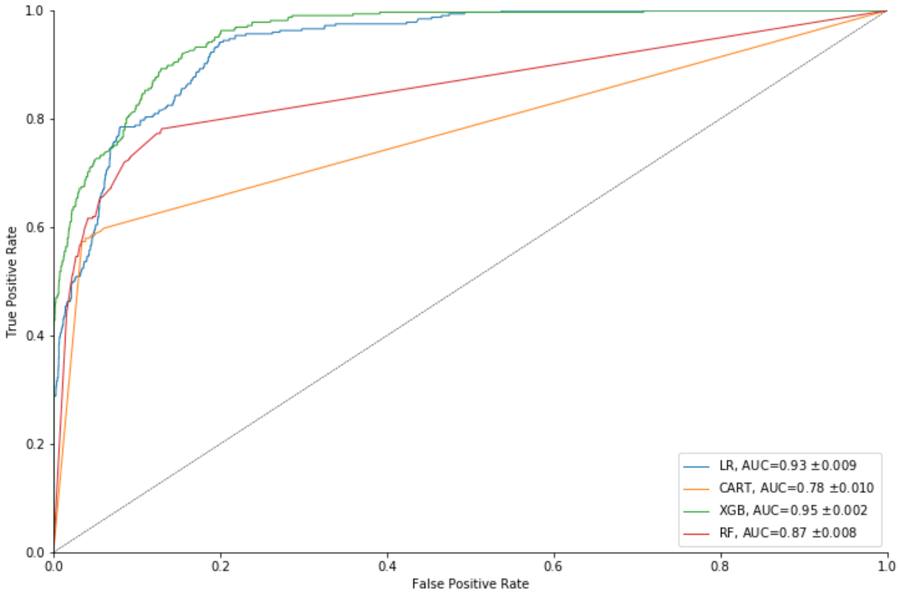


Figure 5.6: The average Receiver Operating Characteristic (ROC) obtained from 10 iterations of FPR and TPR between 0 and 1 for the each classification model

5.2.5 SHAP - Anomaly Feature Analysis

SHAP analysis was conducted to highlight the features that contribute to the anomalous HCP which push the base value to the model output. Figure 5.7 shows features pushing the prediction higher (in red) and those pushing the prediction lower (in blue). Figure 5.7 highlights diagnostic radiologists is one of the major contributors to the predictions. There is a higher average payment across similar practitioner types and injuries have a major impact on the model output. Similarly, the results show a higher average unique count and services across HCP tend to increase the SHAP values while the majority of the lower values decrease the predication value.

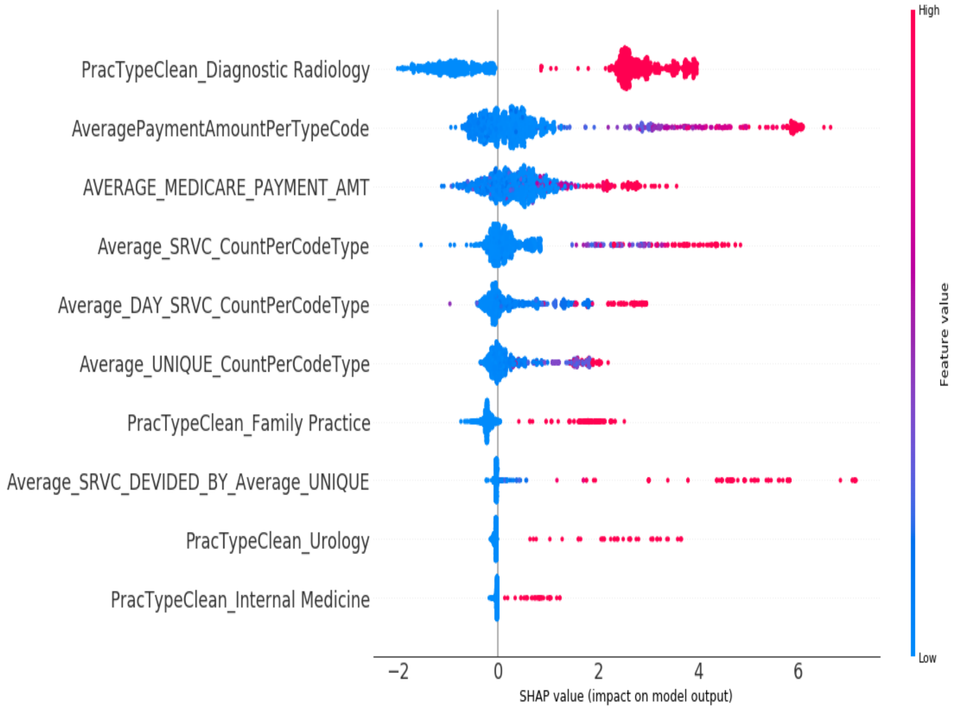


Figure 5.7: The top 10 features that contributes to anomalous HCP obtained by the SHAP values

Figure 5.8 provides a deeper explanation for each sample in the Medicare dataset. The graph shows how each of the top 10 feature contributes to moving the model output from the initial base value to the model output. Features such as the average number of services provided across similar HCP (AverageSRVCCountPerCodeType) contributes 1.05, average unique beneficiaries receiving the service (AverageUniqueCountPerCodeType) contributes 4.8, average unique beneficiary/per day services (AverageDaySrvcCountPerCodeType) contributes 4.17 to the model prediction by changing the base value of -1.5 to the value of 5.9. Alternatively HCP that are Diagnostic Radiologist (PracTypeClean_Diagnostic Radiology) seems to push the model prediction lower.

Figure 5.9 shows the SHAP feature dependence plot, that is, average unique benefi-

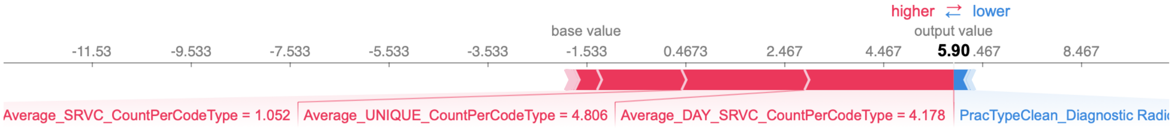


Figure 5.8: Force plot interpreting the features contributing to the anomaly for each sample

ciaries receiving the service across the same HCP interacting with diagnostic radiologists. In cases close to 0, the occurrence of radiologists increases the prediction of anomalies. As the unique number of beneficiaries, increases the occurrences of radiologists reduce. This is not a causal model, so reasons for the increase in unique beneficiaries might be due to other reasons such as visits to family practitioners.

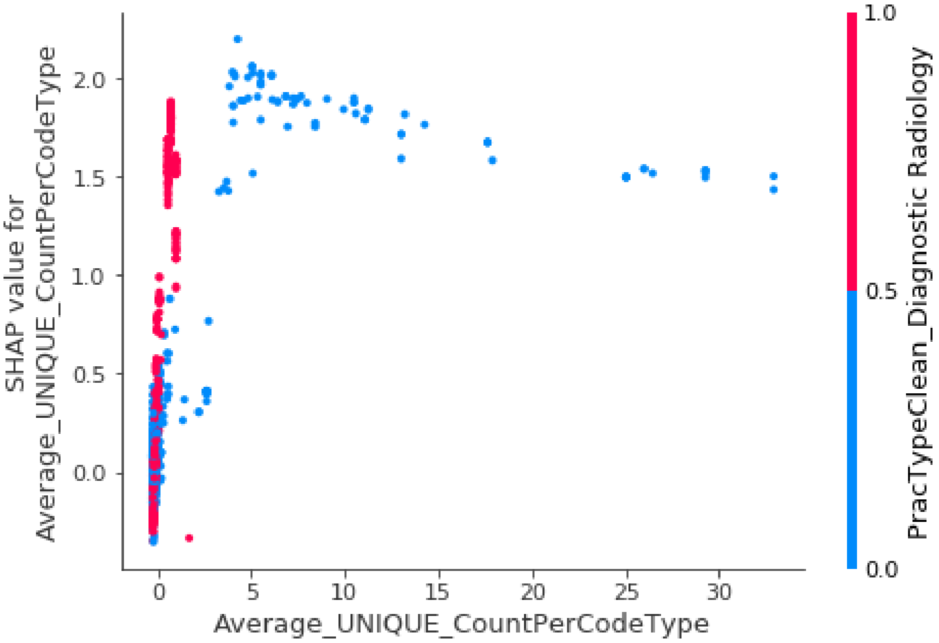
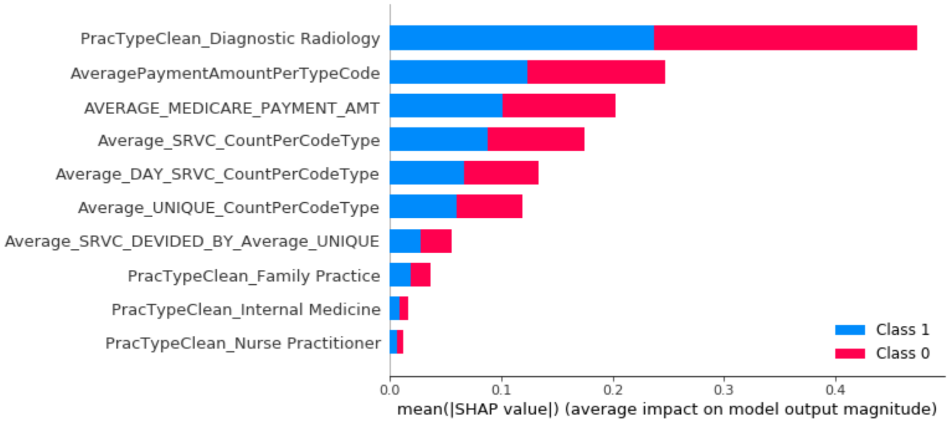
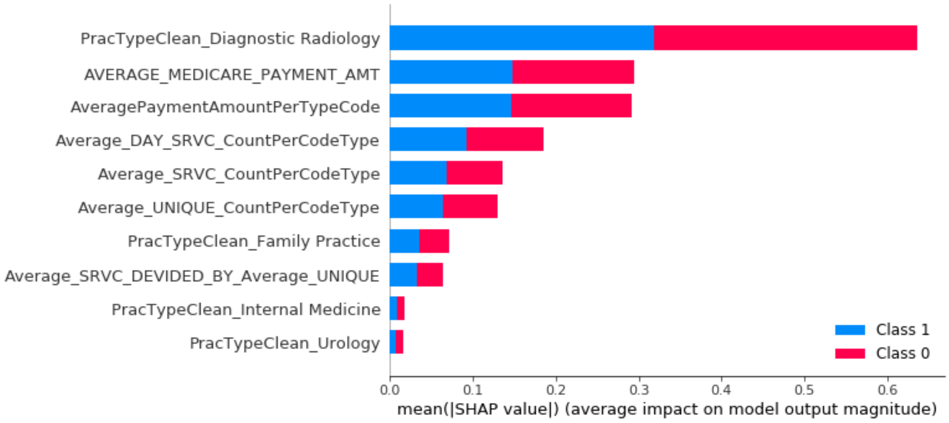


Figure 5.9: Feature Dependency Plot obtained for different combinations of AverageUniqueCountPerCodeType and AverageUniqueCountPerCodeType based on PracTypeClean-Diagnostic Radiology visits

Figure 5.10 shows a summarised overview of the top 10 important features across the four classification models. Across all four models, Diagnostic Radiology has proven to be the most important feature. Features such as PracTypeClean-Diagnostic Radiology, AveragePaymentAmountPerTypeCode, AverageMedicarePaymentAmt, AverageSRVC-CountPerCodeType, AverageDaySrvcCountPerCodeType, AverageUniqueCountPerCode-Type, AverageUniqueServicesCountPerCodeType were consistently part of top ten important features in detecting anomalies across all the models. Features such as PracType_FamilyPractice, PracType_InternalMedicine, PracType_NursePractitioner, PracType_Urology, PracType_OrthopedicSurgery are the rest of the important features which are not consistent in all of the four classification models.

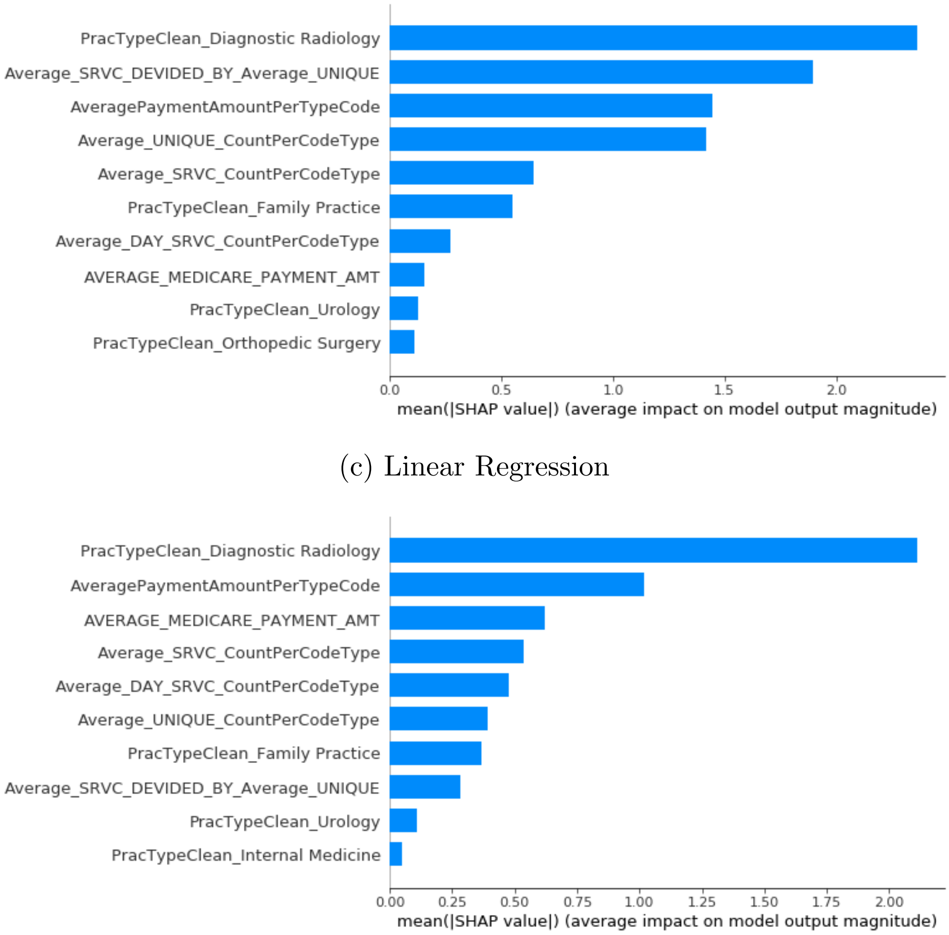


(a) RF



(b) DT

Figure 5.10: The feature importance across the various classification models



(c) Linear Regression

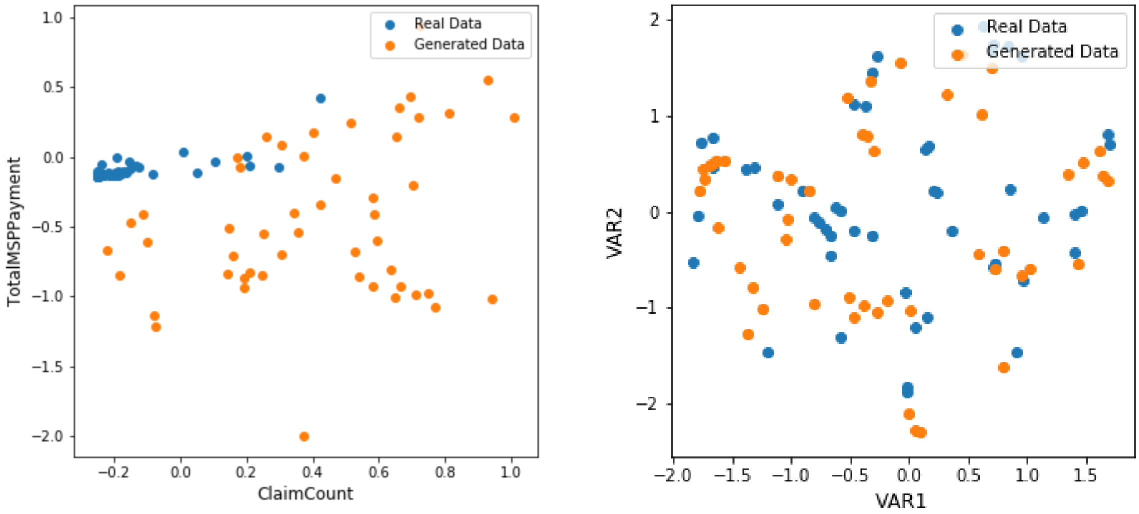
(d) XGB

Figure 5.10: The feature importance across the various classification models

5.3 Private results

5.3.1 Generation of realistic data

The trained GANs model is generated against the 2018 real data across the different features. The data distribution for the generated data and real data is represented across the two features $TotalMSPPayment$ and $ClaimCount$ in Figure 5.11a. For 300 iterations or epochs, the distribution of the generated data across these two features shows an even distribution across the values -0.2 and 1, whereas the real dataset seems to be clustered around -0.2 and 0.2. Furthermore, across the TSNE reduced dataset shown in Figure 5.11b, the generated data is evenly distributed. The results in Figure 5.11 infers that the GANS model, specifically the generator, is suitable for generating data that can represent real data.



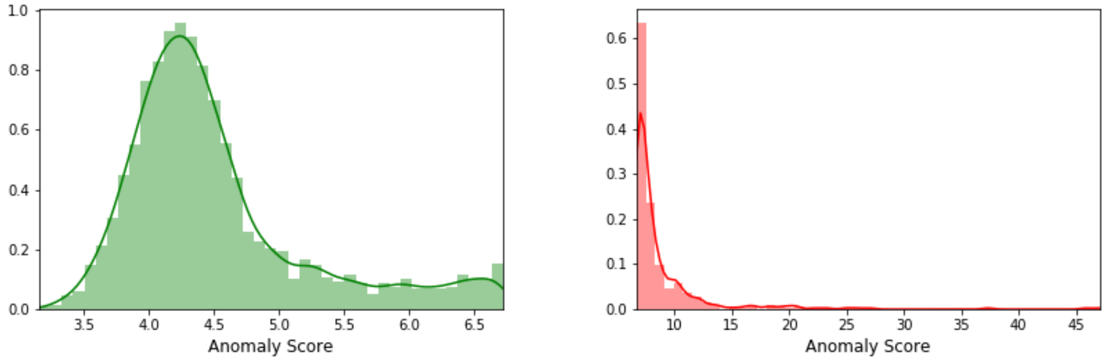
(a) Comparison of generation and real data across variables without using TSNE

(b) Comparison of generation and real data across variables using TSNE

Figure 5.11: Comparison of generation and real data obtained for different values across 2 variables

5.3.2 Anomaly detection

The detection of anomalous HCP was determined by the anomaly score and the application of a 90% threshold. Figure 5.12 shows the distribution of the anomaly score on the normal and anomalous HCP and indicates a normal distribution of scores ranging between 3 and 7. In contrast to the scores of normal HCP, the graph on the right of Figure 5.12 indicates anomalous HCP has higher scores. The results in Figure 5.12 shows that the majority of the anomaly scores range between 9 and 15. It is expected that anomalous HCP has much higher scores due to the feature matching between the normal data and latent space.



(a) Distribution of the anomaly score for normal HCP

(b) Distribution of the anomaly score for anomalous HCP

Figure 5.12: Distribution of the anomaly score for normal and anomalous HCP

5.3.3 Validation of GANs results

In this section, the results generated from the GANs model are discussed and validated against the unsupervised K-Means algorithm. The second part is unique to the private data where the results from the GANs model are shared with business domain experts.

GANs versus K-Means

Figure 5.13 shows the optimal number of clusters in the private data was 6 clusters. We analysed and evaluated our model output against the K-Means algorithm using confusion matrix and evaluation metrics illustrated in Section 3.5.

Table 5.5 summarised the normal and anomalous HCP between the K-Means and the GANs algorithms. In the current context, the results from the GANs algorithm is assumed to be the ground truth. With reference to our experimental results, we clearly see that the proposed GANs algorithm shows similar specificity results when compared to the K-Means technique. The confusion matrix shows a true positive rate (TPR) and

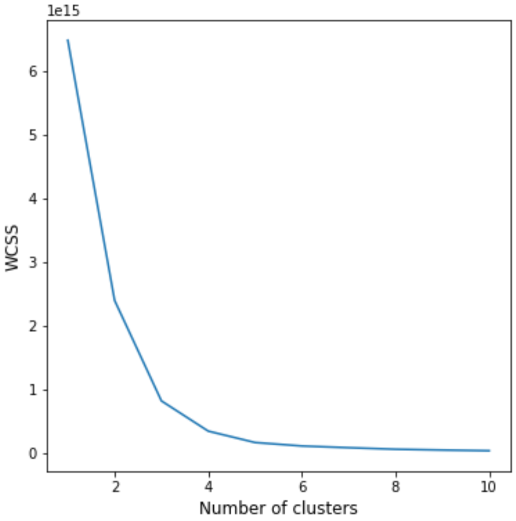


Figure 5.13: K-Means number of clusters on the private data

a true negative rate (TNR) of 33.3% and 92.5% respectively. Comparing to the GANs model, 33.3% of the HCP were also identified as anomalous by the K-Means model. However, the GANs algorithm has labelled an additional 238 HCP as being anomalous.

Table 5.5: Confusion Matrix - GANs versus K-Means

		K-Means	
		Normal	**Anomaly**
GANs	**Normal**	2 972	238
	Anomaly	238	119

5.3.4 Validation of GANs labels

The GANs model generated normal and anomaly labels from the private data and these labels were shared with business domain experts to validate our results. The results from the GANs algorithm contained a total of 3 567 records. Figure 5.14 shows the split between normal and anomalous records were 3 210 (89.9%) and 357 (10.01%) respectively.

Shown in Table 5.6 is the comparison between the GANs results against feedback

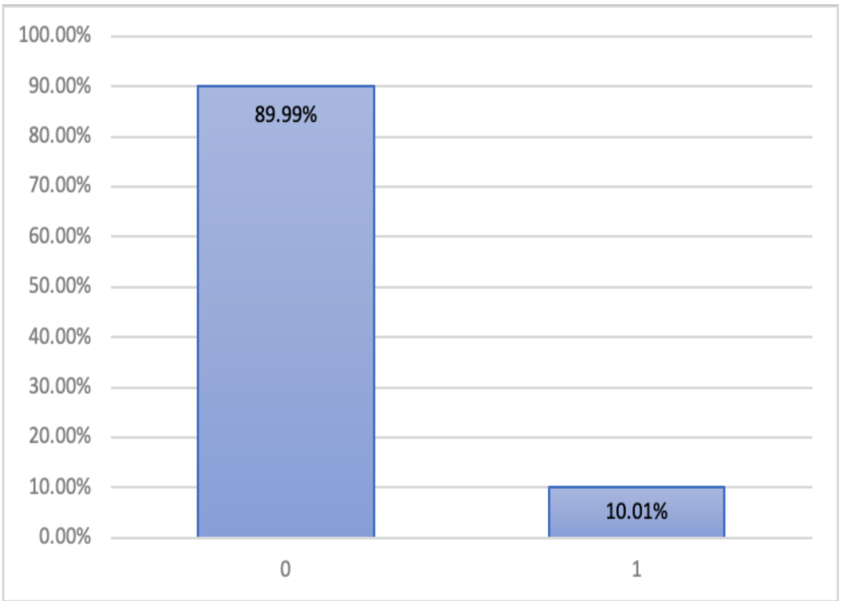


Figure 5.14: Anomaly class distribution obtained for normal HCP and anomalous HCP represented by 0 and 1 respectively

from business experts. The results show that the TPR and TNR are 81.23% and 99.16% respectively. Furthermore, the results show a false positive rate of 18.77% which indicates that the GANs model incorrectly classified 67 HCP as being anomalous. Usually, the major concern for the application of anomaly detection systems or models is to try to minimise the false positive rate. High levels of false-positives results within a claims administration company can result in sending this HCP for investigation. This often results in payments not being authorised. Based on the results in Tables 5.7 and 5.7 there is a possibility of some private organisations overlooking the false positive and negative rates due to their risk appetite and low volumes.

Table 5.6: Confusion Matrix - GANs versus Actual (Percentage of HCP)

		Actual	
		Normal	**Anomaly**
GANs	**Normal**	99.16%	0.84%
	Anomaly	18.77%	81.23%

Table 5.7: Confusion Matrix - GANs versus Actual (Number of HCP)

		Actual	
		Normal	**Anomaly**
GANs	**Normal**	3,183	27
	Anomaly	67	290

Feedback from the business experts was gathered and summarised highlighting some of the key reasons why a HCP was labelled as anomalous. From the results presented in Figure 5.15, and based on the data submitted, the following observations were made that could require in-depth investigations:

1. The majority of HCP labelled as anomaly seemed to send medical invoices more than 3 months after the treatment of the claimant.

2. There has also been evidence that 86 HCPs have submitted the same line items multiple times. This could potentially imply that the claimant is being over-serviced by the HCP.

3. There are many cases whereby HCP treat claimants over a weekend when the incidents occurred during a weekday. This can result in higher weekend rates being charged and billed to the claims insurance administration companies.

5.3.5 Classification Model Interpretation

The supervised modelling process used 60% of instances for training and the remainder in test (40%). Highlighted in Table 5.8, the anomaly class is under-sampled. To overcome this imbalance of anomaly labels, a minority over sampling technique was applied on the training data. The minority over-sampling automatically balances the minority class (anomaly) with the majority class (normal). The updated distribution of the training dataset is represented in Table 5.9.

The results were based on a ten-fold cross validation process across the LR, RF, DT and XGB algorithms. Table 5.10 shows the average results from the cross-validation process across each of the supervised classification-based models.

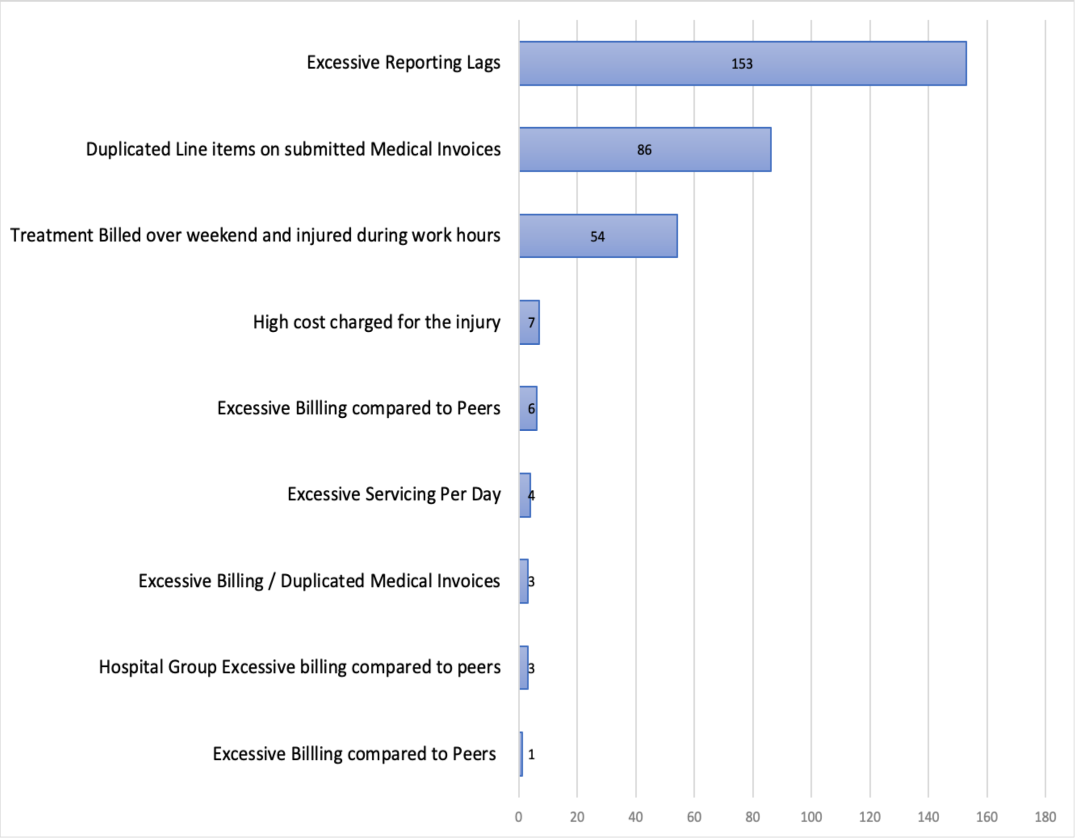


Figure 5.15: Summary of the number of cases obtained for each anomaly reason

Table 5.8: Class Distribution - Pre-Sampling

Class	Count
Normal	1 934
Anomaly	206

Table 5.9: Class Distribution - Post-Sampling

Class	Count
Normal	1 934
Anomaly	1 934

Figure 5.16 shows the performance across all the models on the private data based on AUC. Similar to the Medicare process, the training and prediction for each model went

Table 5.10: Model results on the private data

Model	Accuracy	Sensitivity	Specificity	AUC
CART	0.958 ±0.037	0.743 ±0.081	0.979 ± 0.015	0.861 ±0.018
LR	0.954 ±0.006	**0.942 ± 0.058**	0.953 ±0.023	0.987 ±0.016
RF	0.959 ±0.018	0.763 ±0.101	0.977 ±0.013	0.98 ±0.014
XGB	**0.963 ± 0.004**	0.932 ±0.044	**0.981 ± 0.017**	**0.992 ± 0.014**

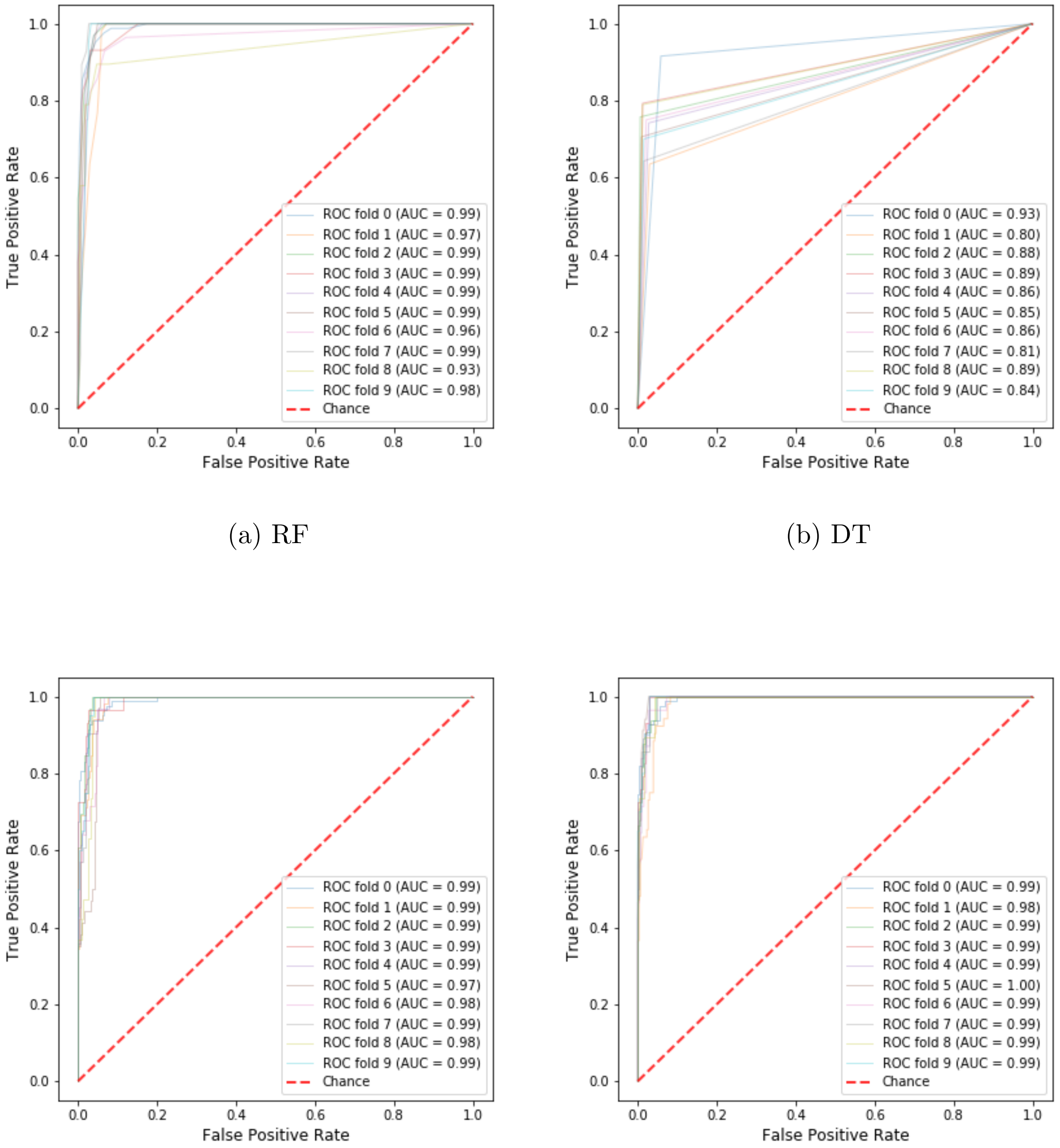
through a 10-iteration process. The RF model, Figure 5.16a shows the AUC outputs ranging between 80% to 93% per iteration. The DT model, Figure 5.16b shows majority of outputs ranging between 80% and 89% for each iteration AUC. The remaining models, XGB and LR show similar AUC results and behaviour of the ROC curve.

Figure 5.17 shows the averaged results from the 10 iterations with the AUC across all the models greater than 86%. The best performing model based on AUC was XGB with 99% followed by DT with 99%. The sensitivity rate in Table 5.10 shows the LR model at 94.2% which is slightly higher when compared with the XGB model of 93.2%. The high sensitivity rate indicates the classification models perform optimally in classifying the anomalous labels identified by the GANs model. The specificity of the XGB model (98.1%) performs better than the DT model (97.9%), followed by the RF model (97.7%) and thereafter the LR model (95.3%). The specificity rate reflects the models' ability to correctly classify the HCP that is not anomalous.

In the context of the current classification problem, a high sensitivity value is preferred as the sensitivity will be correctly identifying anomalous HCP. The next section will evaluate the key features contributing to the model.

5.3.6 SHAP - Anomaly Feature Analysis

Figure 5.18 represents the feature's importance of anomalous HCP from the XGB model with top five SHAP values for AvgMSPReportingLag, AvgMSPPayment, Groupings_Hospital, TotalMSPPayment and WeekendTreatmentUniqueCount. In Figure 5.18 shows high values for features such as AvgMSPReportingLag, AvgMSPPayment, Groupings_Hospital and TotalMSPPayment are influencing the model output. In contrast, the model output



(a) RF

(b) DT

(c) LR

(d) XGB

Figure 5.16: The average Receiver Operating Characteristic (ROC) obtained from 10 iterations of FPR and TPR between 0 and 1 for the each classification model

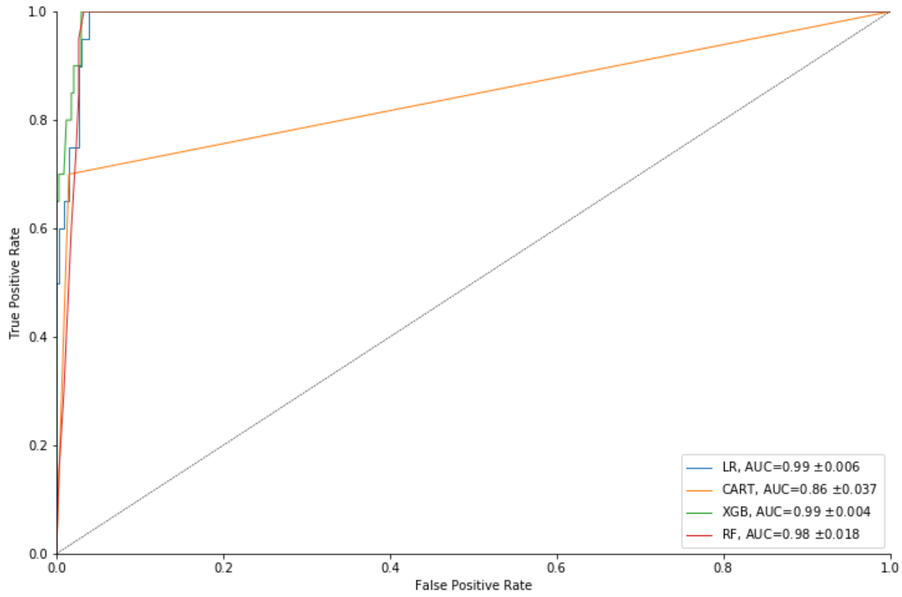


Figure 5.17: The average Receiver Operating Characteristic (ROC) obtained from 10 K-Fold iteration of FPR and TPR between 0 and 1 for the each classification model

is deceased by a high number of unique claimants being treated over the weekend (WeekendTreatmentUniqueCount), low reporting lag from HCP (AvgMSPReportingLag) and average healthcare payments (AvgMSPPayment).

Figure 5.19 provides a deeper explanation for each sample in the private dataset. The graph shows how each feature contributes to moving the model output from the base value (the average model output over the training dataset we passed) to the model output. Similarly, to the explanation in Section 5.2.5, features such as an average of unique treatments during the week (WeekdayTreatmentUniqueAvg), the count of unique treatments during the weekend (WeekendTreatmentUniqueCount), the number of claims treated and total HCP payment (TotalMSPPayment) has a positive impact on pushing the model prediction higher. Alternatively, features like average payment (AvgMSPPayment) and average reporting lag (AvgMSPReportingLag) by HCP pushes the model prediction lower.

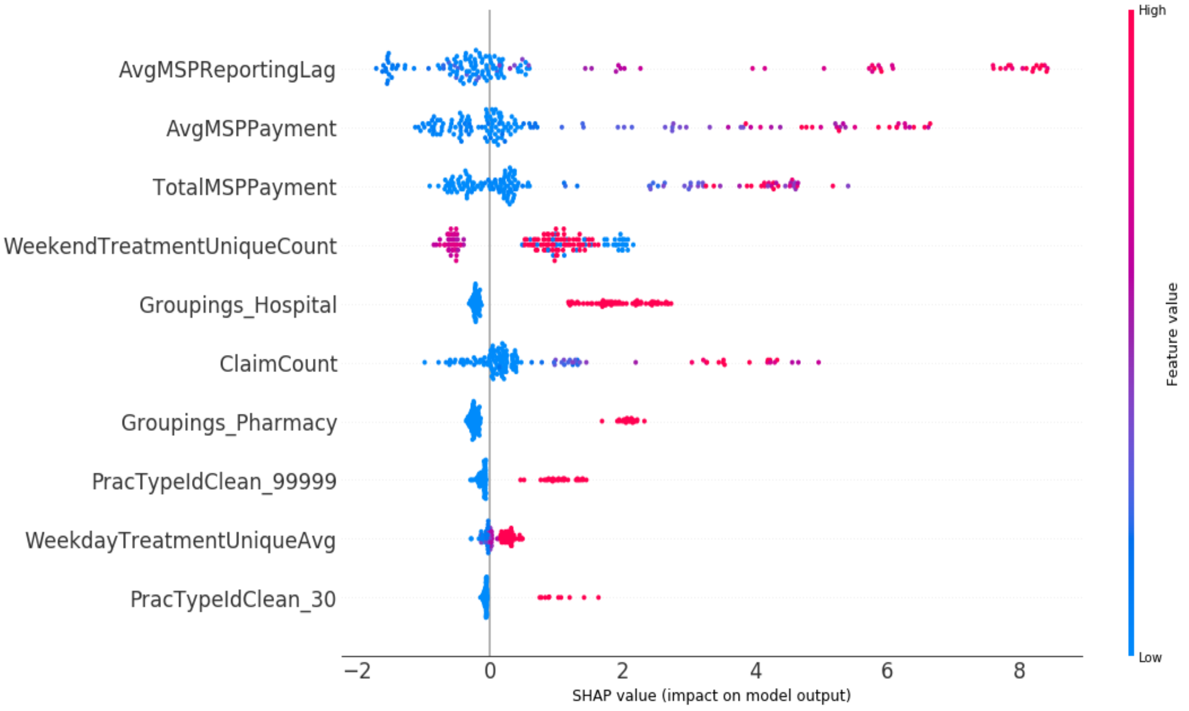


Figure 5.18: The top 10 features that contributes to anomalous HCP obtained by the SHAP values

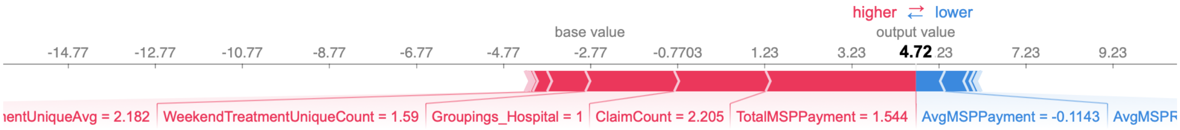


Figure 5.19: Force plot interpreting the features contributing to the anomaly for each sample

Figure 5.20 shows the effect the average reporting lag of HCP (AvgMSPReportingLag) has on the prediction. We plotted the SHAP value of AvgMSPReportingLag against hospitals' (Groupings_Hospitals) SHAP values in the dataset. We learned that a change in the average payment of HCP has a great impact on the detection of anomalies across both on hospitals and non-hospitals.

Figure 5.21 shows a summarised overview of the top 10 important features across the four classification models. Features such as AvgMSPReportingLag, AvgMSPPayment, TotalMSPPayment, ClaimCount, WeekdayTreatmentUniqueAvg, WeekendTreat-

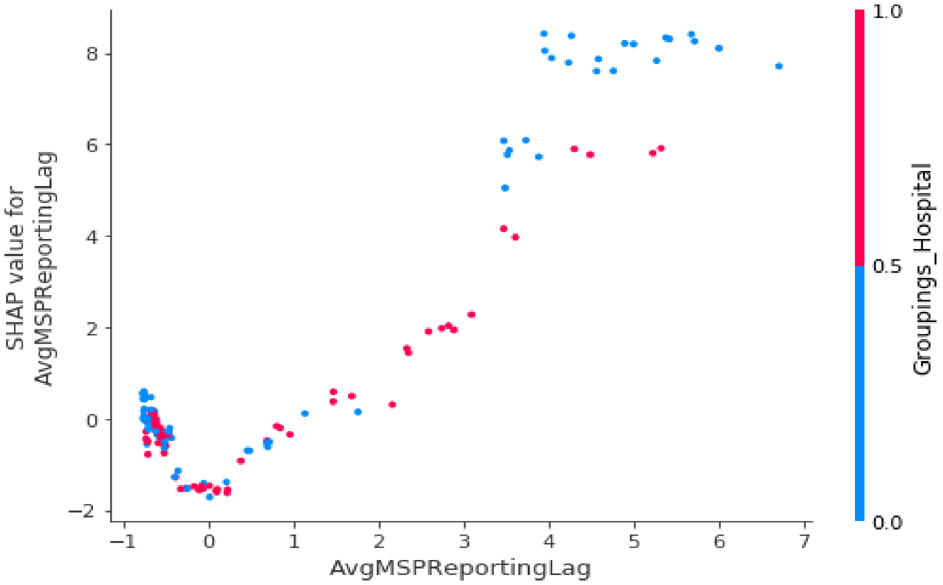
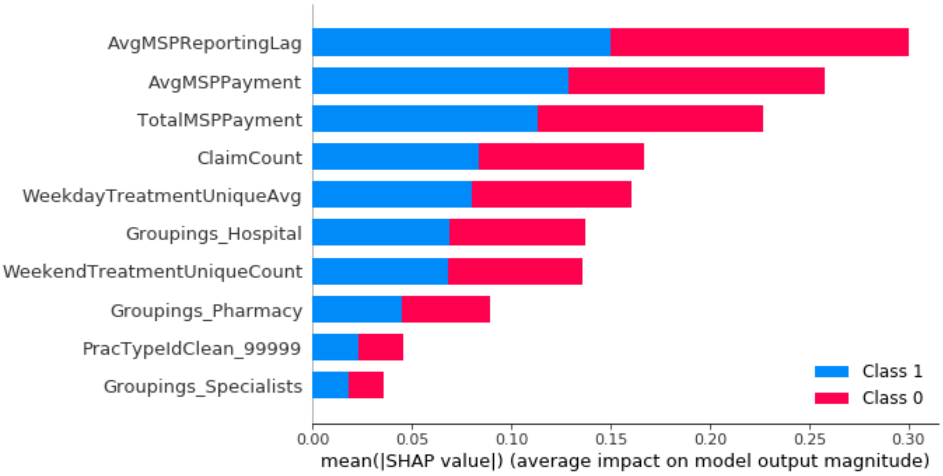


Figure 5.20: Feature Dependency Plot obtained for different combinations of AvgMSPReportingLag based on Hospital visits

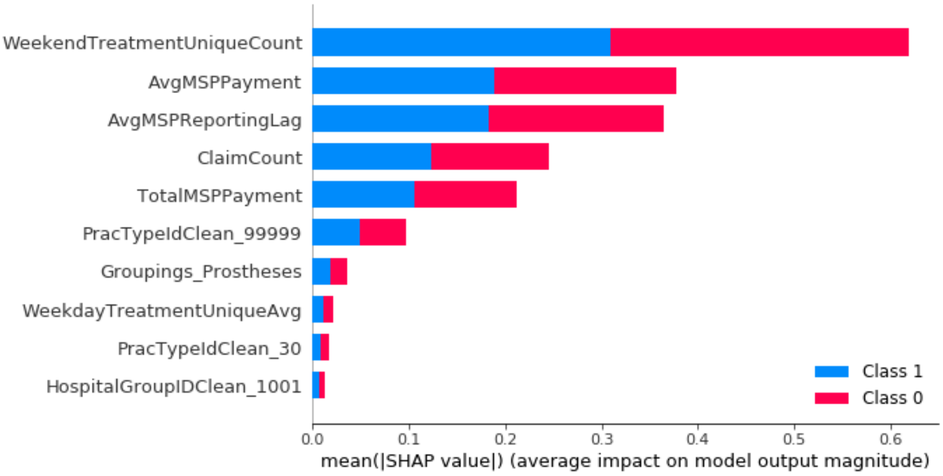
mentUniqueCount, PracTypeIdClean_9999 were consistently in the top 10 important features across all the models. Features such as Groupings_Hospital, Groupings_Pharmacy, Groupings_Specialists, Groupings_Prostheses, PracTyeIdClean_30, HospitalGroupIdClean featured as important in some of the models.

5.4 Chapter Summary

In summary of this chapter, it has been shown that GANs are applicable in generating of labels by the use of an anomaly score. Furthermore, the results were presented according to the two different datasets used in the experiments. Across the Medicare data, the classification models were evaluated with the XGB model showing high results based on the sensitivity and specificity rates. The SHAP results further highlighted diagnostic radiology, higher payment across similar injuries, and higher payments from the HCP impact model out for anomalous providers.

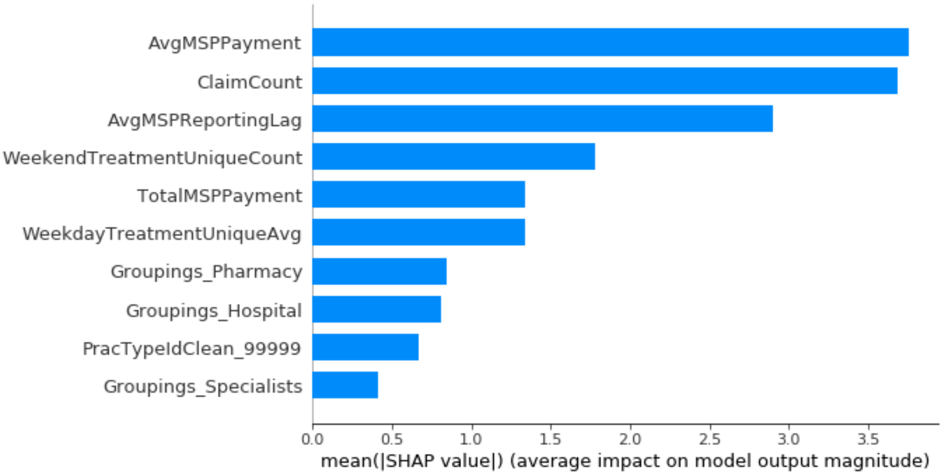


(a) RF

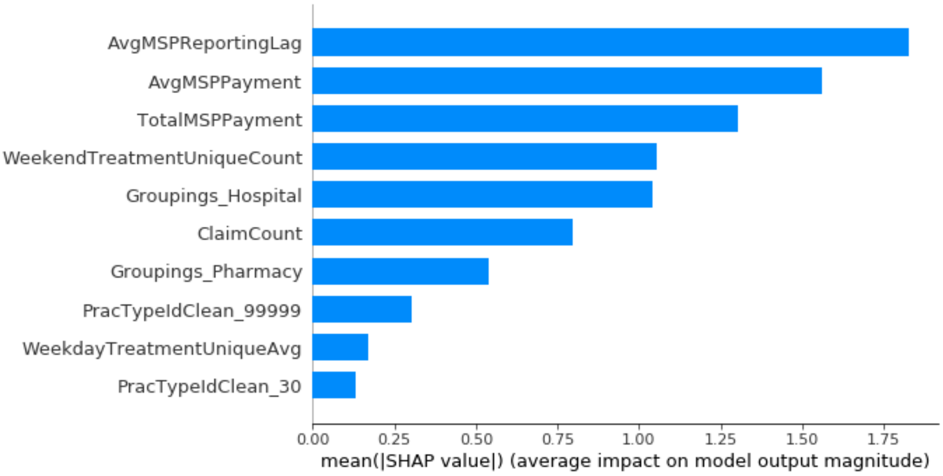


(b) DT

Figure 5.21: The feature importance obtained for SHAP values for the RF, DT, LR and XGB models



(c) Linear Regression



(d) XGB

Figure 5.21: The feature importance obtained for SHAP values for the RF, DT, LR and XGB models

Chapter 6

Conclusion

This chapter concludes the thesis by summarising the main findings and contributions in line with the research objectives in Section 6.1, and proposes topics for future research in Section 6.2.

This thesis addressed the anomaly detection challenge, specifically identifying anomalous HCP without having labelled data. It proposed a model that can identify anomalous HCP across the various healthcare types and services provided. The research looked at payment transactions received from HCP for services rendered to claimants based on their injuries or conditions. These transactions contained variables that provide general information ranging from injury information, HCP information, claim information, and the value of the transaction. Besides looking at the variables initially collected, additional contextual variables were created to assist the model building process in detecting anomalous HCP.

Thereafter, a number of experiments were conducted, using a combination of unsupervised deep learning models and supervised classification models to detect anomalies. These models were applied to two different datasets to create an approach that can be generalised to automatically detecting anomalies and provide insights into the variables that are responsible for the anomalies.

The model building process followed a systematic process, whereby the first step was to collect, clean and transformed the data. Thereafter, the required features were chosen which formed part of the modelling process. Secondly, an unsupervised GANs model was applied to the data to identify anomalies through the application of an anomaly score. Lastly, different supervised classification models were trained and evaluated to propose the best model that can be used to identify features contributing to the anomalies. These features were explained using the SHAP models which highlighted the features that contributed the most to anomalous HCP.

6.1 Summary of the Research Problem

The objective of this study was to define a model that can detect HCP without having labels. Furthermore, a secondary objective was to give context to why the model labelled HCP anomalous. These two main objectives were achieved by answering the following research questions:

6.1.1 Can a deep learning algorithm detect HCP relating to fraud and cost abuse without having labels?

The first objective of the study was to investigate a deep learning algorithm specifically GANs and to determine the ability of a GANs model in providing labels. The results from Sections 5.2.2 and 5.3.2 show across the two datasets that GANs can generate labels to determine anomalous HCP. The current approach used an unsupervised approach and through the use of an anomaly score, we were able to generate normal or anomalous labels. Apart from the generation of labels GANs model can generate data.

6.1.2 Can a machine learning model interpret the reasons for anomaly detection?

Section 2.6 provided a theoretical perspective of a method to interpret anomalous HCP. The application of the SHAP framework in Section 5.2.5 and 5.3.6 shows that SHAP values across each of the features. The SHAP technique has proven to explain the features contributing to the anomalies ranging from the difference between the average prediction to the prediction of the instance. These SHAP values can form a basis in which we can interpret why a model detected anomalous HCP.

6.1.3 Which features explain the reasons the anomalous HCP

Section 4 showed the features in the both sets of data that includes basic HCP payment information details on HCP, and claimant information. The SHAP results from the second part of the study showed the features that contributed to identifying anomalous HCP. These features from the Medicare dataset that contributed to the anomaly detection were:

- The average amount that Medicare paid for a treatment service (AverageMedicarePaymentAmt)
- The average amount that Medicare paid across the same healthcare type and medical service (AveragePaymentAmountPerTypeCode)
- The average number of services provided across the same HCP and medical service (AverageSRVCCountPerCodeType)
- The average unique beneficiary per day services across the same HCP and medical service (AverageDaySrvcCountPerCodeType)
- The average unique beneficiaries receiving the service across the same HCP and medical service (AverageUniqueCountPerCodeType)
- The average number of services received by each distinct Medicare beneficiaries (AverageUniqueServicesCountPerCodeType)

The important features from the private dataset which contributed to anomalous HCP were:

- The average reporting variance between service and reporting date across HCP with the same injury, injury severity, healthcare type and healthcare group (AvgMSPReportingLag).
- The average payment across HCP with the same injury, injury severity, healthcare type and healthcare group (AvgMSPPayment).

- The number of claims across HCP with the same injury, injury severity, healthcare type and healthcare group (ClaimCount).
- The sum of the payments across HCP with the same injury, injury severity, healthcare type and healthcare group (TotalMSPPayment)
- Total unique number of claimants the HCP have treated over the weekend (WeekendTreatmentUniqueCount)
- The total number of unique claimants the HCP have treated from Monday to Friday (WeekdayTreatmentUniqueCount)

Overall, across the two datasets, there were common features that contributes in detecting anomalous HCP. These features included average payments and number of treatments across the same HCP and injury.

6.2 Future Work

This thesis yielded some interesting research directions that can be explored in future. The proposed research topics are discussed below.

6.2.1 GAN Architectures

The first step of the modelling only used GANs to identify labels that identified normal versus anomalous HCP. Future research can look at alternative GAN architectures such as deeper networks or different hyperparameters to improve the classification accuracy of anomalies. Additionally, we can use a different loss function such as WGAN which applies the earth mover distance to detect anomalies. A different distance function such as the Earth-Mover (also called Wasserstein) distance can be an alternative to solving the training challenge. It can also improve the detection of anomalies. Additionally, architectures like the BiGANs which incorporate an encoder that maps the latent space can be used. Since the features in the latent space can yield important information about the data, it can also be an alternative to the anomaly score

6.2.2 Reconstruction Error

The anomaly score defined in Equation 3.3 is calculated using the feature matching method of the discriminator and the generator loss. The feature matching method is the reconstruction error between the original sample and the preceding layer in the discriminator. The one challenge with this approach is, if one sample in a specific feature contain an outlier, it can increase the anomaly score of that feature. An alternative approach to reconstruction error is an interesting problem to solve for future research.

6.2.3 Additional data and features

As indicated in Section 4.2 the private dataset used in the experiment was extracted from a claims administration organisation. The data consisted of payment transactions made to various types of HCP for services rendered. The approach of using GANs to generate anomalous or normal labels and SHAP to interpret the anomalies can be applied to a range of healthcare systems around the world. In the future, we want to extend the proposed design across similar healthcare data from other claim administration organisations across both the public and private sectors. Additional features can be explored to potentially improve the model outputs, for example, geo-location coordinates of a claimant's place of work and HCP offices. We can identify patterns between claimants visiting HCP living close by versus claimants travelling long distances. These features can aid in detecting collusion between claimants and HCP.

6.2.4 Alternate Distance Metrics

The GANs model uses Jensen-Shannon divergence as a distance function and can lead to training difficulty. A different distance function such as the Earth-Mover distance can be an alternative in solving the training challenge and also improve the detection of anomalies.

In closing, this study was successful in creating normal versus anomalous labels across the different HCP. The GANs model is versatile and can be adapted for future problems where there is an absence or imbalance of labels. Furthermore, a proposed approach through the use of SHAP values can also give context to the anomalous HCP.

Bibliography

[1] Samet Akcay, Amir Atapour-Abarghouei, and Toby P. Breckon. *GANomaly: Semi-Supervised Anomaly Detection via Adversarial Training.* PhD thesis, 2018.

[2] Adedotun Akintayo, Kin Gwn Lore, Soumalya Sarkar, and Soumik Sarkar. Early Detection of Combustion Instabilities using Deep Convolutional Selective Autoencoders on Hi-speed Flame Video. *SIGKDD*, 2016.

[3] Hussain Alkharusi. Categorical Variables in Regression Analysis: A Comparison of Dummy and Effect Coding. *International Journal of Education.*

[4] Tsatsral Amarbayasgalan, Bilguun Jargalsaikhan, and Keun Ryu. Unsupervised Novelty Detection Using Deep Autoencoders with Density Based Clustering. *Applied Sciences*, page 1468, 2018.

[5] M. Ancona, C. Öztireli, and M. Gross. Explaining deep neural networks with a polynomial time algorithm for Shapley values approximation. *36th International Conference on Machine Learning, ICML 2019*, 2019-June:400–409, 2019.

[6] Liat Antwarg, Bracha Shapira, and Lior Rokach. Explaining Anomalies Detected by Autoencoders Using SHAP. *SIGIR 2019 - Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1281–1284, 2019.

[7] D.B. Araya, K. Grolinger, H. F. ElYamany, M.A.M. Capretz, and G Bitsuamlak. An ensemble learning framework for anomaly detection in building energy consumption. *Energy and Buildings*, 2017.

[8] M Arjovsky, S Chintala, and L Bottou. Wasserstein GAN. 2017.

[9] Preeti Arora, Deepali, and Shipra Varshney. Analysis of K-Means and K-Medoids Algorithm for Big Data. *Physics Procedia*, pages 507–512, 2016.

[10] Richard A Bauder and Taghi M Khoshgoftaar. The Detection of Medicare Fraud Using Machine Learning Methods with Excluded Provider Labels. *The Thirty-First International Florida Artificial Intelligence Research Society Conference*, pages 404–409, 2017.

[11] Andreas Bayerstadler, Linda van Dijk, and Fabian Winter. Bayesian multinomial latent variable modeling for fraud and abuse detection in health insurance. *Insurance: Mathematics and Economics*, 71:244–252, 2016.

[12] Amanda Berg, Jörgen Ahlberg, and Michael Felsberg. Unsupervised Learning of Anomaly Detection from Contaminated Image Data using Simultaneous Encoder Training. 2019.

[13] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection. *ICLR*, pages 1–13, 2018.

[14] Andrey Bondarenko, Ludmila Aleksejeva, Vilen Jumutc, and Arkady Borisov. Classification Tree Extraction from Trained Artificial Neural Networks. *Procedia Computer Science*, pages 556–563, 2016.

[15] Andrea Borghesi, Andrea Bartolini, Michele Lombardi, Michela Milano, and Luca Benini. Anomaly Detection using Autoencoders in High Performance Computing Systems. 2018.

[16] A. Bosman, A. Engelbrecht, and M. Helbig. *Fitness Landscape Analysis of Weight-Elimination Neural Networks*. 2018.

[17] Adrian Bănărescu. Detecting and Preventing Fraud with Data Analytics. *Procedia Economics and Finance*, pages 1827–1836, 2015.

[18] Luiz F.M. Carvalho, Carlos H.C. Teixeira, Wagner Meira, Martin Ester, Osvaldo Carvalho, and Maria Helena Brandao. Provider-Consumer Anomaly Detection for Healthcare Systems. *Proceedings - 2017 IEEE International Conference on Healthcare Informatics, ICHI 2017*, pages 229–238, 2017.

[19] P. Cerda and G. Varoquaux. Encoding high-cardinality string categorical variables. pages 1–16, 2019.

[20] Raghavendra Chalapathy and Sanjay Chawla. Deep Learning for Anomaly Detection: A Survey. pages 1–50, 2019.

[21] Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. Robust, Deep and Inductive Anomaly Detection. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10534 LNAI:36–51, 2017.

[22] Varun Chandola, ARINDAM BANERJEE, and VIPIN KUMAR. Survey of Anomaly Detection. *ACM Computing Survey (CSUR)*, 41(3):1–72, 2009.

[23] Sanjay Chawla and Aristides Gionis. *k -means–: A unified approach to clustering and outlier detection.* PhD thesis, 2013.

[24] Jidong Chen, Ye Tao, Haoran Wang, and Tao Chen. Big data based fraud risk management at Alibaba. *The Journal of Finance and Data Science*, pages 1–10, 2015.

[25] CMS. CMS: Research, Statistics, Data and Systems, 2014.

[26] Lucas Deecke, Robert Vandermeulen, Lukas Ruff, Stephan Mandt, and Marius Kloft. Image anomaly detection with generative adversarial networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11051 LNAI:3–17, 2019.

[27] Federico Di Mattia, Paolo Galeone, Michele De Simoni, and Emanuele Ghelfi. *A Survey on GANs for Anomaly Detection.* PhD thesis, 2019.

[28] Asimenia Dimokranitou, Gavriil Tsechpenakis, Jiang Yu Zheng, and Mihran Tuceryan. Adversarial Autoencoders for Anomalous Event Detection. 2017.

[29] Rémi Domingues, Maurizio Filippone, Pietro Michiardi, and Jihane Zouaoui. A comparative evaluation of outlier detection algorithms: Experiments and analyses. *Pattern Recognition*, pages 406–421, 2018.

[30] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial Feature Learning. *ICLR*, (2016):1–18, 2016.

[31] Prajna Dora and G Hari Sekharan. Healthcare Insurance Fraud Detection Leveraging Big Data Analytics. *International Journal of Science and Research*, pages 2073–2076, 2015.

[32] Rebecca Fiebrink, Perry R Cook, and Daniel Trueman. Human Model Evaluation in Interactive Supervised Learning. 2011.

[33] Zahra Ghafoori. (PhD) Robust and Efficient Unsupervised Anomaly Detection in Complex and Dynamic Environments. (January), 2018.

[34] Nicolas Goix. Machine Learning and Extremes for Anomaly Detection-Apprentissage Automatique et Extrêmes pour la Détection d'Anomalies Spécialité "Signal et Images" présentée et soutenue publiquement par. 2016.

[35] Fahrettin Gökgöz. Anomaly Detection using GANs in OpenSky Network. pages 1–7.

[36] Markus Goldstein and Seiichi Uchida. A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data. *PloS one*, 2016.

[37] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. *Advances in neural information processing systems*, 2014.

[38] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved training of wasserstein GANs. *Advances in Neural Information Processing Systems*, 2017-Decem:5768–5778, 2017.

[39] I. Haloui, JJ.S. Gupta, and V. Feuillard. Anomaly detection with Wasserstein GAN. pages 1–36, 2018.

[40] Matthew Herland, Richard A. Bauder, and Taghi M. Khoshgoftaar. The effects of class rarity on the evaluation of supervised healthcare fraud detection models. *Journal of Big Data*, 2019.

[41] Cun Ji, Xiunan Zou, Yupeng Hu, Shijun Liu, Lei Lyu, and Xiangwei Zheng. XG-SF: An XGBoost Classifier Based on Shapelet Features for Time Series Classification. *Procedia Computer Science*, 147:24–28, 2019.

[42] P.J Jones, M.K James, M.J Davies, K. Khunti, M. Catt, T. Yates, A.V. Rowlands, and E.M. Mirkes. FilterK : A new outlier detection method for k-means clustering of physical activity. *Journal of Biomedical Informatics*, 2020.

[43] Hossein Joudaki, Arash Rashidian, Behrouz Minaei-Bidgoli, Mahmood Mahmoodi, Bijan Geraili, Mahdi Nasiri, and Mohammad Arab. Using Data Mining to Detect Health Care Fraud and Abuse: A Review of Literature. *Global Journal of Health Science*, pages 194–202, 2014.

[44] Melih Kirlidog and Cuneyt Asuk. A Fraud Detection Approach with Data Mining in Health Insurance. *Procedia - Social and Behavioral Sciences*, pages 989–994, 2012.

[45] Aleksandar Lazarevic, Levent Ertoz, Vipin Kumar, Aysel Ozgur, and Jaideep Srivastava. A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection. *Army High Perform- ance Computing Research*, pages 25–36, 2013.

[46] T.G. Legotlo and A. Mutezo. Understanding the types of fraud in claims to South African medical schemes. *South African Medical Journal*, page 299, 2018.

[47] Dan Li, Dacheng Chen, Jonathan Goh, and See-kiong Ng. Anomaly Detection with Generative Adversarial Networks for Multivariate Time Series. *Applications on the ACM Knowledge Discovery and Data Mining*, pages 1–10, 2018.

[48] Bo Liu, Ying Wei, Yu Zhang, Qiang Yang, and Hong Kong. Deep Neural Networks for High Dimension, Low Sample Size Data. *International Joint Conference on Artificial Intelligence*, pages 2287–2293, 2017.

[49] Scott M Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions Scott. *Neural Information Processing Systems Conference*, pages 426–430, 2012.

[50] David McDaid, Sherry Merkur, and Anna Maresso. EuroHealth Report. *European Observatory on Health systems and Policies*, pages 1–44, 2011.

[51] Pulane Molefe. CMS News The Council for Medical Schemes'. Technical report, 2018.

[52] Gonçalo Mordido, Haojin Yang, and Christoph Meinel. Dropout-GAN: Learning from a Dynamic Ensemble of Discriminators. 2018.

[53] K. Naidoo and V. Marivate. Unsupervised Anomaly Detection of Healthcare Providers Using Generative Adversarial Networks. *19th IFIP WG 6.11 Conference on e-Business, e-Services, and e-Society*, 12066:419–430, 2020.

[54] Quoc Phong Nguyen, Kar Wai Lim, Dinil Mon Divakaran, Kian Hsiang Low, and Mun Choon Chan. GEE: A Gradient-based Explainable Variational Autoencoder for Network Anomaly Detection. *2019 IEEE Conference on Communications and Network Security, CNS 2019*, pages 91–99, 2019.

[55] Ke Nian, Haofan Zhang, Aditya Tayal, Thomas Coleman, and Yuying Li. Auto insurance fraud detection using unsupervised spectral ranking for anomaly. *The Journal of Finance and Data Science*, pages 58–75, 2016.

[56] A. Nicolaides and F. De Beer. Practitioner Ethics , Medical Schemes and Fraud in the South African Private Healthcare Sector. *Medical Technology SA*, pages 1–11, 2017.

[57] Xuetong Niu, Li Wang, and Xulei Yang. A Comparison Study of Credit Card Fraud Detection: Supervised versus Unsupervised. *Association for the Advancement of Artificial Intelligence*, 2019.

[58] OECD Publishling. *Health at a Glance: Europe 2018: State of Health in the EU cycle*. 2018.

[59] G.A. Ogunbanjo and D. Knapp van Bogaert. Ethics in health care : healthcare fraud. *South African Family Practice*, pages 10–13, 2014.

[60] Dong Yul Oh and Il Dong Yun. Residual error based anomaly detection using auto-encoder in SMD machine sound. *Sensors (Switzerland)*, pages 1–14, 2018.

[61] World Health Organization. Prevention not cure in tackling health-care fraud. *Bulletin of the World Health Organization*, pages 853–92, 2011.

[62] A. Prabakar, R. Rajeswari, and R. Rajaram. Procedia Engineering Network Anomaly Detection by Cascading K-Means. *International Conference on Communication Technology and System Design*, 2012.

[63] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. pages 1–16, 2015.

[64] K. Saito, Y. Ushiku, T. Harada, and K. Saenko. Adversarial dropout regularization. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, pages 1–14, 2018.

[65] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

[66] M. Saunders and P. Lewis. *Doing Research in Business and Management: An Essential Guide to Planning Your Project*, volume 51. Financial Times Prentice Hall, 2013.

[67] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Georg Langs, and Ursula Schmidt-Erfurth. f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks. *Medical Image Analysis*, 54(May):30–44, 2019.

[68] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. *Information Processing in Medical Imaging*, pages 146–147, 2017.

[69] C.R. Sekhar, Minal, and E. Madhu. Mode Choice Analysis Using Random Forrest Decision Trees. *Transportation Research Procedia*, pages 644–652, 2016.

[70] Yuliang Shi, Chenfei Sun, Qingzhong Li, Lizhen Cui, Han Yu, and Chunyan Miao. A Fraud Resilient Medical Insurance Claim System. *Proceedings of the 30th Conference on Artificial Intelligence (AAAI 2016)*, pages 4393–4394, 2016.

[71] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. *34th International Conference on Machine Learning, ICML 2017*, 7:4844–4866, 2017.

[72] Emanuel H Silva and Johannes V Lochter. A study on Anomaly Detection GAN-based methods on image data.

[73] Václav Šmídl, Jan Bím, and Tomáš Pevný. Anomaly scores for generative models. 2019.

[74] Dallas Thornton, Michel Brinkhuis, Chintan Amrit, and Robin Aly. Categorizing and Describing the Types of Fraud in Healthcare. *Procedia Computer Science*, pages 713–720, 2015.

[75] Peter Tino, Lubica Benuskova, and Alessandro Sperduti. *Artificial Neural Network Models*. 1997.

[76] Alexander Tong, Guy Wolf, and Smita Krishnaswamy. A Lipschitz-constrained anomaly discriminator framework. pages 1–15, 2019.

[77] O.S. Topçu, T. Çakmak, and G. Doğan. Data Standardization in Digital Libraries: An ETD Case in Turkey. *Procedia - Social and Behavioral Sciences*, 147:223–228, 2014.

[78] Paul Vincke. Fighting Fraud & Corruption in Healthcare in Europe: a work in progress. Technical report, 2016.

[79] C. Wang, Y.M. Zhang, and C.L. Liu. Anomaly Detection via Minimum Likelihood Generative Adversarial Networks. *Proceedings - International Conference on Pattern Recognition*, 2018-Augus:1121–1126, 2018.

[80] X. Wang, Y. Du, S. Lin, P. Cui, Y. Shen, and Y. Yang. adVAE: A self-adversarial variational autoencoder with Gaussian anomaly prior knowledge for anomaly detection. *Knowledge-Based Systems*, 2019.

[81] Yibo Wang and Wei Xu. Leveraging deep learning with LDA-based text analytics to detect automobile insurance fraud. *Decision Support Systems*, 2018.

[82] Ke Xu, Agnes Soucat, Joseph Kutzin, Callum Brindley, Nathalie Vande Maele, Hapsatou Touré, Maria Aranguren Garcia, Dongxue Li, Hélène Barroy, Gabriela Flores, Tomas Roubal, Chandika Indikadahena, Veneta Cherilova, and Andrew Siroka. Public Spending on Health: A Closer Look at Global Trends. Technical report, 2018.

[83] Yasin Yazıcı, Stefan Winkler, Georgios Piliouras, Chuan Sheng Foo, Kim Hui Yap, and Vijay Chandrasekhar. The unusual effectiveness of averaging in GaN training. *7th International Conference on Learning Representations, ICLR 2019*, pages 1–22, 2019.

[84] Houssam Zenati, Chuan Sheng Foo, Bruno Lecouat, Gaurav Manek, and Vijay Ramaseshan Chandrasekhar. Efficient GAN-Based Anomaly Detection. *ICLR*, pages 1–7, 2018.

[85] Houssam Zenati, Manon Romain, Chuan Sheng Foo, Bruno Lecouat, and Vijay Chandrasekhar. Adversarially Learned Anomaly Detection. *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2018-Novem:727–736, 2018.

[86] Shuangfei Zhai, Yu Cheng, Weining Lu, and Zhongfei Zhang. Deep structured energy based models for anomaly detection. *33rd International Conference on Machine Learning, ICML 2016*, 3:1742–1751, 2016.

[87] Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, Bo Zong, Haifeng Chen, and Nitesh V. Chawla. A Deep Neural Network for Unsupervised Anomaly Detection and Diagnosis in Multivariate Time Series Data. *Association for the Advancement of Artificial Intelligence*, 2018.

[88] Panpan Zheng, Shuhan Yuan, Xintao Wu, Jun Li, and Aidong Lu. One-Class Adversarial Nets for Fraud Detection. 2018.

[89] Xun Zhou, Sicong Cheng, Meng Zhu, Chengkun Guo, Sida Zhou, Peng Xu, Zhenghua Xue, and Weishi Zhang. A state of the art survey of data mining-based fraud detection and credit scoring. *MATEC Web of Conferences*, 189, 2018.

[90] Changsheng Zhu, Christian Uwa Idemudia, and Wenfang Feng. Improved logistic regression model for diabetes prediction by integrating PCA and K-means techniques. *Informatics in Medicine Unlocked*, page 100179, 2019.

[91] Tommaso Zoppi, Andrea Ceccarelli, and Andrea Bondavalli. On algorithms selection for unsupervised anomaly detection. *Proceedings of IEEE Pacific Rim International Symposium on Dependable Computing, PRDC*, pages 279–288, 2019.

Appendix A

Appendix

Table A.1: Private dataset

Variable Name	**Description**
Paymentrequestid	Unique identifier representing the payment record
ClaimID	Unique identifier representing the claim
MedicalInvoiceID	Unique identifier representing the invoice from the healthcare provider
Industry	Represents the sector the claimant is employed in
Crpayamount	Amount paid excluding VAT
Crpayvat	VAT amount
CrPayAmountIncl	Amount paid including VAT
Date	Date the payment made to the healthcare provider
Product	The type of product that is covered by the claim
BenefitGroup	Refers to the medical benefit being paid to healthcare providers
BenefitCodename	Refers to the medical benefit name being paid to healthcare providers
PrimaryICD10Code	The ICD10 code that indicates the type of injury
DRGCode	The group which the ICD10 belongs to
datepaid	Date the payment made to the healthcare provider

Variable Name	Description
StabilisedInd	Indicator to represent if the injured party is recovered from the injury
ServiceDate	Date the healthcare provider attended to the claimant
DateReceived	Date the invoice received from healthcare provider
Segmentation	Grouping of similar DRG into segments based on costs
CareType	Indicates if the type of injury is acute or subsequent injury. Acute is payments within the first 2 years of an injury, Subsequent indicates payments after a 2 year period of injury
SubClass	The specific sub class the employer belongs to. Relates to the type of Industry the claimant is employed in
EventDescription Groupings	Indicates if the claim is an accident or disease claim Represent the group a healthcare provider belong to (i.e Physiotherapist, Hospitals, Radiologist,etc)
LOS	Length of Stay. Number of days the claimant was in hospital
InjurySeverity	The severity of the injury (Mild, Moderate or Severe
MedicalServiceProviderID	Unique identifier representing the healthcare provider
PractitionerTypeID	Identifier representing the practitioner type the healthcare provider belongs to
HospitalGroupID	Identifier representing the hospital group the healthcare provider belongs to
IsSuspiciousInd	Indicates if the Medical Invoice submitted by the healthcare provider has been flagged as an anomaly

Table A.2: Medicare variables

Variable Name	Description
NPI	National Provider Identifier (NPI) for the performing provider on the claim.

ProviderType	Healthcare provider type
HCPCSCode	HCPCS code used to identify the specific medical service furnished by the provider.
HCPCSDrugIndicator	Identifies whether the HCPCS code for the specific service furnished by the provider is a HCPCS list
LineSrvcCount	Number of services provided
BeneUniqueCnt	Number of distinct Medicare beneficiaries receiving the service.
BeneDaySrvcCnt	Number of distinct Medicare beneficiary/per day services.
AverageMedicarePaymentAmt	Average amount that Medicare paid after deductible and coinsurance amounts have been deducted for the line item service
AveragePaymentAmountPerTypeCode	Average amount that Medicare paid after deductible and coinsurance amounts across same healthcare type and medical service
AverageSrvcCountPerCodeType	Average number of services provided across the same healthcare provider and medical service
AverageUniqueCountPerCodeType	Average distinct Medicare beneficiaries receiving the service across the same healthcare provider and medical service
AverageDaySrvcCountPerCodeType	Average distinct Medicare beneficiary/per day services across the same healthcare provider and medical service
AverageSrvcByAverageUnique	Number of distinct Medicare beneficiary/per day services across the same healthcare provider and medical service

IsSuspiciousInd	Suspicious Indicator based on average cost greater than the average of similar practitioner type and similar injury

Appendix B

Acronyms

GANs	Generative Adversarial Networks
WGAN	Wasserstein Generative Adversarial Network
ADGAN	Anomaly Detection Generative Adversarial Network
VAEs	Variational Autoencoders
AnoGAN	Anomaly Generative Adversarial Network
f-AnoGAN	Fast Anomaly Detection Generative Adversarial Network
DCGANs	Deep Convolutional Generative Adversarial Networks
CART	Classification and Regression Trees
DT	Decision Tree
LR	Logistic Regression
XGB	Extreme Gradient Boosting
TanH	Hyperbolic Tangent
ReLu	Rectified Linear Unit
SHAP	SHapley Additive exPlanation
GAN-AD	Generative Adversarial Networks Anomaly Detection
DRG	Diagnosis Related Group
ICD10	CInternational Classification of Diseases and Related Health Problems
ROC	Receiver Operating Characteristics
AUC	Area Under Curve

TP	True Positive
TN	True Negative
TPR	True Positive Rate
TNR	True Negative Rate

Appendix C

Symbols

C.1 Chapter 2: Literature Review

z	Noise variable
G	Generator network
$G(z)$	Maps noise variable to data space
D	Discriminator network
f	Activation function
net	Weighted input signal
ϕ	SHAP value
$f(x)$	Model output
$E[f(z)]$	Base value

C.2 Chapter 3: Methodology

i^{th}	The number of observation
j^{th}	Predictor variable for the i^{th} observation
Y_i	the target variable output of i^{th} observation
β	Coefficient value
Θ	The learned parameter set

l	Loss function that measures the difference between the predictions
$\hat{y}$	Prediction variable
y_i	Target variable
Ω	Is the regularization term
m	Random forest classifier
K	Classification trees
x	Input data set
p_z	Input noise variables
x	Normal data distribution
p_g	Generated sample from p_z
$\mathcal{Z}$	Latent space
D	Discriminator network
G	Generator network
$D(x)$	Represents the probability that x came from x or the generator
$\min_G$	Minimize the generator network
$\max_D$	Maximize the discriminator network
fD	represents activations in the intermediate layer of the discriminator
d	Represents the dropout value
$A(x)$	Represents the anomaly score
λ	Weight to the anomaly score
$\mathcal{L}_G$	Generator loss
$\mathcal{L}_D$	Discriminator loss
f_D	Feature matching used in the generator and discriminator loss functions
ϕ	Threshold used to identify anomaly, set to 90%
α	Learning rate
β_1	Beta rate used in the optimizer

Appendix D

Derived Publications

K. Naidoo and V. Mariate, Unsupervised Anomaly Detection of Healthcare Providers using Generative Adversarial Networks *19th IFIP WG 6.11 Con-ference on e-Business, e-Services, and e-Society*, Vol. 12066, pages 419–430, 2020.

Index