Machine Learning for Money Laundering Risk Detection in Online Gambling

Charitos Charitou

Department of Computer Science
City, University of London

This thesis is submitted for the degree of
Doctor of Philosophy

May 2021

Declaration

I declare that this thesis titled, "Machine Learning for Money Laundering Risk Detection in Online Gambling", which is submitted in fulfillment of the requirements for the Degree of Doctor of Philosophy, represents my own work except where due acknowledgement have been made. I further declared that it has not been previously included in a thesis, dissertation, or report submitted to this University or to any other institution for a degree or other qualifications.

Charitos Charitou

May 2021

Abstract

This thesis addresses the issue of money laundering in online gambling. Over the years, the online gambling industry has evolved into one of the most profitable industries on the internet. While stringent new regulations have required the industry to become more vigilant, methods used to process proceeds from illicit activities have also advanced and have become more sophisticated. This research examines the application of machine learning for the detection of high-risk money laundering cases in online gambling. This work was part of a collaboration with Kindred Group, a major gambling operator.

Money laundering as a fraud detection problem suffers from the binary class imbalance issue in data mining. This research focuses on investigating data and algorithmic level techniques to provide a solution to that issue. An in-depth analysis of supervised learning algorithms is carried out and a supervised learning framework is proposed to improve the detection rate of high-risk money laundering cases relative to the existing rule-based system. Results showed immediate improvement in the identification rate. Furthermore, it examines Generative Adversarial Networks (GANs) to provide a solution to the class imbalance problem by generating new synthetic data to oversample the minority class. Our GAN-based approach outperformed popular oversampling techniques when combined with supervised learning classifiers. Building on our GAN-based architecture, we then introduce a novel generative adversarial framework, based on semi-supervised learning and sparse auto-encoders, for the detection of fraud in online gambling. Experimental results show that the proposed framework outperforms mainstream discriminative techniques without the need of generating synthetic instances. We validated our system by applying it to other domains that suffer from the binary class imbalance problem.

Finally, unsupervised anomaly detection (AD) framework based on encoder-decoder long short-term memory (LSTM-ATT) networks and Gaussian estimation is examined to discover new patterns in customer behaviours that could be related to money laundering risk, something which is not possible with a supervised framework. Our AD system is evaluated with the help of Kindred's compliance team on specific cases. The feedback received from our research partners suggested that the detected anomalies indicated risk of money laundering and that the proposed framework can be included in their existing anti-money laundering (AML) process.

Acknowledgements

First and foremost I would like to thank my supervisor Dr Artur d'Avila Garcez for all his help and guidance, from the early stages until the writing of this thesis. I would also like to thank him for his confidence in me, allowing me the freedom to develop my own ideas and approach the problem from my own point of view.

I also want to say a big thank you to my supervisor Simo Dragicevic, with whom I spent hours discussing my work, providing me with his very helpful perspective. I'm thankful for his involvement in this project, his assistance and constant inspiration.

This work has been funded by the Kindred Group Plc to whom I am grateful. I am grateful to all the employees of Kindred Group that helped me through this journey from the data science team to the compliance team.

I would also like to thank the members of the Machine Learning group at City University, for all their support during my PhD.

Finally, I would like to thank my parents, who have always provided me with love and support. They have always believed in me and they constantly support me through all my life. Thank you for always being there for me.

I would like to dedicate this thesis to my loving parents...

Contents

List of Figures

List of Tables

List of Abbreviations

ADASYN	Adaptive Synthetic Sampling
AML	Anti-Money Laundering
AD	Anomaly Detection
B-SMOTE	Bordeline Synthetic Minority Oversampling Technique
cGAN	Conditional Generative adversarial networks
CNN	Convolutional Neural Network
FIAU	Financial Intelligence Analysis Unit
GMM	Gaussian Mixture Models
HMM	Hidden Markov Models
GAN	Generative Adversarial Network
LR	Logistic Regression
LSTM	Long Short-Term Memory
NB	Naïve Bayesian
MLP	Multi-layer perceptron
RF	Random Forest
SARs	Suspicious Activity Reports

STRs	Suspicious Transaction Reports
SDG	Synthetic Data Generation
SHAP	SHapley Additive exPlanations
SMOTE	Synthetic Minority Oversampling Technique
SSGAN	Semi-Supervised Generative Adversarial Network
SVM	Support Vector Machines
XGBoost	Extreme Gradient Boosting

Chapter 1

Introduction

Section 1.1 of this chapter introduces the money laundering problem in gambling. Subsequently, Section 1.2 discusses the motivation and need for better detection processes in the online gambling industry. The research scope and questions, along with the contributions the thesis makes to the literature, are presented in Sections 1.3, 1.4, 1.5 and 1.6, respectively. Finally, the thesis structure is outlined in Section 1.7, while the papers published as part of research activities related to this study are presented in Section 1.8.

1.1 Overview of Money Laundering in Online Gambling

Money laundering is the world's third largest 'industry', with an estimated US $2 trillion laundered every year [5]. If criminals want to profit from crime and avoid prosecution, they must find a way to cover the origins of their stolen gains. Thus, every crime that involves stolen money ends with money laundering or spending the proceeds of crime. The global war against money laundering is an ongoing problem that has mainly been faced by the gambling industry since the internet has become widely accessible to a vast range of people. Needless to say, land-based casinos may be considered a haven for such activities. Interestingly, however, online gambling is less vulnerable to money laundering than land-based gambling (different levels of monitoring are in placed in an online gambling environment) at venues such as casinos and

racetracks [6].

In the past, the traditional gambling industry, as a cash-oriented business, has offered many opportunities for criminals to engage in money laundering and spend the proceeds of their crimes. Previously, there was little in the way of Know-Your-Customer (KYC) checks in gambling and it was arguably easy to spend and recycle stolen money in physical casinos and other land-based gambling establishments, such as high street bookmakers. Although online gambling has required that all players be registered, KYC checks were traditionally not mandatory for those depositing and gambling with larger amounts of money. The last decade has seen the introduction of new, much more stringent regulations that have required the online industry to become much more vigilant. However, as standards have begun improving, the methods used to process financial gains from illicit activities have also evolved and become more sophisticated.

Different ways of executing money laundering in online gambling exist as mentioned by gambling industry stakeholders in [7]: i) Deposit large amount of money on a betting account and place few small bets for appearance and then empty the whole account, ii) distributing cash into a number of smaller transaction amounts to evade threshold requirements and reduce suspicion, iii) setup dozens of smaller betting accounts with deposit well below threshold likely to attract attention, iv) small size money launderers can buy a lot of pay safe cards and introduce the money into the financial system through a gambling account, v) player to player games have been very popular since players can lose intentionally all their money to other members of their organisation e.g. poker and vi) a drastic change in betting behavior i.e. unexpected high activity, may indicate that account is now being used by someone for money laundering.

The money laundering process usually involves three sequential steps: placement, layering and integration [8]. In this process criminals attempt to hide the origin of money gained from illegal operations so that it appears to have been gained legally. The money cannot be used by criminals until it is 'clean' in order to avoid any connection to their criminal operations that could expose them to financial crime agencies. In the placement stage, criminals use different techniques to place illicit funds within legitimate financial systems. This can be done by channelling money through legal businesses that deal heavily in cash transactions by smuggling

illegal funds to jurisdictions with weak anti–money laundering (AML) controls. Layering – the most complicated step in any laundering scheme – involves separating proceeds from their illegal source via multiple complex financial transactions (i.e. bank-to-bank or wire transfers) to obscure the audit trail and hide the proceeds. Placing several deposits and withdrawals to vary the amount of money in accounts, including currency changes and purchases of high-value items, are monetary movements that make difficult to follow the money. After sufficient time in the layering process, criminals can extract their funds and reintroduce them to the financial system as legitimate money, a stage of the process known as integration. While layering costs may decrease the value of the placed funds, they will likely still be used during integration to make high-value purchases, such as real estate, luxury goods and residential or commercial property. Online gambling can be used as a medium in all three stages of money laundering.

1.2 Research Motivation

Gambling regulators have set high expectations for anti–money laundering practices that all gambling operators are obligated to follow. The European Union's 4th Anti–Money Laundering Directive, which came into effect in June 2017, increased the pressure on the gambling industry to ensure that it is not used as a vehicle for financing terrorism, money laundering or leisure spending of the proceeds of crime (collectively, these activities all fall under the category 'AML'). Until recently, the industry primarily tackled the identification of crime in online gambling with rule-based systems. Rule-based systems rely on encoding rules and expertise based on human experience. Whilst such systems are capable of easily embedding regulatory requirements that focus on simple and static thresholds, they are unable to adapt to new requirements to proactively monitor the activity of millions of online customers and, importantly, to change behavioural patterns related to criminal activity online.

Whilst improvements have been made, the online gambling industry needs to continue evolving and raising standards in compliance monitoring. In the United Kingdom, the Gambling Commission sent a clear message to operators to raise their compliance standards and to place

consumers at the heart of their businesses, warning that regulatory breaches could lead to higher financial penalties and even the possibility of license review and revocation. The Gambling Commission has also stated that compliance with AML and counter-terrorism finance (CFT) starts with a supportive culture at board and senior management levels.

In recent years, the Gambling Commission has begun to take more punitive action against operators for regulatory breaches. For example, 888 Holdings was subject to a record regulatory settlement of £7.8 million for failings in social responsibility practices, with one customer having stolen £55,000 from their employer to fund their gambling habits. The Gambling Commission ordered Ladbrokes Coral Group to pay £2.3 million for failing to intervene after two problem gamblers lost £1.3 million in stolen funds whilst using its online casino. Most recently, William Hill was required to pay a regulatory settlement of £6.2 million for failing to protect consumers and prevent money laundering.

The major motivations for this research are to understand the problems faced by the gambling industry with regard to raising standards in money laundering detection and to identify the key problems that the current systems cannot solve. A qualitative analysis is undertaken via industry stakeholder interviews in [7], ascertained the main issues with the current approaches and explained why there is a need for more sophisticated methods. The major issues centred on the need for greater vigilance and accuracy in ongoing monitoring, moving beyond simple methods that criminals can easily circumvent. We investigate how machine learning can be used and applied to improve the identification rate of customers at high-risk of money laundering in comparison with rule-based detection systems. The project is supported with data provided by Kindred Group, one of the world's largest online gambling groups and operator of a number of major online gambling brands, including UNIBET and 32Red.

The AML process is conceptually similar to fraud detection, an area that has been the focus of a great deal of research in recent years. It has been shown that applying machine learning techniques to detect fraud can solve the problem to a certain degree, with the best results achieved with supervised learning. However, the problem with supervised learning is that it requires labelled data for both non-fraudulent and fraudulent behaviours in order to train a model.

Kindred supported this project by making anonymous data available with labels describing high-risk (fraudulent customers) and no-risk of money laundering (non-fraudulent customers). Notwithstanding, the non-fraudulent customers are much greater in number compared to customers with high money laundering risk.

1.2.1 Business and Regulatory Challenges in Combating Crime in Gambling

This section includes the challenges that operators and regulators face and how emerging technologies can be used to improve the identification of suspicious activities in the online gambling environment. In the study that we completed in [7] criminals were consistently described as sophisticated by gambling stakeholders, in that they have the ability to stay 'under the compliance radar' for long periods of time, making it difficult for compliance departments to track them. Despite this, it was stressed that this certainly does not apply to all cases, and the cases publicised by the Gambling Commission have concerned obvious examples of high spending individuals being missed by operators.

Further, it is relatively easy to develop strategies using multiple online gambling accounts to evade their respective compliance controls and checks. The study [7] argued that the current rule-based systems adopted by most operators for their AML and proceeds of crime monitoring checks are too rigid, as criminals can quickly adapt to known rules and thresholds. Moreover, criminals are often well educated about relevant regulations and can use laws (e.g. data privacy and protection laws) against operators themselves to help cover their tracks. That said, it was also stressed that, whilst the General Data Protection Regulation (GDPR) does rightly afford players privacy, law enforcement agencies and the Gambling Commission can make retention of information and information requests. These requests would effectively lift the veil on criminals' privacy if they were made in the event of detecting and preventing crime and in the public interest.

Moreover, it is critical that the industry does everything possible to keep criminals guessing –

in effect, 'outsmarting' criminals through the development of new methods and systems with which they are unfamiliar. However, to reduce risk and eliminate criminal activity, today's industry needs more targeted and sophisticated strategies that provide new ways to identify suspicious and criminal activity. In parallel, the industry should not decrease its focus on correctly and consistently applying very basic measures within businesses – something the Gambling Commission has repeatedly failed to see in enforcement casework.

1.2.2 Key Technical and Legal Challenges in Combatting Crime in Gambling

The industry has embraced a risk-based approach to ensure that measures to avoid or mitigate money laundering, terrorist financing and spending of the proceeds of crime are proportionate to the risks identified, confirming that resources can be allocated in the most efficient way. Increased monitoring expectations pose significant operational challenges for operators, since compliance costs have increased significantly and increasing coverage using the current systems and tools could easily double the size of compliance teams. A counter argument is that compliance costs have increased due to sustained underfunding in this area for a long time. This has led to poor standards and subsequent regulatory enforcement cases, which has required licensees to invest heavily in order to raise their basic standards to an acceptable level.

Having more effective systems in place to analyse and process these risks is therefore becoming increasingly strategically important. Until now, operators have integrated systems with specific rules and thresholds to monitor their business, often by adapting and evolving existing back office systems that undertake core gambling processes (e.g. registration, player wallets, payments). However, such rule-based systems have disadvantages as set forth below.

- Ongoing maintenance: Adding new knowledge to the system to solve other problems could lead to contradictions with old rules.
- Ineffective: Rule-based systems are not effective at widening the net of analysis; rather they focus on absolutes and often extremes

- Easy to understand: Criminals can very easily remain undetected if they know a system's rules. They can adjust their approach and use different methods to stay unnoticed while the system's rules remain static rather than dynamic.

Although larger operators have improved their procedures and checks, medium-sized operators have found it challenging to find resources to implement 'step changes' in capability, as the majority of spending is focused on competing with larger brands (i.e. increasing customer acquisition and retention costs). This means that more investment is vital for the industry to demonstrate that it is approaching this issue in smarter and better ways.

1.2.3 Data Challenge

A major limitation of the AML process today is that (with some exceptions) crime agencies provide limited feedback to operators. In Malta, for example, the Financial Intelligence Analysis Unit (FIAU) issues an annual report which makes public the analysis of suspicious activity reports (SARs) received by the unit. The National Crime Agency also produces an annual analysis of the SARs submitted to it. Moreover, every operator receives a receipt and score regarding the quality of their submitted reports. Finally, the FIAU informs operators whether the incident has been investigated. This is not the case with intelligence agencies in other jurisdictions, although some of these practices are likely to be increasingly adopted (e.g. in the United Kingdom). This represents an opportunity for authorities and industry to work together to improve the process, primarily to enable operators to share data and learn from previous experiences. However, as no legal gateway currently exists that enables law enforcement bodies or regulators to share information or intelligence about other businesses, the lack of process here is a barrier.

The main reason for crime agencies' limited feedback on cases is that thousands of reports are submitted every day to be reviewed and investigated. Agencies therefore struggle to manage the increasing volume of cases. In addition, agencies must be very cautious in light of the possibility that criminals may have connections at gambling operators that could compromise

their investigations, making data sharing and feedback a sensitive area.

The absence of feedback increases operators' accountability and responsibility. Regulators and financial crime agencies cannot be responsible for advising whether operators should close a customer account due to a SAR being raised. Rather, compliance teams have to make critical decisions about whether to continue accepting money from these customers. In addition, they must make these decisions with the knowledge that commercial teams often have the opposite business objectives. In cases where evidence is sparse, the industry has some very challenging decisions to make, as legitimate customers could be turned away and driven to competitors with less robust monitoring in place, as outlined earlier.

Today, crime agencies are confronted with greater volumes of suspicious transaction reports (STRs) and SARs in light of the growing pressure on operators to do more to flag suspicious behaviours. Accordingly, it would be beneficial for all parties involved if better communications could be established between operators and crime agencies.

Another area flagged as having potential for improvement was the submission of STRs and SARs, which is currently a highly manual and time-consuming process with different formats and standards in different jurisdictions. Developing a single technical submission format (e.g. a consistent API or XML standard) for STRs and SARs would be beneficial and could save costs that could be re-invested in improving detection capabilities.

1.3 Research Questions

Based on the motivations described above, this research investigates whether machine learning techniques can be used to effectively predict fraud in online gambling. Thus, the main research question in this thesis is as follows:

How can machine learning methodologies be used to identify and mitigate fraud risk in online gambling?

This main research question is further divided into the three sub-questions addressed in this

thesis, as follows:

1. What are the main challenges of detecting money laundering in online gambling?
2. How can a model developed with the supervised learning approach be used to effectively analyse gambling fraud?
3. How can a model flag new behaviours which could potentially be related to money laundering and learn new patterns?

Finding answers to these research questions can help address the implementation, validation and evaluation of the proposed approaches. One major concern in building and evaluating the performance of any machine learning model in a realistic condition is the challenge of obtaining ground truth labels for money laundering as explained in Section 1.2.3.

1.4 Research Aim

Research on the detection of money laundering in online gambling is in its early stages, and further investigation on building a real-time AI monitoring system is needed. The research aim is addressed through the following main objectives:

- Understand fraud and money laundering issues in the gambling industry;
- Develop features to represent customers' behaviour;
- Build a supervised framework to predict which customers are at high risk of money laundering;
- Tackle the class imbalance problem in fraud detection through oversampling and semi-supervised learning;
- Develop an unsupervised learning framework to detect abnormal behaviours, as customers with certain anomalies could be more likely to commit fraud.

1.5 Research Methods

The research methodology adopted in this thesis can be divided into three major steps. As shown in Figure 1.1, the first step of this thesis was to examine how to answer the research questions proposed in Section 1.3. Initially, we focused on understanding money laundering in the gambling industry. To achieve this first objective, we interviewed a variety of gambling industry experts and stakeholders. The outcomes of these interviews helped us understand the general viewpoint of the industry and what capabilities exist to tackle this problem. More specifically, we looked at where technology can be used to raise standards. Finally, we summarised the key discussions taken from the interviews – which included experts from national crime agencies, regulators and trade associations – in the form of a white paper [7].

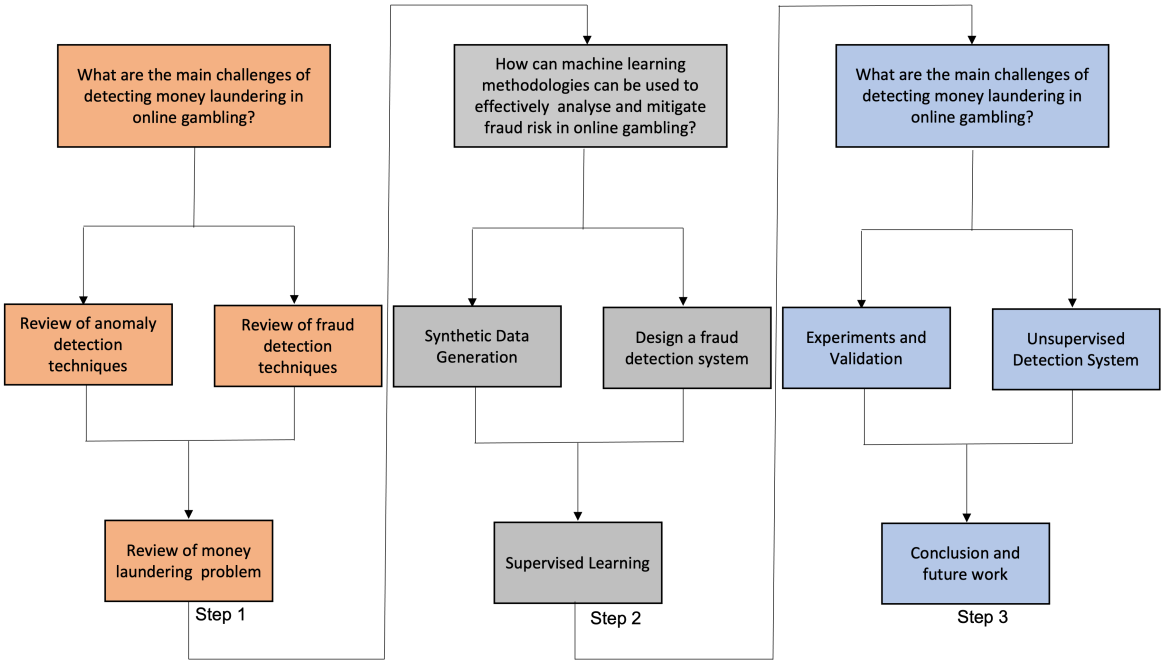


Figure 1.1: Illustration of research methodology

At the same time, we investigated the literature in order to provide a comprehensive overview of topics such as fraud and anomaly detection. We also reviewed the approaches that have been implemented in the existing literature to deal with the challenge of the lack of publicly available money laundering datasets. Further, we studied popular machine learning techniques that have been used for fraud detection. Various applications of supervised machine learning methods that have been applied in fraud detection problems and the common approaches used to handle

imbalanced data problems in supervised learning were examined, as well. Next, we examined different anomaly detection techniques. Finally, we investigated the evaluation techniques for effectively measuring a fraud detection system.

The second step involved building a robust supervised framework with high identification rate. To achieve this objective, the generation of high-quality synthetic data was essential. Good-quality data are a precious commodity and – like most precious commodities – can be difficult to obtain. The lack of quality data related to money laundering is a major problem faced by the gambling industry, since data collection is often difficult, time consuming, expensive or outright impossible. In addition, rich datasets are rarely shared due to privacy constraints. Even when good-quality data are available, many datasets suffer from another inherently common issue: the class imbalance problem. Initially, we explored oversampling techniques together with powerful machine learning models to tackle the imbalanced classification problem. Generative adversarial networks have been widely used to generate images from scratch, but they can also be used to generate sound, speech and text. They have proven to be very useful for semi-supervised, fully supervised and reinforcement learning. Since GANs have proved to be a powerful tool for data generation, we proposed a framework based on GANs for high-quality synthetic data generation. Further, we expanded the capabilities of our GAN framework by proposing a semi-supervised approach for the classification of fraud in online gambling. We tested our methodology against different types of imbalanced datasets.

Next, we focused on the discovery of behaviours that could result in money laundering. Unsupervised learning techniques were investigated to spot anomalies in players' behaviour and a framework based on LSTM and Gaussian estimation was proposed. Subsequently, we evaluated the anomalies detected by our system with the help of the compliance team at Kindred group where a positive feedback was received regarding the usability of our system.

1.6 Contribution to the Knowledge

We provide an in-depth study and extend the current industry practice that revolve around rule-based system to automated machine learning tools for the detection of fraudulent behaviours. Rule-based systems have the disadvantage that sophisticated fraudulent behaviours can be undetected due to fraudulent players deciphering easily the system to stay under the radar. Partnering with industry leaders, precisely with Kindred group we aimed to create a practical foundation to base a new system that would limit both false positives and false negatives. The work discussed in this thesis makes the following contributions to the existing knowledge in this field:

1. Designed new and enhanced existing features abstracting gambling behaviours (Chapter 3)

2. Provided a supervised learning framework for the systematic study of these behaviours and in-depth comparative analysis of popular supervised learning techniques applied to this problem that can be used for future studies (Chapter 4).

3. Using SHapley Additive exPlanations (SHAP), a feature explanatory analysis is provided solving for the problem of explainability, commonly cited as the reason of non-usage of machine leaning in finance (Chapter 4).

4. Identifying that the imbalance dataset problem is fundamental for enhancing machine learning techniques to gambling industry. A method using the generator of GANs is suggested showing promise over standard oversampling methods (Chapter 5).

5. Extending the usage of GANs beyond the data level, incorporating them directly not just the generator to solve the imbalance problem at the algorithmic level. This work proposes a novel system based on semi-supervised GANs to predict fraud in online gambling. By adding another output in the discriminator's architecture, GANs can perform classification. Semi-supervised GANs were able to classify imbalanced data without the need many training examples. This approach was shown to be an effective method for

imbalanced dataset classification, as the results indicate when compared with standard oversampling and machine learning techniques (Chapter 6).

6. Exploring unsupervised learning in solving for unlabelled behaviours. Providing a detailed study using a two step framework for anomaly detection based on the encoder-decoder LSTM model with Attention mechanism. Positive industry feedback (Kindred compliance team) was provided, which can be used as a future industry benchmark (Chapter 7).

1.7 Organisation of Thesis

Chapter 2 provides a literature review on the topics of money laundering and fraud detection and provides basic knowledge regarding the fundamental techniques used in this area. It also discusses the existing approaches that have been used in the literature to deal with the challenge of imbalanced datasets. Further, it discusses common algorithmic solutions that have been applied in the financial industry that are relevant to our problem. We also examine how GANs have been utilised to provide solutions to related problems. Finally, anomaly detection approaches are investigated.

Chapter 3 contains a explanatory analysis of the data provided by Kindred, as well as a discussion of the most important characteristics that could be used to generate new features and presents the new dataset that resulted from the data pre-processing.

Chapter 4 includes a comparative study based on supervised learning techniques. Using the new features generated in Chapter 3, it evaluates the effectiveness of the new data in assisting in the identification of high-risk players.

Chapter 5 illustrates a GAN-based framework for generating new synthetic data to improve the classification of imbalanced datasets. The new proposed method is evaluated on benchmark datasets and against other oversampling techniques as well as on the gambling fraud dataset of Chapter 3.

Chapter 6 presents the novel semi-supervised framework based on semi-supervised GANs for

detecting fraud in online gambling. We evaluate our method against standard machine learning techniques that have been used extensively for imbalanced dataset classification. We show that our system outperforms popular classification techniques even when combined with standard oversampling techniques.

Chapter 7 illustrates the unsupervised approach based on encoder-decoder LSTM models with Attention mechanism to detect trends and patterns that had not been seen before. The anomalies detected for the test cases are then further evaluated by Kindred's compliance team.

Chapter 8 concludes the thesis by summarising the findings and highlighting the main contributions with respect to the thesis objectives. Plans for future work in the field of fraud and anomaly detection are also explored.

1.8 Publications

- Charitou, C.,Garcez, A., Dragicevic, S., "Raising Standards in Compliance Application of artificial intelligence to online gambling data to identify anomalous behaviours.", Jul2018. [Online]. Available:https://www.city.ac.uk/data/assets/pdffile/0014/421106/City-collaborative-Whitepaper-Anti-Money-Laundering-and-Artificial-Intelligence-02July2018.pdf [7].

- Charitou, Charitos, Artur d'Avila Garcez, and Simo Dragicevic. "Semi-supervised GANs for Fraud Detection." 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, 2020. [9].

- Charitou, Charitos, Artur d'Avila Garcez, and Simo Dragicevic., "Synthetic Data Generation for Fraud Detection using Gans," 2021. [Online]. Available: http://arxiv.org/abs/2109.12546 [10].

Chapter 2

Literature Review of Fraud Detection Techniques

In Chapter 1, we defined the problem of money laundering in online gambling and identified the research areas of this thesis. Imbalanced dataset classification and anomaly detection are the two main areas on which our research is focused, due to the imbalanced nature of fraud datasets and the need for anomaly detection to discover new patterns of fraudulent behaviours. Leveraging the generic categorisation of fraud detection, machine learning techniques in the literature are divided into supervised and unsupervised approaches. Figure 2.1 presents a diagram showing different research areas explored in this thesis.

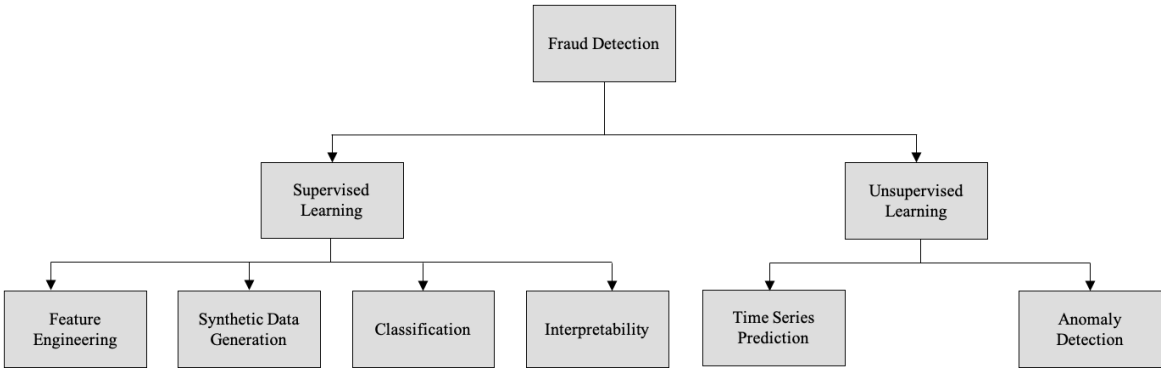


Figure 2.1: Research areas explored in this thesis

This chapter starts in Section 2.1 by exploring how AI has been applied in the gambling industry to either detect fraud or identify problem gamblers. In Section 2.2 and 2.3, we show

how supervised learning techniques have been used to tackle fraud-related problems. Section 2.4 describes how the literature has dealt with the issue of imbalanced classification, explaining the rationale behind the specific techniques selected for this thesis. Section 2.5 discusses relevant work in which machine learning techniques have been used for anomaly detection in the context of fraud detection, with a focus on unsupervised learning. Finally, Section 2.6 elaborates on evaluation techniques, i.e. performance metrics for effectively measuring a fraud detection system.

2.1 Machine Learning in Online Gambling

The evolution of AI and machine learning in the last decade has helped improve our everyday lives and find solutions to some of the most difficult problems. Over the years, machine learning has been introduced into many industries with different applications. The recent exponential growth of online gambling has created opportunities for the industry to better understand its products and customers. Helping those with addiction and flagging illicit activities are some areas on which regulators and the industry have focused.

Braveman and Howard [11] focused on detecting betting patterns that could serve as behavioural markers to predict the development of gambling-related problems, using k-means clustering to identify high-risk players. The characteristics of these were as follows: (i) frequent and (ii) intensive betting, in conjunction with (iii) high variability across wager amount and (iv) increased wager size during the first month of betting.

Based on Braveman and Howard's study, Dragicevic et al. [12] implemented a k-means clustering algorithm to identify groups of gamblers who showed signs of potential problematic behaviours, such as frequency and intensity of betting. Akhter, Syed et al. [13] developed and trained a support vector machine (SVM) system to predict potential online gambling addicts which made predictions for all active users based on their recent usage history.

A comparison study is presented in [14], where the authors used four supervised machine learning techniques – logistic regression (LR), Bayesian networks, neural networks and random

forest (RF) – to predict self-exclusion in online gambling. Self-exclusion is a facility of people who want to stop gambling for six months [14]. Similar to money laundering fraud data, self-exclusion data were heavily imbalanced. The authors explored a synthetic oversampling technique in conjunction with supervised learning techniques to improve their results. RF was deemed the most accurate model for predicting problematic gamblers. Our research initially followed a similar approach, as we examined popular machine learning techniques together with oversampling approaches to predict which players were at high-risk of money laundering (see Chapter 4).

Insights from interviews with industry and public officials [7] indicated the importance of interpretability when using machine learning to make a decision. Building on the research [14], Percy et al. [15] proposed a new variation of the TREPAN algorithm for extracting human-readable logic rules from a neural network. The insights extracted with TREPAN are crucial in explaining potentially harmful gambling behaviour.

Most of the efforts of the gambling industry, as the above research shows, have been focused on responsible gambling and ways to detect and mitigate risk for gambling addiction. However, as the online revolution continues, the online gambling industry needs to become smarter in other areas. Whilst the market of online gambling has grown drastically, the probability of criminals taking advantage of the industry's weaknesses has increased as well. Operators and regulators must utilise the capabilities of machine learning to process vast amounts of data and find better, smarter ways to strengthen their underlying anti-fraud and anti–money laundering processes.

2.2 Fraud Detection

Money laundering is by definition a subsidiary of fraud. Fraud as a wider area of research has received a great deal of attention from academia compared to money laundering. In this section, we provide an overview of the machine learning algorithms that have been applied to discover patterns in data in order to differentiate fraudulent from normal cases. Promising

results have been achieved in the identification of financial and insurance fraud [16, 17, 18]. In the literature, these encouraging results were accomplished using both supervised [19] and unsupervised [20, 21] machine learning algorithms. By definition, a supervised algorithm is a system that learns by example (that is, from labelled data). Here, a predictive model would be trained under supervision with labelled data (labels describing players at high-risk for money laundering versus genuine players). In contrast, unsupervised learning approaches are trained on unlabelled data samples and they are commonly used in outlier or anomaly detection.

2.3 Supervised learning methods for fraud detection

Several groups of researchers have devoted significant effort to studying fraud systems from different perspectives, based on which a portfolio of machine learning techniques has been applied for fraud detection [17]. Usually, supervised methods are implemented as a standard method when the required labels are available. A vast range of algorithms have been used to solve this particular problem, with logistic regression, neural networks and random forest as some of the most common approaches. Supervised learning techniques according to [22] can be categorised into two different groups based on their evaluation approach: (i) supervised profiling and (ii) classification.

Supervised Profiling

Supervised profiling is defined as the approach wherein labels are available to construct distributions or profiles for fraudulent and normal cases [23]. New behaviours are automatically flagged by the system on the basis of similarity to fraudulent behaviour, dissimilarity from normal behaviour or both. In the supervised profiling space, the rules-based profile technique is a popular approach for detecting fraud. In this method, a profile is defined by a set of rules and each behaviour is then matched to each rule. For example, a player who deposits more than £2,000 should be considered at high-risk for money laundering. This set of rules can be defined either by human experience or by rule discovery algorithms [23]. One of the benefits

of rules-based profiling systems is that they are easy to implement and understand. These systems can also be used as part of a more complex framework for filtering behaviours with very low risk of fraud [24].

However, rules-based profiling approaches are not the most effective solution for fraud detection. Rules can be static, and in a dynamic and fast-paced environment like online gambling or the financial industry where criminals are constantly evolving, a great deal of effort is required to keep the rules updated [23]. As a solution, the authors in [25] suggested a weighted ensemble learning approach in which new rules could be added while keeping existing ones. Profiling has also garnered much interest from the telecommunications industry in fraud detection research [26].

Classification

Classification supervised learning methods are used when labelled data are available. Two types of supervised classifiers exist: (a) generative classifiers (e.g. Naïve Bayes (NB), Bayesian networks, hidden Markov models) and b) discriminative classifiers e.g. logistic regression (LR), multi-layer perceptron (MLP), random forest (RF). The main difference between the two types is that a generative model learns the joint probability distribution $p(x, y)$ in order to predict the conditional probability with the help of Bayes' theorem. In contrast, a discriminative model learns the conditional probability distribution $p(y|x)$. In this section, we investigate the application of both groups of supervised classifiers in the fraud detection field.

Algorithms based on decision trees have gained popularity due to their high interpretability, since the decision rules can be easily extracted from the tree [27]. The authors in [28] demonstrated a successful application of decision tree learning for detecting fraudulent activity in energy consumption data. They defined two types of fraud in energy consumption: a) the consumer's smart meter reports less energy consumption than actually consumed and b) the consumer's smart meter reports more energy consumption that actually used due to rogue connections. Their decision tree–based approach managed to profile normal energy consumption behaviour, thus allowing for the detection of potentially fraudulent activity. Another study

showing the strong capabilities of tree-based methods for fraud detection was presented in [2], where Kumar et al. proposed a fraud detection system based on an RF algorithm for predicting fraud in real-world credit card data. The authors' results indicated that their system could identify fraudulent cases with high accuracy. Yao, Zhang and Wang [29] examined supervised learning methods for the detection of fraud in financial statements. The authors developed a hybrid method for choosing the most important features that combined extreme gradient boosting (XGBoost) with various classification algorithms, of which RF ultimately produced the best and most stable results. Similarly, in [30], the authors performed a comparative study of RF, XGBoost and decision tree algorithms to identify the best fit model for classifying credit card fraud. The results showed that XGBoost outperformed the other two techniques. Although tree-based algorithms can be effective and achieve high prediction accuracy, they have also been criticised for poor generalisation and proneness to overfitting [23].

Variations of the basic regression model have been applied in solving different fraud and anomaly detection problems. Dalton S. Rosario [31] examined the efficiency of LR on hyper-spectral data with a proposed model based on model's asymptotic behaviour. Moreover, Min Seok Mok et al. [32] presented a random effects LR model to predict anomaly detection which assisted in identifying not only risk factors for exposure but also the uncertainty not explained by such factors. The authors in [16] applied LR to help identify fraud in auto insurance. The proposed model provided them with probabilities which showed the percentage of risk in each claim.

Another class of algorithm favoured by researchers in the fraud detection community is the artificial neural network (ANN). A neural network can be defined as a series of algorithms that endeavours to recognise underlying relationships in a set of data through a process that mimics how the human brain operates. Khan, Akhtar, and Qureshi in [33] showed that an ANN trained with a simulated annealing algorithm achieved higher identification rates in predicting credit card fraud than standard training procedures. Yu et al. [34] developed a deep neural network with focal loss to detect fraud in credit card data as well. Focal loss was added for training difficult examples. Their method outperformed standard machine learning methods such as LR and SVM. In [35], the authors used a convolutional neural network (CNN) to capture important patterns in fraud behaviour.

Further, Mubarek and Adali [36] examined how well machine learning algorithms worked in intrusion detection. After data pre-processing, the authors compared three machine learning algorithms, which were applied on the NSL-KDD dataset for intrusion detection: Naïve Bayesian network, decision tree, and multi-layer perceptron (MLP). Ultimately,MLP was the most accurate compared to the other two algorithms. However, as they suggest neural networks come with the trade-off of explainability, since they are difficult to interpret and they have been characterised as black box models.

The goal of the support vector machines (SVM) algorithm is to find a hyperplane in an n-dimensional space (where n is the number of features) that distinctly classifies the data points [37]. Many possible hyperplanes could be chosen to achieve separation of the two classes of data points. The objective is to find a plane with the maximum margin, i.e. the maximum distance between data points of both classes. SVM is preferred for solving non-linear problems since it produces significant accuracy while using less computational power and small training sample sizes. These characteristics have made this method attractive in the fraud detection space. Gyamfi et al. [38] used SVM to detect bank fraud and found that their method outperformed a back-propagation network. SVM was also used to identify fraud in credit card transactions in [39, 40] and [41].

Generative supervised models have been used for fraud prevention, similar to discriminative models. Yong et al. [42] proposed a system based on the Naïve Bayes algorithm to detect abnormal fraudulent behaviour among passengers to identify illegal logins by hackers. In [43], the authors implemented a Naïve Bayes classifier to determine whether a text message was spam or from a human. Hidden Markov models (HMM) are another generative method commonly used in the fraud detection field. In [44] and [45], the authors proposed frameworks based on HMM for the prevention of financial fraud in credit cards. In both studies, the results showed that the HMM method could appropriately detect fraud.

In this thesis, we investigate six supervised classifiers, commonly used for fraud detection in the literature: LR, RF, XGBoost, SVM, MLP and NB. LR is a generalised linear model. It is easy to use and is one of the most commonly used techniques for data mining in practice but

is also vulnerable to overconfidence [46]. RF and XGBoost classifiers can capture relationships in non-linear data and are also easier to understand but are prone to overfitting. SVMs have a regularisation parameter that is used to prevent overfitting [23]. MLP can have high prediction accuracy but are very complex and computationally expensive. Finally, NB, in contrast to the discriminative methods, is able to understand the underlying distribution of a dataset in order to make a decision. In Table 2.1 a summary of these methods is provided.

2.4 Class Imbalance Problem

Most supervised learning algorithms are not designed to cope with a large difference in the number of cases belonging to different classes [52]. This problem is known in the literature as class imbalance and is an issue regularly encountered by researchers. The problem corresponds to the issue faced by inductive learning systems when dealing with domains where one class is represented by a large number of samples while the other class is represented by fewer samples. In such cases, the reliability and validity of the results are questionable since prediction algorithms tend to have a bias towards the majority class. In the data provided by Kindred for this research, AML labels (i.e. labels indicating high-risk for money laundering) only constitute approximately 3% of the total number of observations. Such an imbalanced dataset could lead to unintended model performance – for example, classifying all customers as normal and managing to achieve almost perfect accuracy. This, however, is not helpful in real-world situations. Therefore, the problem arises of how to improve the identification of the minority class as opposed to achieving higher overall accuracy. The class imbalance problem has been the subject of extensive research [53, 54, 55]. Apart from the problem of misclassifying the minority class, training on imbalanced data could result in considering minority examples as outliers [56].

Although in most cases the imbalanced class issue will result in misclassification of the minority class, there are cases in which the minority class can be identified accurately [57]. Therefore, other factors could affect the performance of classification algorithms, such as the sparsity of minority data or overlap between majority and minority data [58]. The results in [58] indicate

Table 2.1: Summary of the supervised learning techniques

Methods	Summary	References
Logistic Regression	It is a statistical method for analysing a data set in which there are one or more independence variables that determine an outcome	[31, 32, 16]
Random Forest	It is an ensemble learning method for classification using the decision trees. Each tree tries to predict whether a player belongs to the normal or suspicious group. The classification to a class a class is the result of the majority vote	[47, 48, 49]
XGBoost	It tries to predict a target variable by combining the estimates of a set of simpler, weaker models. Similarly, to RF is based on trees, however uses bootstrapping method for training and each of the weaker models are an improvement of the previous model.	[30, 23, 50, 29]
Multi-layer Perceptron	It's feed-forward neural network, organised in layers and fully inter-connected nodes. Each node contains an activation function. The output will classify the customers into two categories.	[36, 51]
Naïve Bayesian	Based on the Bayes theorem, describes the probability of an event, based on prior knowledge of conditions that might be related to the event.	[42, 43]
Support Vector Machines	It find a hyperplane in an N-dimensional space (Number of features) that distinctly classifies the data points.	[37, 39, 40, 41].

that both class decomposition and class overlapping can affect performance when learning from imbalanced data. Napierala et al. [59] showed that a very important step in the classification of the minority class is to analyse the characteristics of local examples. The authors created four different categories for the minority class samples: safe, borderline, rare and outliers. The last three categories are viewed as unsafe.

Many techniques for handling imbalanced data have emerged in the literature [60, 61, 62, 63]. Solutions have been implemented at the algorithmic, data and hybrid level. At the algorithmic level, algorithms are adjusted in order to reduce bias towards the majority class and improve classification. At the data level, sampling techniques are applied for synthetic data generation to balance the dataset. Finally, hybrid-level approaches combine data-level and algorithmic-level techniques. In Sections 2.4.1 and 2.4.2, we present the relevant literature. The class imbalance issue is observed in binary and multi-class classification problems. Since fraud detection is a binary classification problem, this thesis focuses on binary classification.

2.4.1 Data-Level Techniques

Data-level methods are described as the sampling techniques used to balance a dataset [56]. This means that the number of instances of each class is adjusted either by increasing the instances of the minority class or by decreasing the instances of the majority class. In general, applying sampling algorithms will result in the alteration of the distribution of an imbalanced dataset until it is balanced. Various studies have shown that a balanced dataset can improve the performance of a classifier [57, 64]. Oversampling, undersampling and hybrid methods have been applied to achieve a balanced dataset.

Random undersampling balances a dataset by randomly eliminating majority class examples [65]. While this strategy can reduce bias towards the majority class, it can also discard useful information, which could lead to inaccurate classification performance [66]. When using this approach, it can be assumed that many samples in the majority class are redundant. Therefore, after removing some at random, the final distribution should not deviate much from the original. Pozzolo, Caelen and Bontempi [67] suggested that undersampling can be effective only in specific

conditions. The impact of undersampling depends on the number of samples, the variance of the classifier, the degree of imbalance and the value of the posterior probability.

In contrast to undersampling, oversampling replicates existing instances or generates new synthetic ones. In general, oversampling has shown to produce better results than undersampling [68]. Synthetic oversampling (i.e. generating new synthetic instances) and random oversampling (ROS) are the two methods of oversampling. In ROS, minority samples are added to the training set by randomly replicating minority class samples. Although the performance of a prediction algorithm can be improved with ROS [69], Chawla [66] has suggested that it could also cause overfitting – since the same data may be used more than once – and could be more computationally expensive.

Notwithstanding the problems originating from ROS, advancements in the field of imbalanced classification show that most issues can be overcome with synthetic oversampling. Synthetic oversampling methods generate new synthetic instances in order to balance a dataset. Examples of synthetic oversampling techniques include but are not limited to ADAptive SYNthetic sampling (ADASYN) [62] and Synthetic Minority Oversampling TEchnique (SMOTE) [61].

In SMOTE, the minority class is oversampled by taking each minority class sample and introducing a synthetic example along the line segments joining all of the k-nearest neighbour members of the minority class. Depending on the amount of oversampling required, points from the k-nearest neighbours are randomly chosen. This approach solves the problem of overfitting since synthetic data are generated and not replicated [61]. Synthetic samples are generated by taking the difference between the feature vector (sample) under consideration and its nearest neighbour. The difference is then multiplied by a random number between 0 and 1, which is added to the feature vector under consideration. While SMOTE alleviates the overfitting caused by ROS, as it generates synthetic examples rather than replicating instances, it does not consider that neighbouring examples can be from other classes. This can increase the overlapping of classes and introduce additional noise. A popular extension to SMOTE includes selecting instances of the minority class that are misclassified, such as with a k-nearest neighbour classification model. This modified SMOTE method is called Bordeline-SMOTE

(B-SMOTE) [63].

The key difference between ADASYN and SMOTE is that ADASYN uses a density distribution criterion to automatically decide how many samples need to be generated for each minority data point. First, it improves learning by reducing the bias caused by the imbalance in class priors. Second, it improves performance because the classification decision boundary is adaptively shifted toward 'difficult examples'[62].

Finally, hybrid methods incorporate both oversampling and undersampling techniques. Ganguly and Sadaoui [70] utilised a hybrid method of data oversampling and undersampling to improve effectiveness in addressing the issue of highly imbalanced auction fraud datasets. Their results showed a significant classification improvement for various well-known classifiers. Other popular hybrid methods in the literature include SMOTE+TOMEK [71] and SMOTE+ENN [72]. SMOTE+TOMEK aims to clean overlapping data points for each of the classes distributed in sample space, while SMOTE+ ENN deletes any instance of the majority class which its nearest neighbours are misclassified.

2.4.2 Algorithmic Techniques

Algorithmic-level methods can be divided into cost-sensitive methods and hybrid or ensemble methods [73]. Cost-sensitive methods are solutions at the algorithmic level which aim to improve the task of imbalanced classification by considering misclassification costs during the training stage of a classification algorithm. Including classification costs could be significant when working with sensitive data such as medical datasets [74] (e.g. the classification cost of misclassifying a cancer patient). By assigning different costs, such models have been found to produce good results [75]. Cao et al. [76] presented a cost-sensitive neural network intended to improve classification performance by simultaneously optimising for the best pair of features, structure parameters and misclassification costs. The authors in [77] introduced a cost-sensitive SVM for imbalanced classification.

A hybrid method for tackling class imbalance may incorporate multiple techniques to address

the problem or may use a combination of algorithms in a specific stage of the general solution. Among the hybrid methods in the literature, many are focused on SVMs, tree-based methods or neural networks [78]. Wu, Shen and Zhang [79] developed a fuzzy multi-class SVM algorithm for imbalanced data. Shukla and Bhownick [80] used a k-means clustering algorithm to balance an imbalanced dataset, followed by an SVM to perform the classification task. In [81], the authors proposed different techniques to enhance the classification performance of LR based on cost-sensitive learning to deal with imbalanced datasets.

Ensemble techniques such as bagging, AdaBoost and RF have been used to address imbalanced classification [82]. In [83], the empirical results showed that ensemble algorithms were valuable because they could lead to better performance in comparison to sampling techniques. In that study, the author further noted that the RUSBoost and UnderBagging methods outperformed more complex techniques. Chen et al. [84] introduced two variations of RF for imbalanced classification, i.e. balanced and weighted RF.

2.4.3 Generative Methods for Synthetic Data

Apart from the classical oversampling approaches reviewed in the previous section, the evolution of generative adversarial networks (GANs) has begun to shift attention towards generative techniques for the generation of synthetic data [85]. Our research tests the possibility of applying deep generative models for the generation of new samples, with a focus on GANs, since they have achieved prominent success in image data generation. Together with variational autoencoders, they are the one of the most popular models for learning complicated distributions and have already shown positive results in generating many kinds of complex data [86].

Typically, generative models attempt to learn the underlying data distributions of the original dataset [87]. At the same time, they capture the joint probability of the input data and labels $P(x, y)$, which can be used to generate new data samples similar to existing ones. For example, considering images as input data, each sample (image) has thousands of dimensions (pixels). The generative model's job is to capture the dependencies between pixels (e.g. pixels close to each other may form a recognisable object) [88]. However, this is not sufficient to generate

more samples similar but not identical to those already in a database, which is the purpose of imbalanced learning. Mathematically, we should achieve distribution P, which is as close as possible to the original data distribution P_{ori} and from which we can obtain new samples.

Generative Adversarial Networks

GANs are one of the most popular and successful generative technique for synthetic data generation [3], especially image generation [89][90]. In Figure 2.2 we provide an example which shows the ability of GAN to generate synthetic faces of famous people [1].

Figure 2.2: Synthetic faces generated by a GAN trained on human pictures [1]

Literature on using GANs for oversampling structured data has also emerged. Douzas and Bacao [91] used a conditional GAN (cGAN) to approximate the true data distribution and generate data for the minority classes of various imbalanced datasets. They compared their results against standard oversampling approaches and showed improvements in the quality of data generation.

Lei et al. [92] designed CTGAN, a cGAN-based method to balance tabular datasets with both continuous and discrete columns. They designed a benchmark with seven simulated and eight real datasets and several Bayesian network baselines. CTGAN outperformed Bayesian methods on most of the real datasets while other deep learning methods did not. The authors in [93] proposed oversampling by training a GAN with vanilla GAN loss on only minority class observations. They compared their method against SMOTE and no oversampling and

reported mixed results. Experiments showed that a classifier trained on the augmented dataset outperformed the same classifier trained on the original data.

In this work, GANs were used extensively for tackling the imbalanced class issue. Initially in Chapter 5, we used GANs to generate high quality synthetic data and balanced our training dataset. Then in Chapter 6, we extend the capabilities of a GAN framework to be able to perform classification on imbalanced data.

2.5 Anomaly Detection for fraud identification

The challenge in fighting fraud is that fraudsters are intelligent, learn from mistakes and continuously develop new types of fraud. Hence, techniques are needed that can robustly capture known fraud patterns as well as new types of fraud. So far, we have discussed how supervised machine learning can be used to detect known fraud patterns. In order to detect new fraud patterns and types, however, we need to leverage unsupervised machine learning, e.g. anomaly detection. In this section, we review traditional anomaly detection techniques and how anomaly detection has been used for fraud identification.

2.5.1 Categories of Anomalies

Understanding the types of outliers that an anomaly detection system can identify is essential for obtaining the greatest value from generated insights. Generally speaking, anomalies fall into three main categories [94]:

- Point anomalies: An outlier is defined as a value which is significantly different from the expected value of the time series at that time.
- Contextual outliers: This type of anomaly has values that significantly deviate from other data points in the same context. An anomaly in the context of one dataset may not be an anomaly in another.

- Collective anomalies: This term refers to a group of data points that could be characterised as outliers for the whole dataset.

In this thesis, when we refer to an anomaly, we are referencing a contextual outlier. The raw gambling data used in this project included time series of transaction data and time series of betting data. Due to the nature of our problem, our research focuses on time series anomaly detection and unsupervised machine learning.

2.5.2 Anomaly Detection Techniques

In this section, we explore the methods that have been established for fraud and anomaly detection, from clustering to neural networks. First, we examine an anomaly detection survey published by Chandola et al. [94] which focused on traditional machine learning methods for anomaly detection. The authors presented clustering techniques, SVMs, Bayesian networks and neural networks as the techniques with the most success in fraud detection. Even though certain techniques in [94] had some success, the challenge associated with detecting fraud is that it requires real time detection and prevention. Fraud detection refers to the detection of illegal actions across various industries including banking, telecommunications, insurance and healthcare. Prevention is a complex task, since criminals can adapt.

Another interesting survey of anomaly detection was presented in [95], where the authors reviewed a broad spectrum of deep learning algorithms and demonstrated their applications in different areas of anomaly detection. The authors described fraud detection as one of the main areas in which anomaly detection has been used. As the survey showed, unsupervised sequential deep learning techniques, such as recurrent neural networks (RNNs) and CNNs, have been at the centre of this research field. A typical approach involves monitoring a user or system profile and flagging any deviations. However, one challenge of this approach is that it is not very scalable, as it is difficult to implement when monitoring millions of users.

Traditional Anomaly Detection methods

Traditional machine learning techniques have been used in anomaly detection for fraud identification (i.e. clustering-based, Bayesian networks, SVMs). Efrem et al. [96] used k-means clustering, local outlier factor (LOF) and one-class SVM (OC-SVM) algorithms to find anomalies in data related to drug use in hospitals, with the aim of finding anomalies in time series data in an unsupervised way. A k-means clustering algorithm finds outliers by grouping data into clusters, then comparing each cluster's centre point with all its instances. An instance with a distance above a specific threshold can be considered an anomaly. OC-SVM attempts to find the best hyperplane to separate the data, based on which data can be defined as normal or anomalous using a pre-defined threshold [97]. LOF is used to find outliers in data by comparing the distance of the density of each data [98]. The lowest density is considered an anomaly. All three algorithms showed that they were capable of identifying deviations from normal data, with OC-SVM outperforming the other two methods.

Pu et al. [99] developed an unsupervised hybrid anomaly detection method which combined sub-space clustering (SSC) and OC-SVM to detect attacks without any prior knowledge. The proposed approach was evaluated using the well-known NSL-KDD dataset. The experimental results demonstrated that their method performed better than some existing techniques, namely k-means and DBSCAN.

Monamo et al. [100] examined the use of trimmed k-means (a variation of traditional k-means) – a method capable of simultaneous clustering of objects and fraud detection in a multivariate setup – to detect fraudulent activity in Bitcoin transactions. The number of known anomalies detected successfully was improved compared to k-means. In [101], the authors proposed a Bayesian network by establishing the topology and determining the value of nodes and parameters. They used the probabilistic inferences of Bayesian networks to analyse fraud risk. Gaussian mixture models (GMMs) are another type of probabilistic model that has been applied effectively for anomaly detection. Yusoff et al. [102] proposed a new GMM-based detection algorithm for identifying fraud in the telecommunications industry. The model outputs a risk probability indicating whether an instance is fraudulent or normal.

Deep Learning for Anomaly Detection

A variate of deep learning anomaly detection techniques have been applied in fraud detection. Schreyer et al. [103] showed that the reconstruction error from a deep auto-encoder regularised by the entry's individual attribute probabilities could be interpreted as a highly adaptive anomaly assessment. Their method was evaluated on two real-world datasets and produced better F1 scores compared to the benchmark methods of OC-SVM and principal component analysis (PCA). Wedge et al. [104] also applied deep auto-encoder networks for anomaly detection in credit card transactions by generating new features before using a classification algorithm for prediction. Their method dramatically reduced false positives. A similar approach was followed in [105], where the authors extracted features from an auto-encoder and then fed them to the one-class neural network to detect fraud in credit card transactions.

The use of deep convolutional networks (DCNs) for the identification of fraud in mobile communication networks was examined in [106], where DCNs outperformed traditional machine learning techniques. Zhang et al. [107] introduced a model based on convolutional networks for anomaly detection in online transactions. Their method constructed an input feature sequencing layer that reorganised raw transaction features to form different convolutional patterns and outperformed traditional CNNs. Liu et al. [108] combined temporal convolutional networks to extract features of a time series GMM using Bayesian inference to identify anomalies.

Sequential Models for fraud detection

Although these approaches have shown effectiveness in different applications, they may be unable to work well on multivariate time series data, since they cannot appropriately capture temporal dependencies. To address this problem, temporal statistical prediction methods have been used to model temporal dependency and perform anomaly detection, namely autoregressive moving average (ARMA), autoregressive integrated moving average (ARIMA) [109][110] and their variants. However, these models are sensitive to noise and thus may increase false positives when noise is severe.

HMMs are another popular sequential method for modelling spatio-temporal data and identifying fraud. In [111], the authors used subtractive clustering in combination with HMMs for anomaly detection. First, subtractive clustering is used to build normal anomaly patterns over the dataset. Then the HMM correlates the observation sequence and state transitions to predict the most probable intrusion sequence. The authors in [112] proposed a multiple HMM approach wherein each HMM was trained using a different size of hidden states. HMM responses were ultimately combined in the receiver operating characteristics (ROC) space according to the maximum realisable ROC (MRROC) technique. In [113], the authors introduced a new anomaly detection methodology for data with a latent dependency structure and derived a HMM that extended the regular OC-SVM. Abhinav et al. [114] implemented a discrete HMM credit card fraud detection system, training an HMM on normal cardholder behaviour. If an incoming transaction did not achieve high probability, it was considered fraudulent.

RNNs are a type of deep neural network that has been extensively investigated in the literature on time series prediction and anomaly detection. However, RNNs have difficulty handling long-term dependencies, as explored in [115]. To solve this issue, different variations of RNNs have been proposed, with LSTM networks achieving the best results.

Heryadi et al. [116] investigated deep learning models for learning short-term and long-term patterns from imbalanced input datasets. Their research examined the effect of the non-fraud to fraud sample ratio from 1 to 4 using three deep learning models: CNN, stacked LSTM (SLSTM) and hybrid CNN–LSTM. Their results suggested that CNN achieved the best performance, followed by CNN–LSTM and SLSTM. In [117], the authors evaluated RNNs to detect fraudulent acts on the internet. The outcome of the study supported that RNNs can be highly effective in identifying fraudulent behaviour. Pankaj et al. [118] examined SLSTM networks for time series fraud detection.

Wang et al. [119] also used RNNs to tackle and prevent fraud with anomaly detection. The authors introduced CLUE, a system for real-time anomaly detection. Their neural network output a risk score associated with the possibility of fraudulent activity and managed to correctly flag cases with high identification accuracy. Lp et al. [120] introduced a sandwich-structured

sequence learning architecture which combined RF and a gated recurrent unit (GRU) neural network to detect fraud in transaction data.

LSTMs were the focus of [121] and [122], in which the authors built an LSTM encoder and attempted to reconstruct the normal behaviours of a time series. After calculating the reconstruction error, they then used it to detect anomalies in the signal. The results in both studies showed that LSTM was a viable method for detecting anomalies in time series data. Although LSTMs can capture long-term dependencies, they are sometimes unable to select the relevant driving series to make predictions.

In this thesis, the goal of the anomaly detection task is to find unusual occurrences and patterns in a player's gambling behaviour and relate this information to potential fraud. The anomaly detection step could be added as an additional step for the detection of unknown patterns. Since LSTMs have elicited the best results in anomaly detection on fraud-related problems, we built our system based on LSTMs. In Chapter 7, we present a two-step anomaly detection framework based on LSTM encoder–decoder architecture, with Attention.

2.6 Classification performance measures

The success of computational intelligence algorithms is an important step in determining their suitability for solving particular problems. This is especially true for fraud-related problems like money laundering, where minor improvements in performance can lead to capturing more criminals. Performance metrics in classification are fundamental in assessing the quality of learning methods and models. However, many different measures have been defined in the literature with the aim of facilitating better choices in general or for specific application areas [123]. Choices made based on one metric may be different from choices made based on other metrics. Various standards can be applied to evaluate the performance of an algorithm, such as absolute ability, visual medium and probability of success [124].

This research was based on binary classification (lower-risk and high-risk classes). In binary classification, there are four possible outcomes, as shown in Table 2.2.

Table 2.2: Confusion Matrix

	Positive Class	Negative Class
Positive Class	True positives (TP): Number of examples correctly predicted as pertaining to the positive class.	False positives (FP): Number of examples predicted as positive, which are from the negative class.
Negative Class	False negatives (FN): Number of examples predicted as negative, whose true class is positive.	True negatives (TN): Number of examples correctly predicted as belonging to the negative class.

In the money laundering risk detection problem at hand, FP are individuals belonging to the Normal group who are incorrectly classified by the system as fraudulent (positive). Similarly, FN are those cases where the system should have detected as fraudulent but they remain undetected. The most common performance measure is accuracy, defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

where accuracy measures how many times the algorithm guesses the correct classification. Although widely used, classification accuracy is almost universally inappropriate for imbalanced classification, since high accuracy (or low error) can be achieved by a no-skill model that predicts only the majority class. Therefore, different performance indicators are needed. Consequently, we introduce recall, specificity, precision and F_1 score.

$$Recall/Sensitivity = \frac{TP}{TP + FN} \tag{2.2}$$

$$Precision = \frac{TP}{TP + FP} \tag{2.3}$$

$$Specificity = \frac{TN}{TN + FP} \tag{2.4}$$

$$F_1 = 2 * \frac{Recall * Precision}{Pecall + Precision} \tag{2.5}$$

Recall, also known as TP rate, is the number of correctly classified suspicious players divided by the total number of suspicious players in the test set. $Specificity$, also known as the TN rate,

measures the proportion of normal customers whom the system classified correctly. *Precision* is the number of correctly classified suspicious players divided by the total number of players classified as suspicious by the model. Finally, F_1 score represents the balance between FP and FN. Models with similar accuracy may exhibit different behaviours – for example, low recall and high precision or vice versa. This should be taken into account during model selection: not all binary classifications are equal, and it is important to think deeply about the consequences of all types of misclassification.

2.7 Summary

The literature review in this chapter highlighted the areas where our research was focused on. Initially, supervised learning techniques were explored to provide solution in the fraud detection problem in online gambling. The review also underlined the important issue of class imbalance that exists in fraud detection field which can affect the classification performance of supervised learning algorithms. Further, in-depth solutions to the class imbalance problem were discussed with particular focus on the use of synthetic data generation techniques e.g. using GANs for synthetic data generation. Finally, an overview of anomaly detection techniques was provided since the discovery of new patterns is not possible with supervised learning methods. Sequential models such as LSTM found to be the most effective when we are dealing with anomaly detection on temporal data.

Chapter 3

Data Analysis

3.1 Introduction

In this chapter, we present the analysis of our gambling data together with some descriptive statistics. In Section 3.1, we explain the current AML framework that is used by our partners and identify the areas that this research was focused on improving. In Section 3.2, we show the explanatory analysis of the provided data. In Section 3.3, we discuss the new behavioural features that have been created to profile an online gambling player and used to form the new gambling fraud dataset, which will be our main experimental dataset.

3.1.1 Anti-Money Laundering Process

AML practices involve both KYC screening and online behaviour monitoring. Gambling operators must balance conducting a thorough KYC check as efficiently as possible with minimising hassle for customers so as not to jeopardise the customer experience and revenue opportunities. An AML regulatory framework must adopt measures, policies, controls and procedures commensurate to the risks of money laundering and funding of terrorism.

The current AML system in place at Kindred is composed of different monitoring levels. As can be observed from the workflow of the current process depicted in Figure 3.1, an automated rule-

based system and Kindred employees are responsible for monitoring the activity at both the transactional and gaming level. In the first phase of the AML process, the AML team, together with the responsible gambling (RG) team and other employees, flags any suspicious activity. At the same time, the rule-based system automatically flags players who break the business rules and are deemed at high-risk for money laundering. The flags raised by all parties are then transferred to the AML team for review. Then, the AML team is responsible for deciding whether an internal risk report (IRR) should be raised for the examined case. If an IRR is raised, the de-risking process begins for the specific customer with two potential outcomes: a) The customer could fail the de-risking process, which means the AML team must submit a suspicious transaction report (STR) to the financial crime agency of the relevant jurisdiction or b) the customer passes the de-risking process and is able to continue betting.

One contribution of this research is the improvement of the identification rate of high-risk cases by improving the quality of cases sent to the compliance team for investigation (i.e. reducing false positives and negatives). Since the number of online customers has increased drastically

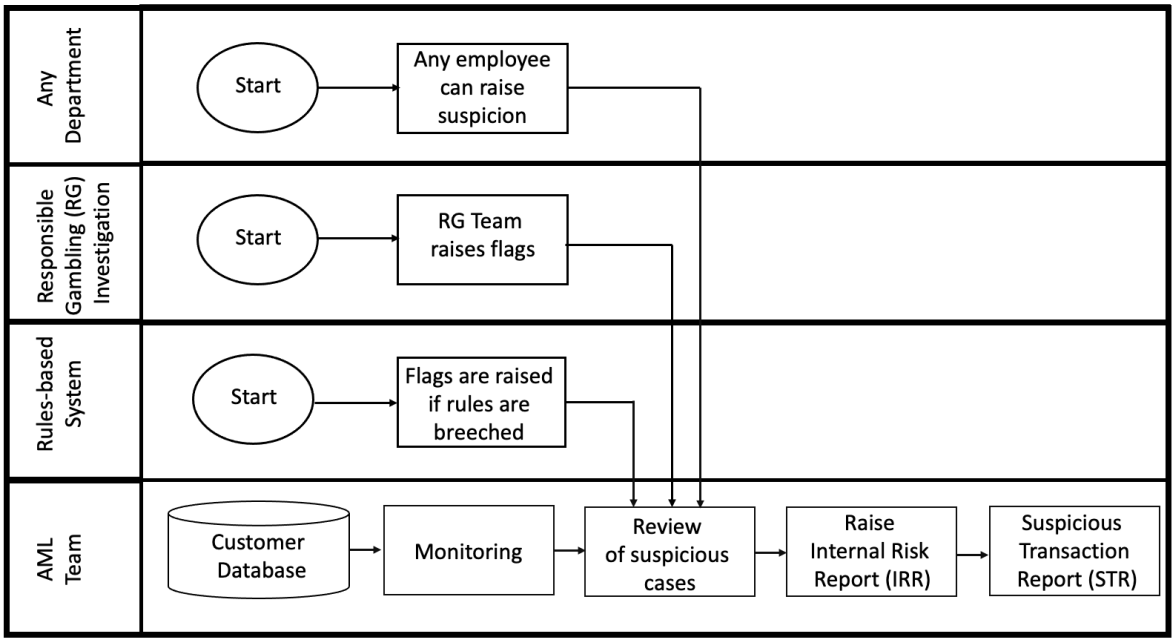


Figure 3.1: AML process monitoring workflow of Kindred Group. All employees can raise an AML flag when they noticed suspicious patterns. The AML team is responsible to evaluate suspicious cases and decide whether an internal risk report should be raised. Then a de-risking evaluation process starts which if the customer fails a SARs report is submitted by the AML team.

over the last few years, the potential threats have increased as well. Therefore, in order to handle the amount of new customers, operators must either hire more staff to ensure better monitoring and investigation of high-risk cases (an expensive solution) or put in place smarter, more efficient systems. For this research, the company provided us with both IRR labels and the automated system detection flags.

Assumption: *The IRR labels are assumed to be the ground truth of high-risk AML cases (true positives). Our improvement target is the flags from the automated system – that is, improvement of true positive flags (defined as the flags that match IRR cases). In addition, we aimed to reduce false negatives – players who remained undetected by the rule-based system but for whom a flag was manually raised by an employee (thus reducing the risk of missing a high-risk case). Also, reduce false positives - players who have been classified as high-risk by the rule-based system, however at the end AML team decided that were not at high money laundering risk.*

3.2 Analysis of Gambling Data

The data available to the gambling operators are typically used for analysis in marketing, risk and compliance activities. These data can be broadly and conceptually classified as personal, machine generated and social network data. Personal data could include information related to personal identification. Machine generated data includes web logs, click streams, sessions records, system monitoring records. Finally, social network data incorporates friends related information, recommendations or likes. Nevertheless, due to privacy restrictions many of the data are difficult to be accessed and utilised e.g social network data. The data of this research are classified as machine generated data.

This section sets out the gambling data used in this study, how it is transformed into behavioural variables for inclusion in the supervised learning models and how over-identifying variables are treated. The anonymised gambling data were collected from customers registered on three Kindred customer facing gambling sites; Unibet, Maria Casino, and Bingo.com.

When constructing a fraud detection model, it is very important to use those features that allow accurate classification. Typical models only use raw transactional features, such as time or amount of transaction.

The datasets incorporate information about transaction details of the customers which includes the type (deposit or withdrawal), the amount, the date, the payment method and the balance before and after the transaction. Moreover, the dataset includes customers' betting information such as number of bets, total bet amount, timestamp and result of bet. Further, the data show that customers are highly active in both betting sports-book and casino games. On one hand, sports-book can include popular in-play bets (where the result is known during the sports event duration), whilst casino can include both slots and table games (such as roulette).

We split online players into two groups: (a) the high-risk/AML group (suspicious), representing players with IRR flags; and (b) the Normal group / Normal players, representing customers with no indication of money laundering risk. In this thesis, we refer to the high-risk group as the AML players or AML group. The four datasets used in our study are listed as follows (see Table 3.1): a) players dataset, b) detection dataset, c) transaction dataset, d) gaming dataset. In the following sections, an overview of these four core datasets is presented.

Table 3.1: Datasets Description

Datasets	Description
Players Dataset	Includes information on players registration date. The gaming platform the use, the currency and finally if they have been flag for AML or not.
Detection Dataset	Includes information if the players have been flagged with flag from the automated rules based system.
Transaction Dataset	Includes information about the transaction details of players e.g. date of the transaction, amount of transaction, type of the transaction.
Gaming Dataset	Includes information about the gaming/betting details of players e.g. date of betting session, amount, number of bets, product.

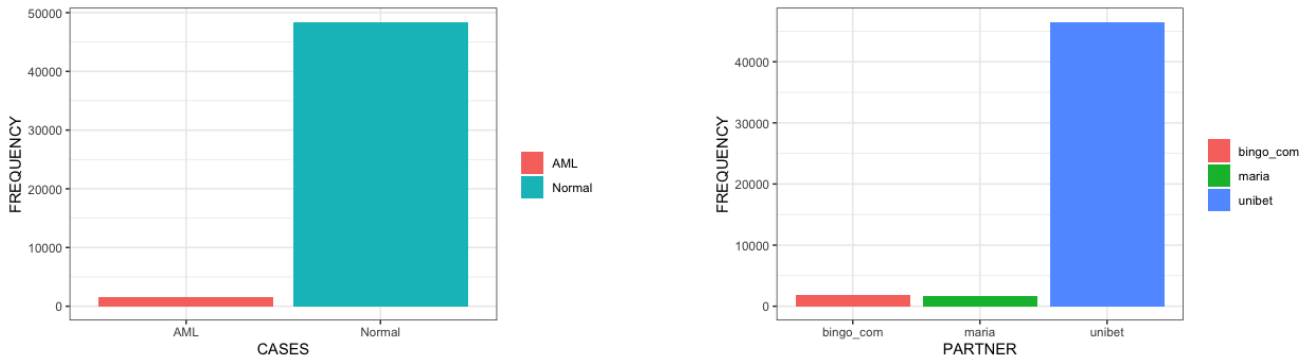
Players Dataset

We begin the analysis of the provided data with the players dataset. This group of data contains general information about players, such as their registration date, the platform and the currency they use to make their transactions and finally if they have been flagged with an IRR (AML cases). In this study, we present the AML players as the positive class and normal players as the negative class. In total, the dataset contains samples of 50,000 gambling players for the period between the 1st of April of 2018 until the 30th of March of 2019.

Table 3.2: Attributes of Players Dataset

Field	Description	Datatype
PLAYER_ID	ID of a Player	Integer
PARTNER_ID	ID of a Partner	String
REGISTERED_DATE	Registration Date	DateTime
VIP_LEVEL_ID	ID of VIP player	Integer
PLAYER_CURRENCY_ID	Transaction Currency	String
AML_CASE	AML Label	Integer

As shown in Figure 3.2a, AML players are the minority in our dataset. More precisely, of the 50,000 samples in the dataset, only approximately 3.2% is identified with high-risk of money laundering, which translates to 1,582 players.



(a) Count Plot for AML cases

(b) Count Plot for partners

Figure 3.2: Figure 3.2a shows the AML and Normal cases of players we had in our data. Figure 3.2b shows what platforms our users are registered with.

From Figure 3.2a it is evident that the dataset was highly imbalanced, meaning that the ratio of Normal players (majority class) to fraudulent (minority class) players was very high – in short, that there were very few fraudulent players. This supports the assumption of the imbalanced

class issue in fraud related problems. Training a binary classifier without handling the class imbalance problem could lead to misleading classification results, since the model will be biased towards the majority class.

Further, as Figure 3.2b shows, the majority of players in the dataset $\approx 93\%$ were registered with Unibet , while the rest were with Maria Casino ($\approx 3.3\%$) and bingo.com ($\approx 3.7\%$). In addition, around 99% of the players with high-risk for money laundering were registered with Unibet. Therefore, since the majority of the overall players' population were betting on Unibet, we decided to exclude any individual that was not registered with Unibet which leave us with 46,512 samples.

Detection Dataset

The detection dataset includes information regarding flags automatically generated by the rule-based system. These flags (only Unibet platform), as mentioned in Section 3.1, are raised when specific thresholds and rules are breached. In total, 6,671 flags had been raised for 2,307 players, indicating that some individuals were flagged more than once. According to Kindred's compliance process, after a risk flag is generated, the case is reviewed by the AML team, who is responsible for deciding whether an internal risk report should be submitted or not. Evaluating all the flags for each player could be extremely time consuming, meaning that higher quality flags are needed to improve the efficiency of the process.

Table 3.3: Attributes of Detection Dataset

Field	Description	Datatype
PLAYER_ID	ID of a Player	Integer
AML_DETECTED_DATE	Date that flag was raised	Date

In this research, we defined FP as cases where players were flagged by the automatic system and managed to pass the de-risking process (i.e. no IRR flag). Similarly, we defined false negatives FN as cases where an IRR was raised for a player after being flagged by an employee but the player was not initially detected by the rule-based system. We summarise the outcome of the rule-based system for the examine period in Table 3.4. This is set as the benchmark upon which

Table 3.4: Results observed from the rule-based system. In total the system in the span of one year raised 6,671 flags which correspond to 2,307 players. Out of those flags 1,104 flags resulted to false positive where 269 players were missed by the rule-based system.

Rules-based detection System	Total Number of Flags
Total Number of Flags Raised by the system	6,671
Unique Number of Players that are flagged	2,307
Total Number of False Positives	1,104
Total Number of False Negatives	269

we attempt to enhance through machine learning algorithms. The primary goal of this research is to improve the identification rate of high-risk cases. Operators and gambling regulators have been working to making online gambling a safer place, meaning better detection of criminal procedures.

Transaction Dataset

In Table 3.5, we list the attributes of the transaction dataset provided by Kindred. Behavioural analysis of gamblers begins with identify how they deposit or withdraw money, whether they are aggressive or they deposit large amounts of money and the frequency with which they perform each of these actions. Subsequently, it is important to analyse and extract transactional behaviour indicators. The transaction dataset has timestamps of every completed transaction as well as the type of transaction (deposit or withdrawal), currency, amount, player account balance before and after, status, source and location of execution.

Before beginning the analysis, we split the transactions into Normal and AML players' transactions. Statistical analysis was performed separately for the two groups (Normal and AML) in order to find similarities and differences at the transactional level. In total, we analysed 2,208,419 transactions which as expected, the majority of those were executed by the Normal group (1,900,190).

Cash-based payment methods, such as prepaid cards, as well as emerging payment methods such as digital currencies, allow customers to deposit money without having to rely on traditional bank accounts (where typically enhanced KYC checks will have been undertaken in a face-to-

Table 3.5: Attributes of transaction dataset

Field	Description	Data type
TRANS_ID	ID of the transaction	Integer
PLAYER_ID	ID of a Player	Integer
PARTNER_ID	ID of the Partner	Integer
CURRENCY_ID	3 letter iso Code	String
DATE	Date of transaction	DateTime
AMOUNT	Amount of transaction	Float
OPEN_BAL	Balance before transaction	Float
CLOSE_BAL	Balance after transaction	Float
STATUS	Status of the transaction	Boolean
GEOLOCATION	latitude and longitude	Float
DEPOSIT_SOURCE	Source of deposits	String

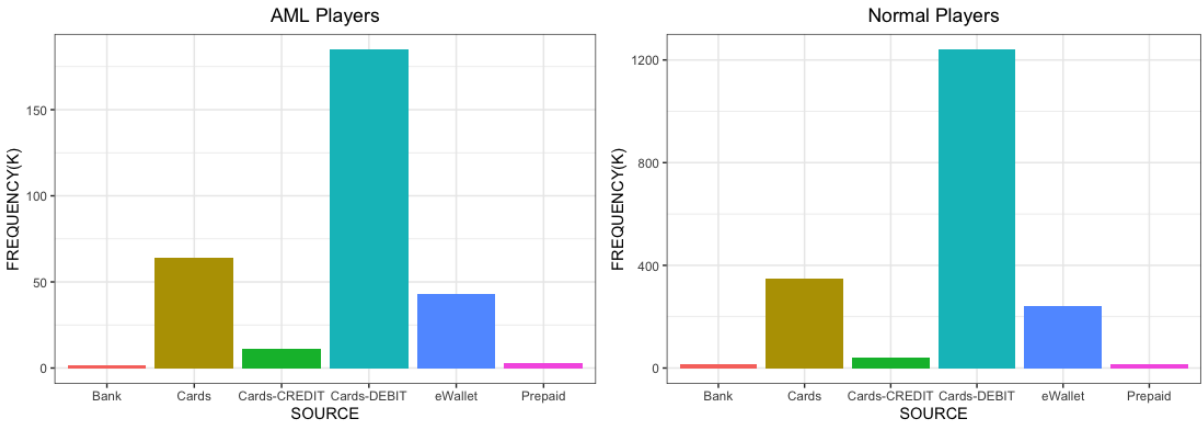


Figure 3.3: Debit sources for the AML and Normal Group

face manner). As a result, today the source of funds of online gambling customers can remain unknown and difficult to trace. Figure 3.3 shows the debit sources used by players to execute their transactions. Despite the general consensus in the gambling industry [7] that the use of e-wallets and prepaid cards to deposit money is associated with higher-risk for money laundering, debit cards are the most popular method for adding funds to individual accounts, as Figure 3.3 suggests, at 64.49% for the AML group and 65.22% for the Normal group. Of the riskier deposit methods, prepaid cards were the least common for players in both groups at 0.74% and 0.66% for the normal and AML groups, respectively. E-wallet transactions were - third most popular method in both groups at 14% for the AML group and 12.6% for the Normal group. No concrete conclusions can be drawn from the payment preferences of the two groups of players.

Next, we examined the distribution of transaction amount for the two groups. Due to the

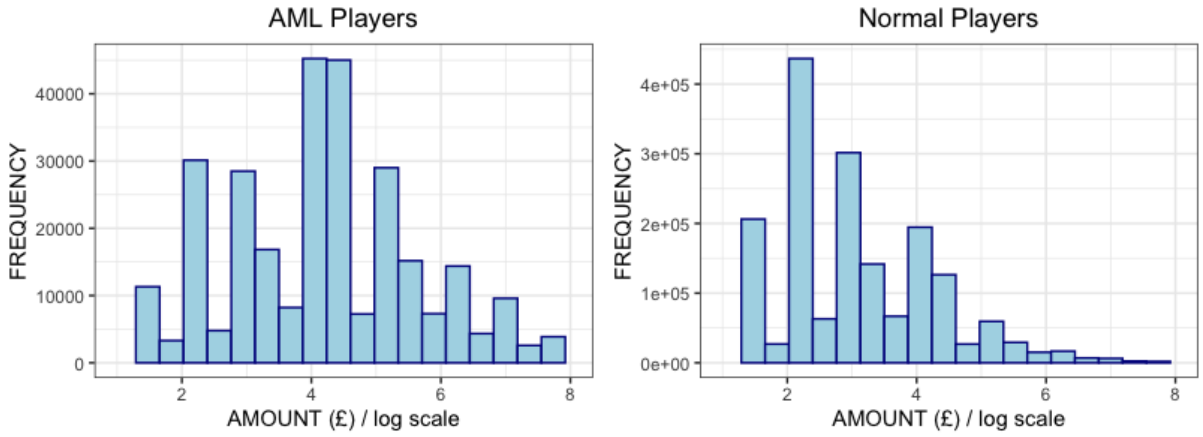


Figure 3.4: Histogram of transactions amount for players with high-risk for money laundering and low-risk for money laundering. We used log transformation on the data to reduce the skewness.

high number of transactions, we randomly select a sample of 1,000 players from each group to visualise the distributions of the amount deposited or withdrawal for the two groups of players (see Figure 3.4). In general, the transaction amount in both groups is right skewed which means we expect many small amount transactions with a few higher amount transactions. Since both groups data distributions are heavily skewed, for visualisation purposes we used log transformation to reduce the skewness. Taking logs brings in the extreme values on the right (high values) relative to the median, while values at the far left (low values) tend to get stretched back, further away from the median. From the first graph, the AML group has more symmetrical distribution with median at 4.5, while the distribution of the Normal group is still positively skewed with median around 3.8 indicating large number of small transactions.

We further analysed the transaction amount by the type of transaction. Figure 3.5 is a boxplot showing the deposit and withdrawal amount distribution for the AML and Normal group. By definition a boxplot is a standardised way to display the distribution of data based on a five-number summary (minimum, first quartile, median, third quartile and maximum). In addition, it can show outliers and their values and can indicate whether a dataset is symmetrical, how tightly data are grouped, and whether and how data are skewed. Again we use log transformation to transform the data to log scale for visualisation purposes. The transactions in the AML group have higher median for both deposits and withdrawals compared to the Normal group.

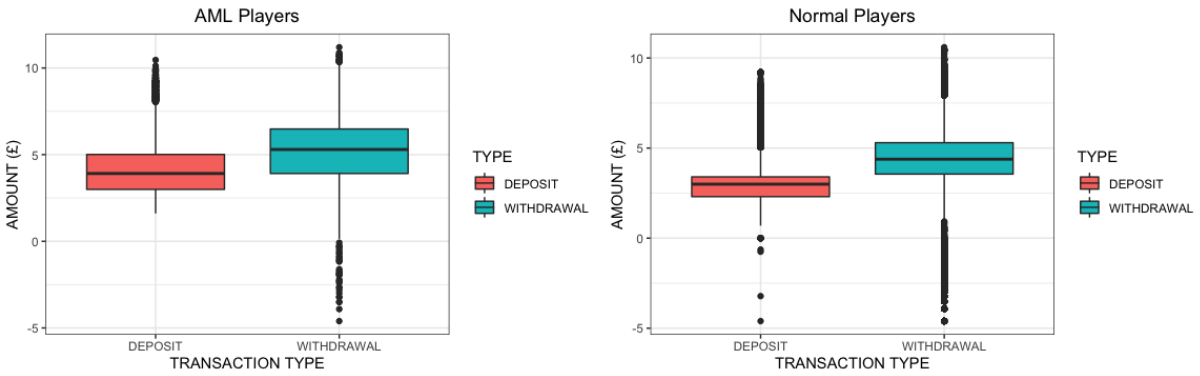


Figure 3.5: Box Plot of transaction amount for AML and Normal Group. We use log transformation on the actual amount since data were positively skewed. AML group has higher median values for both withdrawals and deposits.

A comprehensive way to consider a player spending behaviour is to derive some features using transaction aggregation strategy [125]. The idea of aggregation strategy is based on grouping the transactions executed for a specific period, by transaction type followed by calculating the total amount during that period. The process of aggregating features from the transaction dataset is presented in detail in Section 3.3.

Gaming Dataset

Similar analysis to the transaction dataset is repeated for the gaming dataset with attributes presented in Table 3.6. The gaming dataset, unlike the transaction dataset, presents the data describing a player's betting activity in aggregated form. All bets are grouped by gaming session ID. The total number and amount of bets per session, winnings and any bonuses that individuals have activated are included in the dataset. The start and end date for each session describe the first and last time a bet was placed or paid out within the session. Further, all recorded bets are divided into casino and sportsbook bets, with sportsbook bets dominating the preferences of the players as shown in Figure 3.6. Money laundering risks in the gambling sector are not restricted to casino games. There are a number of ways in which sports activities may be targeted for money laundering, including betting activities. According to Financial Action Task Force [126] sports that could be vulnerable to money laundering problems are either big sports (worldwide like football or on a national basis like cricket, basketball or ice hockey) or sports like boxing.

Table 3.6: Attributes of Gaming Dataset

Field	Description	Data type
PLAYER_ID	ID of a Player	Integer
PARTNER_ID	ID of a Partner	Integer
PRODUCT_ID	ID of a Product	String
SESSION_ID	ID of a session	Integer
START_DATE	Start timestamp	DateTime
END_DATE	End timestamp	DateTime
NUMBER_OF_BETS	Total number of bets	Integer
TOTAL_BET_AMOUNT	Total bet amount	Integer
TOTAL_WIN_AMOUNT	Total win amount	Integer
BONUS_BET_AMT	Bonus amount bet	Integer
BONUS_WIN_AMT	Bonus amount won	Integer

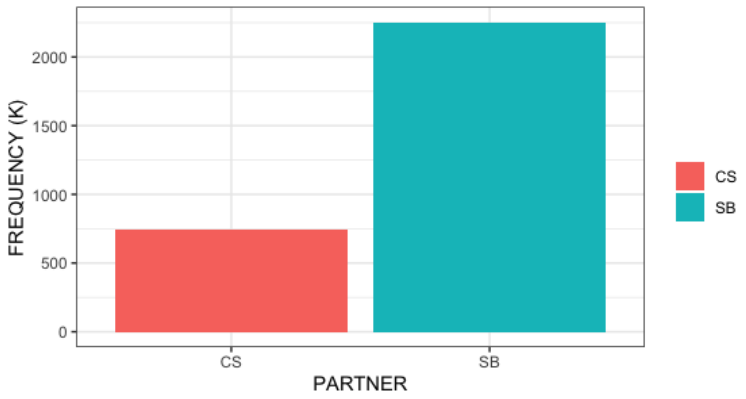


Figure 3.6: Counter plot of casino bets (CS) and sportsbook bets (SB)

The speed and frequency of the gambling opportunity within a game could impact the money laundering risk. Activities that permit high frequency participation are more likely to be associated with harm and more readily facilitate problematic behaviour, such as loss chasing where activities with high stakes could be associated with money laundering. Figure 3.7 compares the distributions of the log transformation of AML and Normal group in respect to: (a) total bet amount, (b) number of bets and (c) total win amount. As in the transactions dataset, the distributions were constructed by randomly selecting 1,000 players from both groups.

Beginning the analysis with total bet amount per session distribution, it is evident that there is a high probability that a high stakes bet can be accumulated by the AML compared to Normal Group. A similar pattern can be observed in the total win distribution with AML group having a higher median compared to the Normal group. Finally, almost identical is the distribution of the number of bets that the two types of players are generating. Although, the distributions

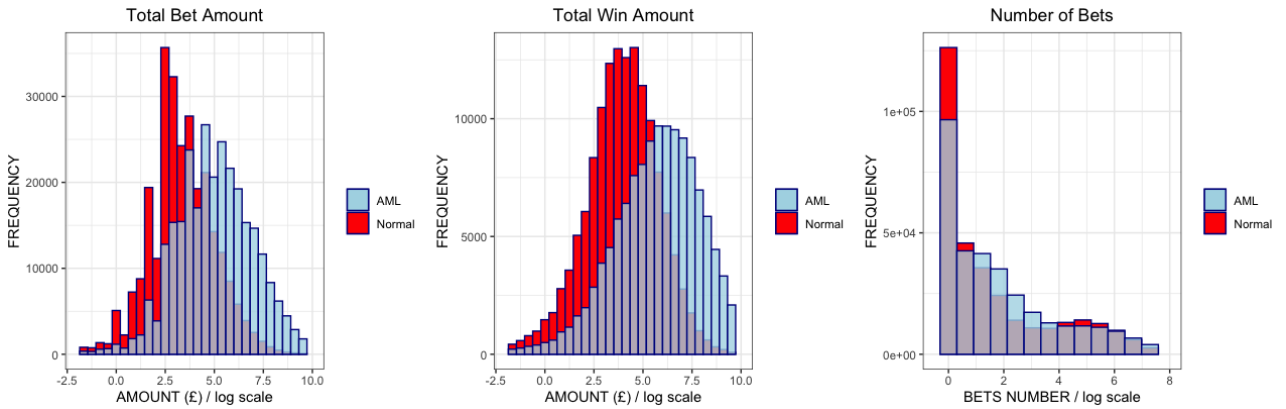


Figure 3.7: Histograms for gaming dataset with log transformation to reduce the skewness in the data.

of Figure 3.7 can provide a good indication and distinction in how the two cohorts of players are behaving within a gambling session, we are not taking to consideration any time variable. Based on these 3 attributes from the gaming dataset, more gambling features will be developed as we show in Section 3.3.

3.3 Derivation of the Global Scope of Gambling Features

Several gambling factors coming from the datasets analysed in the previous sections were considered and converted to gambling features that could potentially serve as inputs to a machine learning model. However, the challenge now is to find patterns within those features that can assist in identifying customers at high-risk. As previously mentioned in literature, in [7] and [127], there are certain customer behaviours that could impose such as risk:

- Increased risk is observed when customers deposit funds and then do not use their account for a significant period of time before looking to withdraw the funds at a later date.
- Risk is associated with drastic changes in betting behaviour (e.g. unusually high activity compared to the expected activity for the customer in question).
- High value deposits in a single transaction or cumulative deposits over a short time frame can heighten risk.

- A customer may deposit illegally acquired funds into his account and then withdraw them without wagering in any product following minimal play or low risk wagering activity by covering all outcomes or using the cash-out feature.

Based on the research from Braveman [11] and Dragicevic et al. [12] five different behaviour indicators were defined to describe players' behaviours in online gambling: (a) 'trajectory', (b) variability in behaviour ('volatility'); (c) time spent in gambling sessions ('time factor'); (d) amount spent in a period of time ('intensity'); and (e) wins and losses ('profitability'). Across these five risk indicators, we generate a total of 36 factors that can encapsulate the absolute level of activity, the statistical significance of a change in gambling behaviour and variables that capture the scale of any such change in behaviour. The global universe of features developed in this research is presented in Table 3.7 along with descriptive statistics for each feature grouped by the two types of players.

When aggregating customers transactions and bets, there is an important question on how much to accumulate, in the sense that every new information may diminish as time passes. However, since every customer has been active for different periods of time meaning some players have been only registered for just a few weeks while others have been active for over a year, we decided to do the aggregation on days where customers have either executed a transaction or placed a bet. Certainly as time passes, information may lose its value, nevertheless with our feature universe we are trying to construct the general betting profile of a player.

As Table 3.7 shows, to capture players 'trajectory' the feature universe includes the averages of deposits, withdrawals, wallet balance after a withdrawal, wallet balance after a deposit or bets. In addition, we have added features describing the players cumulative characteristics such as total deposit amount, total withdrawal, total bet and total winnings. The average deposit and withdrawal per transaction corresponds to the total amount deposited or withdrawn throughout the duration of a customer's recorded activity over the total number of deposits and withdrawals. Similarly, the average amount per bet, the average amount of winnings or the average wallet balance are calculated. These feature are part of the 'trajectory' risk factor. To capture 'volatility' in players' behaviour, the standard deviations (STD) of deposits,

Table 3.7: Descriptive statistics of Global Feature Space

	AML			Normal		
	Mean	Median	SD	Mean	Median	SD
Gambling Fraud Dataset						
AVERAGE_AMOUNT_DEPOSIT	505.71	209.05	929.83	58.42	24.63	157.63
TIME_DEP	2.02	0.79	4.51	9.20	3.44	15.00
WALLET_BALANCE_DEP	569.39	215.62	1389.68	65.89	26.30	223.62
AVERAGE_AMOUNT_WITH	2028.91	933.33	3223.86	257.69	102.00	670.30
TIME_WITH	6.34	3.17	10.03	12.18	2.27	21.37
WALLET_BALANCE_WITH	749.83	141.64	2560.76	54.06	9.33	300.31
TOTAL_AMOUNT_DEPOSIT	34415.14	18393.50	58455.60	1298.34	335.00	3223.90
TOTAL_AMOUNT_WITH	31313.95	13970.66	59984.53	1278.54	326.60	4248.40
WITH_DEP_RATIO	41.47	1.34	1195.45	116.63	1.05	2382.45
COUNT_DEP	187.03	96.00	261.00	36.16	11.00	87.43
COUNT_WITH	38.42	13.00	127.62	6.31	3.00	16.03
AMOUNT_STD_DEPOSIT	383.42	169.63	615.86	35.10	12.23	103.32
AMOUNT_STD_WITH	1331.34	602.06	2045.79	117.04	31.82	367.49
PERC_CHANGE_DEP	63.24	24.00	161.29	13.82	6.00	108.48
COUNT_PAYMENT_METHODS	2.42	2.00	0.83	2.04	2.00	0.59
TRANS_TOTAL_TIME	142.12	103.85	118.02	120.01	78.05	116.20
COUNT_FAIL_TRANS	225.45	117.00	325.75	42.47	14.00	97.72
NUM_DEP_TOTAL_TIME RATIO	2.60	1.22	9.07	2.13	0.23	24.97
TOTAL_AMOUNT_WIN	219539.38	99807.81	493892.05	6696.58	1200.77	20580.30
COUNT_BETS	18490.91	2712.00	48599.70	3521.07	198.00	15804.42
TOTAL_AMOUNT_BET	226107.95	105890.10	502228.62	6841.17	1211.60	20862.84
BET_TOTAL_TIME	144.65	110.46	118.99	122.66	80.17	118.39
COUNT_SESS	246.08	113.50	341.18	62.40	23.00	128.62
NUM_SESS_BET_TIME_RATIO	2.43	1.52	3.35	3.09	0.50	33.60
AMOUNT_STD_BET	5053.91	1655.05	13042.32	287.33	52.20	1241.13
WIN_DEP_RATIO	8.53	5.38	22.97	8.29	3.64	186.95
WINNING_OVER_LOSING	9.14	0.73	24.57	4.30	0.56	14.96
DEPOSIT_BET_RATIO	0.34	0.17	2.16	0.58	0.29	10.76
BET_TIME_DIFF	0.06	0.04	0.04	0.03	0.02	0.03
DEPOSIT_ACTIVE_DAYS_RATIO	1541.78	693.42	2897.99	118.43	40.00	349.57
DIFF MAX BET MIN BET	28332.05	10532.00	78380.58	1217.91	225.40	4298.46
DIFF_MAX_DEP_MIN_DEP	1661.60	900.00	2415.26	109.63	40.00	278.91
PERIODIC_MEAN_SESSION	17.52	18.30	4.63	17.16	17.71	4.29
PERIODIC_MEAN_WITH	15.97	18.41	6.86	16.03	18.04	6.47
PERIODIC_MEAN_DEP	16.46	17.62	5.48	16.23	16.94	4.79
RISKY_METHOD	0.28	0.00	0.45	0.22	0.00	0.42

withdrawals and bet amount are produced.

We note that just this aggregation is not enough, in the sense that these features cannot describe all the risk factors defined previously. A combination of features was taken into consideration to capture data risk indicator of 'profitability' with following ratios being created: a) Winning

losing ratio (total number of winning bets over total number of losing bets), b) Winning over deposit ratio (total amount won over total amount deposited), c) Deposit over withdrawal ratio (total amount deposited over total amount withdrawal). Further, we expand the features' scope to include behavioural variables to capture 'intensity' in the betting and transactional level with total number of active betting sessions over the total betting period, the total number of bets over the total betting period, and total deposit over active days ratio. When using aggregated features, there is still some information that is not completely captured. To be more precise we are interested to analyse the time of a bet or a transaction. It is expected from a player to deposit or bet at similar times. We are interested to find whether a particular time exist for which players with high-risk are more keen on to deposit, withdrawal or bet. We use the von Misses distribution which is known as the periodic normal distribution to model the mean time of a bet or a transaction [128]. The von Misses is defined as

$$D \sim (\mu_{\nu M}, \frac{1}{\sigma_{\nu M}}) \tag{3.1}$$

where $\mu_{\nu M}$ is the periodic mean and $\sigma_{\nu M}$ is the periodic standard deviation. We chose to use periodic mean to model the average time since it provides a much better estimation compared to arithmetic mean. Figure 3.8 shows the improvement in the estimation of the time mean when periodic mean was used compared to the arithmetic mean. One of the main characteristics of money laundering behaviours as it was stressed through our interview process with industry stakeholders [7] was that high-risk players have the tendency to stay inactive for large period of time or stay highly active for a short of time. To be able to identify patterns that are inline with such behaviours our feature scope contains the 'trajectory' which includes the average time that passes between consecutive bets, the ratio between first and last deposit or withdrawal over the total number of active days together with the periodic mean features for deposit, withdrawal and bet. Also, we keep track of the mean average time between successive deposits (TIME_DEP feature from Table 3.7) and successive withdrawals (TIME_WITH). We expand the features outline the 'trajectory' risk indicator by recording the biggest difference between deposits, bets and the biggest percentage change between successive deposits. As the literature suggests [7], certain payment methods i.e. e-wallets or prepaid cards, could impose higher risk

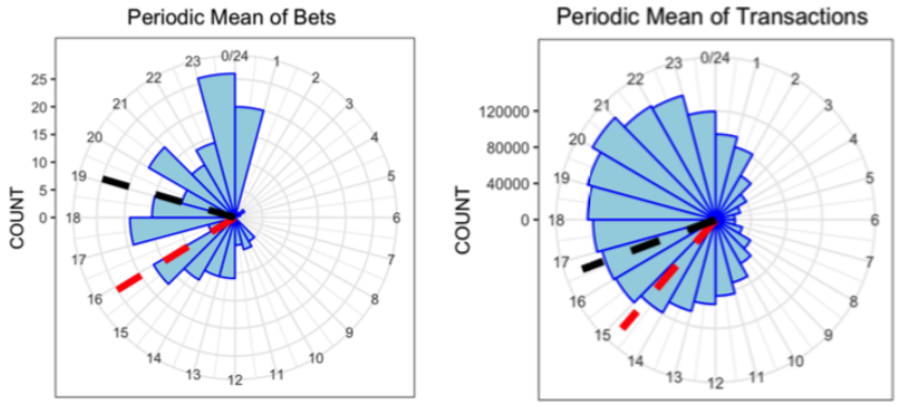


Figure 3.8: Histogram of the time distribution for transaction and bets for random player. The black-line represents the periodic mean and the red-line the arithmetic mean. It is evident the improvement in the estimation of the mean with the periodic mean.

for money laundering. For this purpose we included a binary variable called Risky Method in our scope that describes if a player has used either a prepaid card or an e-wallet to execute a transaction. Finally, we included features such as the number of deposits, withdrawals, betting sessions and number of fail transactions.

3.3.1 Feature Selection

After extracting the features, we had to find a feature set that was most relevant to the degree of money laundering. Adequate feature selection is an important step not only to prevent overfitting the model but also to accelerate the training and help us understand how the machine learning model makes a decision.

Feature selection is a key concept in machine learning that can influence the performance of a machine learning model. When a feature is chosen to be part of the training process of a model, it can have either a positive (relevant) or negative (irrelevant) impact on performance. Random forests are one the most popular machine learning algorithms, providing in general a good predictive performance with low overfitting, and interpretability, meaning is easy to compute how much each variable is contributing to the final decision. As part of the initial analysis of the dataset, we used the feature importance property of the random forest classifier [129] to evaluate the impact of the 36 features we developed. The output of this process helped

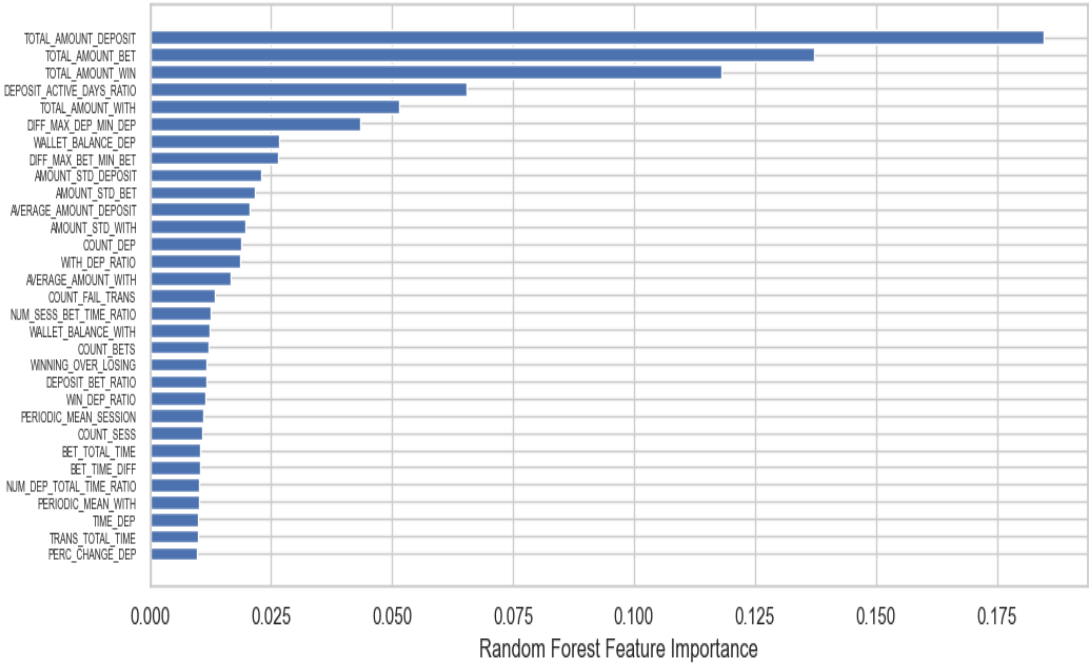


Figure 3.9: Feature importance of Global Scope of features. The last five features importance was insignificant to the solution of our problem. Thus we decided to exclude those features from the experimental dataset. The five features with the smallest importance scores are excluded from the plot.

us to remove any non-essential features. In decision trees, every node is a condition of how to split values in a single feature so that similar values of the dependent variable are in the same set after the split. The condition is based on impurity, which for classification problems is Gini impurity (defined as the probability of obtaining two different outputs at a node). When training a tree, it is possible to compute how much each feature contributes to decreasing the weighted impurity. From the sklearn library [130], we used the feature importance property of the random forest classifier to generate importance scores and the results can be found in Figure 3.9. The five features which the random forest suggested that have the least impact in the prediction and we excluded from the experimental dataset were: 1) the risky method flag, 2) the counter of withdrawals, 3) average time between consecutive withdrawals, 4) the number of payment methods and 5) the average periodic time of deposit.

3.4 Summary

In summary, this chapter proposes a new set of behavioural features that allow a players's behavioural patterns to be captured effectively in an online gambling environment. Further we have presented an overview of the main datasets that Kindred has provided us for this research, together with some statistical analysis. In section 3.1, the current AML framework of our research partners is explained and the weaknesses are identified. As it was stressed, the existing process highly relies on a rules-based automated system to flag individuals with money laundering risk. If criminals are able to adapt to those rules, then it would be fairly hard for someone to detect them. Section 3.2 illustrates an analysis of each provided dataset which helped us to create the global scope of behavioural features that we presented in Table 3.7. Following the feature engineering phase is the feature selection step where feature importance from random forest was implemented to find the most important features from the global scope of variables. At the end, five features were excluded from the final experimental dataset as their importance was identified as insignificant by the random forest.

Chapter 4

Supervised Learning for Fraud Detection: A Comparison

4.1 Introduction

Through all the literature review in Chapter 2, it was persistently shown that pattern recognition models using machine learning algorithms can be effectively used for the identification of fraud. This chapter presents a supervised learning framework for the identification of individuals with high-risk for money laundering. The comparison between six different machine learning techniques namely, logistic regression, multi-layer perceptron, support vector machines, random forest, XGBoost (XGB) and naïve Bayesian is presented. The rationale behind the use of those techniques is that these algorithms have been used successfully in previous studies and they represent a wide spectrum of complexity and interpretability.

In this chapter we aim to demonstrate that: (1) A supervised machine learning algorithm in conjunction with a validated synthetic data generation method to create synthetic fraud data is experimentally more accurate than the existing rule-based method of Kindred for flagging high-risk players. (2) Feature engineering and selection phase was a success by evaluating the performance of machine learning models on the new generated features; (3) existing explainability technique, specifically, SHAP, can be applied to provide insights on the machine

learning model's predictions. Section 4.2 describes the proposed supervised learning framework to predict fraud in online gambling. Section 4.3 provides an overview of the techniques used for classification. Subsequently, in Section 4.4 the experimental results are displayed and a comparison with the existing detection system is presented in Section 4.5. Finally, in Section 4.6, we conclude with the discussion on the limitations of the supervised learning approaches for predicting high-risk for money laundering players.

4.2 Proposed Fraud Detection Framework

The aim of a fraud detection system (FDS) in online gambling is to monitor and identify suspicious behaviour while simultaneously limiting the possibility of too many false alarms being raised. In an online gambling environment, a fraudster attempts to perform a series of actions without being detected, while the FDS tries to recognise any fraudulent behaviour. As a step of these functions, a detection framework is proposed in Figure 4.1.

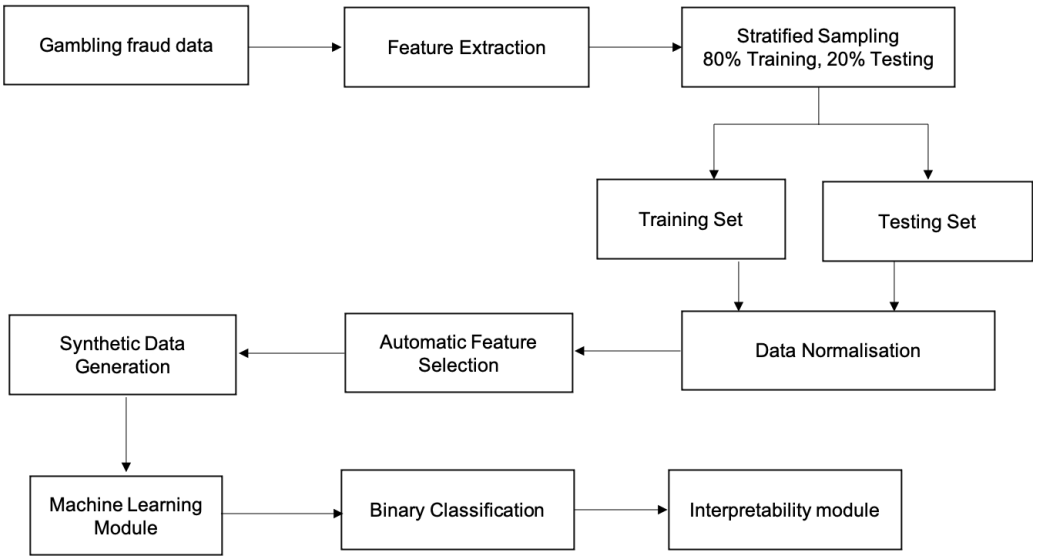


Figure 4.1: Proposed supervised learning framework: gambling data are pre-processed, and new features are created. After the feature engineering stage, the data are split into training and testing sets and normalised, after which feature selection is performed. Then, a synthetic data generation technique is applied before training a machine learning algorithm. The binary output of the trained model will indicate whether an individual is at risk for performing money laundering. Finally, to interpret the results from the machine learning algorithms, a model agnostic approach is applied to understand which features are responsible for the output of the algorithms [2].

To detect gambling fraud, different machine learning classifiers were examined to find the ideal method because, as the literature suggested in Chapter 2, different techniques have been proven effective when applied for fraud identification. The proposed framework consists of three main components: input, process and output. The discussion of these components is as follows:

1. Input: from the framework, the original data were first pre-processed by running several cleaning steps to ensure their quality and validity. Then, a scope of new behavioural features was extracted, as discussed in Chapter 3, and used to construct the gambling fraud dataset used in our experiments. When the feature engineering phase was completed, the dataset was split into the training and testing sets (out-of-sample). Both sets were then normalised (in the range between zero and one) to bring all variables in the same range. The training set was used as input to train a machine learning model, while the testing set was used to evaluate the machine learning model. Following the feature engineering step was the feature selection stage, wherein the most important features from the global scope were selected using the permutation importance of random forest.

   As expected, the newly processed dataset was highly imbalanced, which is a regular characteristic of datasets related to fraud detection. Neglecting data imbalance could lead to inaccurate classification performance. To tackle the imbalanced class problem, we investigated and applied synthetic data generation techniques in the training set prior to training the machine learning model.

2. Process: in this component, six different machine learning models were trained, and their ability to identify high-risk money laundering players was investigated. The performance of the supervised learning algorithms was then analysed and compared, and the best model was selected.

3. Output: as indicated in Figure 4.1, the output from the machine learning models was used to classify the players into two groups, the AML and the Normal group (binary classification). Finally, to verify that our selected machine learning model was unbiased, a model agnostic approach was applied to explain the classification results.

To evaluate the effectiveness of the proposed detection framework, the results from the machine learning techniques were compared against the results of the rules-based detection system on the same test data. The aim of this chapter is to show whether the machine learning algorithms could improve the identification rate of high-risk players.

4.3 Supervised Learning for Classification

As part of the experimental process, preliminary experiments were conducted using machine learning algorithms for predicting high-risk players. As stated in the background research established in Chapter 2, several machine learning methods have been applied to fraud detection with success. However, one of the challenges of deploying machine learning for fraud detection is selecting the appropriate technique.

We performed experiments with six supervised classifiers based on the fact that they are commonly used in the literature for fraud detection problems: logistic regression (LR), random forest (RF), support vector machine (SVM), naïve Bayes (NB), extreme gradient boosting (XGBoost) and multi-layer perceptron (MLP). In this section, an overview of each technique is provided.

Logistic regression is a statistical method which is used to find the probability of an event success or failure (binary dependent variable i.e. 0/1). It supports categorising data into discrete classes by studying the relationship in a given set of labelled data. It learns a linear relationship from the given dataset and then introduces a non-linearity in the form of the sigmoid function [131] which is defined in equation 4.1.

$$\sigma(x) = \frac{1}{1 - e^{-x}} \tag{4.1}$$

Random decision forest is an ensemble learning method wherein randomly trained decision trees are combined. Each decision tree is slightly different from the others, aggregated into an ensemble of trees. This diversity leads to decorrelation between the trees, which is desired, as

different trees have different judgements about the same problem and their combined opinions make stronger and more reliable predictions about the testing points. Therefore, the algorithm is improved in terms of generalisation and robustness. A forest model consists of the same components as a decision tree, so the weak learners (test functions), energy model, leaf predictors and type of randomness influence the prediction/estimation properties of the forest [129].

One of the key points of the RF algorithm is that it can be applied to both classification and regression tasks. It starts at the rood node of a tree considering all the data. Each predictor variable is then estimated to see how well it separates two different nodes [132].

XGBoost is an implementation of gradient-boosted decision trees designed for speed and performance. The main idea of boosting is to sequentially build sub-trees from an original tree so that each subsequent tree reduces the errors of the previous one. Thus, the new sub-trees will update the previous residuals to reduce the error of the cost function [133]. The objective function of the XGBoost, denoted by L, is given as follows:

$$L = \sum_{i=1}^{n} l(y_i, \hat{yi}) + \sum_{k=1}^{K} \Omega(f_k) \tag{4.2}$$

By minimising the objective function L, the regression tree model functions f_k can be learned. The training loss function $l(y_i, \hat{y}_i)$ evaluates the difference between prediction $\hat{y}_i$ and the actual value y_i. The term Ω is used to avoid the overfitting problem by penalising the model complexity as follows:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \left\| w \right\|^2 \tag{4.3}$$

where γ and λ are the regularisation parameters and T and w are the number of leaves and the scores on each leaf, respectively.

Support vector machines is a supervised machine learning algorithm that can be used for both classification and regression tasks [134]. However, it is most popular for its classification ability. It uses a linear model to implement non-linear class boundaries through non-linear mapping of input vector x into the high-dimensional feature space. In this algorithm, each data item is plotted as a point in an n-dimensional space (where n is the number of features

present), with the value of each feature being the value of a particular coordinate. A linear model constructed in the new space can represent a non-linear decision boundary in the original space. In the new space, an optimal separating hyperplane is constructed. The points on either side of the separating hyperplane have distances to the hyperplane. The smallest distance is the margin of separation; q is the margin of the optimal hyperplane. The points that are distance q away from the hyperplane are the support vectors. All other training examples are irrelevant for defining the binary class boundaries [135].

Naïve Bayesian classifier is a statistical Bayesian classifier algorithm [136]. It is called naïve because all variables are mutually correlated and contribute towards the classification which is known as the conditional independence assumption [137]. The naïve Bayes Classifier is based on Bayes' Theorem. It is not a single algorithm but a family of algorithms that share a common principle, i.e. every pair of features being classified is independent of the other. Bayes theorem finds the probability of an event occurring given the probability of another event that has already occurred. It is stated mathematically by the following equation:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{4.4}$$

where P(A) is the prior of A (the prior probability, i.e. the probability of an event before evidence is seen). The evidence is an attribute value of an unknown instance (here, it is event B). $P(A|B)$ is a posterior probability of B, i.e. the probability of an event after evidence is seen. Bayes theorem can be applied as follows.

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)} \tag{4.5}$$

where, y is the class variable and X is a dependent feature vector (of size n). Now, the evidence is split into independent parts. Therefore, if two events are independent, then:

$$P(y|x_1, ..x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1)P(x_2)...P(x_n)} \tag{4.6}$$

An artificial neural network is a function approximation method inspired by the way bio-

logical nervous systems, such as the brain, process information. It consists of interconnected neurons that work together to solve a specific problem [138]. A neural network in its simplest form contains two layers: a) the input layer and b) the output layer. Different structures of neural networks are implemented with the most commonly used configuration, the multi-layer perceptron (MLP).

Multi-layer perceptron is a feedforward supervised type of neural network capable of approximating generic classes of functions, including continuous and integrable functions [139]. Multi-layer perceptrons have a hidden layer and can deliver outputs with more than two classes. The hidden layer should be complex enough, i.e. contain sufficient neurons, to understand the input features and generate the different classes of outputs, but an excessive number of neurons could lead to overfitting. The complexity of a neural network always depends on the problem we are trying to solve. An MLP utilises a supervised learning technique called backpropagation for training. Equation 4.7 defines a single perceptron that can accept multiple inputs and produce a single output:

$$\sum_{i=1}^{n}(I_i * w_i) + \theta \tag{4.7}$$

where I_i is the input features in the neuron, w_i the weights to be optimised when training a neural network, and θ is the bias. An MLP architecture can include a number of perceptrons. The network is trained on a set of paired data to determine the input–output mapping. The weight of the connections between neurons is then fixed, and the network is used to determine the classification of a new set of data [137].

Table 4.1 lists some of the most important advantages and disadvantages [140, 141, 142, 143] of each method described in this section.

Table 4.1: Advantages and disadvantages for supervised learning algorithms

	Advantages	Disadvantages
Logistic Regression	-Easy to implement -Interpretable (weights of each feature) -Less prone to overfitting	-Assumes of linearity between the dependent variable and the independent variables -Only be used to predict discrete functions
Random Forest	-Decision Trees are very simple and fast -It does not require any domain knowledge or parameter setting and it is able to handle high dimensional data -Representation is easy to understand i.e. comprehensible.	-For very large datasets, the size of the trees can take up a lot of memory -It can tend to overfit, so you should tune the hyperparameters
XGBoost	-Good execution speed -Less prone to overfitting -Fast to interpret	-Many hyperparameters to tune -Sensitive to outliers
Naïve Bayes	-It requires short computational time for training and very easy to construct. -Not needing any complicated iterative parameter estimation schemes, so can be applied to large data set. -Easy interpretation of knowledge representation	- Theoretically, naïve Bayes classifier has minimum error rate comparing to other classifiers, but practically it is not always true, because of the assumption of class conditional independence and the lack of available probability data - Less accurate compared to other classifier
SVM	- Performs well when there is a clear margin of separation between classes. -Effective in high dimensional spaces. -Memory Efficient	-Not suitable for large datasets -It does not perform well on noisy data -No probabilistic explanation for the classification
MLP	-Tolerates noisy data as well as able to classify patterns - Can used when we have the little knowledge of the relationship between attributes and classes -Good classification performance	-Involves long training time -Poor interpretability -A lot of parameters to tune

4.4 Experiments

A successful machine learning framework for fraud detection must meet three important requirements [144]: a) it should calculate and provide predictions fast enough to support decision-making; b) it should have a continuous learning ability; and c) there should be a mechanism for feature selection. These requirements can be met with the use of a supervised machine learning algorithm. Section 4.4.1 provides an overview of the dataset used in the experimental process, and in Section 4.4.2, the experimental settings are defined. Section 4.4.3 illustrates the results and a performance summary of each method examined. Section 4.4.4 and Section 4.4.5

presents the results when oversampling and undersampling techniques are used for balancing the training dataset.

4.4.1 Experimental Dataset

The dataset developed from real data in Chapter 3 was used in the experiments conducted in this chapter. The dataset describes the transactions and aggregated betting sessions of players for the period of to 01-03-2018 to 31-03-2019 and contains 15,200 samples of which 1,200 have been flagged as high-risk (positive). We summarise the dataset information in the Table 4.2.

From Table 4.2, it is evident that the total number of samples included in the final experimental dataset was smaller than the initial 50,000 population of gambling players referenced in Section 3.2. In the experimental dataset, we excluded a) any player who was not registered with Unibet, b) players with less than four events (including at least two deposits and two withdrawals), c) players whose registration date was less than one week from the ending period of the dataset and d) players who deposited less than £100 since they posed no money laundering risk. Since our dataset contained far fewer high-risk individuals than Normal group gamblers, it could be described as highly imbalanced, which could lead to unintended model behaviour, such as classifying all gamblers as normal, and superficially maximise the accuracy of the prediction on the dataset even though it would not be useful in a real-world context [14].

4.4.2 Experimental Design

Three sets of experiments were conducted to evaluate the ability of supervised machine learning algorithms to classify individuals with a high money laundering risk. (1) We trained and com-

Table 4.2: Real-world gambling dataset. In total 15,200 players have been selected to form the experimental dataset. The IR corresponds to the imbalance ratio between minority and majority class.

ID	Dataset	#Features	#Instances	IR
1	Gambling Fraud	31	15,200	1:11.6

pared the six machine learning algorithms on the imbalanced gambling fraud dataset. (2) We trained and compared the six machine learning algorithms when the dataset was fully balanced by applying SMOTE and ADASYN. (3) We trained and compared the six machine learning algorithms when both oversampling and undersampling methods were applied to balance the dataset.

The results in Sections 4.4.3, 4.4.4 and 4.4.5 represent the average values obtained after 10 runs. We evaluated the results from the machine learning framework against the rule-based system detection flags and split our data into the training (in-sample) and testing (out-of-sample) sets with ratios of 80% and 20% respectively (Table 4.3).

Table 4.3: Training and Testing set final samples

	Number of instances in the dataset	
Label	**Training Set**	**Testing Set**
Normal Group	11,200	2,800
AML Group	960	240

4.4.3 Experiment I: Results of Imbalanced Dataset

In this experiment, we trained all six machine learning models with the imbalanced dataset from Table 4.3. Table 4.4 summarises and compares the classification algorithms in terms of accuracy, precision, specificity (true negative rate), recall (true positive rate) and F1 score. All results were produced when a baseline form of each algorithm was trained without tuning any of the hyperparameters.

As expected, due to the imbalanced nature of the dataset, we observed higher performance on the accuracy, precision and specificity metrics than on the recall or F1. All machine learning models underperformed on the recall, i.e. the true positive rate (which refers to the share of high-risk players correctly classified by the models), with the LR and SVM scoring the lowest at 35% and 30%, respectively (i.e. there were around 160 misclassified positive players). In terms of the precision, LR achieved the highest score at 85% and NB the lowest at 58%, as can be seen

Table 4.4: Classification results ($mean \pm std$) when training set was balanced with imbalance: accuracy, recall, specificity, precision and F1.

	Performance Metrics				
Algorithms	Accuracy	Recall	Specificity	Precision	F1
LR	0.9471 ± 0.0059	0.3573 ± 0.0317	0.9947 ± 0.0010	0.8446 ± 0.0164	0.5012 ± 0.0316
RF	0.9660 ± 0.0022	0.6750 ± 0.0245	0.9896 ± 0.0013	0.8412 ± 0.0118	$\mathbf{0.7486 \pm 0.0120}$
XGB	0.9619 ± 0.0018	0.6730 ± 0.0297	0.9855 ± 0.0026	0.7912 ± 0.0338	0.7265 ± 0.0190
NB	0.9394 ± 0.0038	0.6902 ± 0.0274	0.9600 ± 0.0034	0.5880 ± 0.0169	0.6346 ± 0.0136
SVM	0.9445 ± 0.0029	0.3052 ± 0.0211	0.9971 ± 0.0005	0.8962 ± 0.0239	0.4552 ± 0.0264
MLP	0.9555 ± 0.0043	0.6164 ± 0.0198	0.9833 ± 0.0039	0.7547 ± 0.0457	0.6781 ± 0.0257

in Table 4.4. A high precision score is related to a low number of false positives. To support this claim of a low number of false positives, the specificity was calculated for each model, and a score higher than 98% was achieved by all algorithms. In our validation set, the number of instances from the Normal group of players was around 3,000, so the 98% specificity (true negative rate) could be estimated as representing 20 false positives. Better overall performance was achieved by the tree-based methods of RF and XGBoost, with recall scores around 67% and precision scores of 84% and 79% respectively. These two methods had the best performance in terms of correctly classifying both classes, which was supported by their performance on the F1 score at 74% and 72%, respectively.

The classification accuracy for the AML group from the machine learning models was unacceptable. Nevertheless, this was exactly what was expected since the actual training dataset was weighted in favour of the Normal group. Furthermore, it was unsurprising that all models performed with similar levels of accuracy and did not markedly outperform a trivial majority class model on accuracy (in which all players were assumed to be low risk). Thus, applying oversampling or some other mechanism to balance the dataset is essential. When evaluating an FDS, we must consider both true positive and true negative rates. On the one hand, we do not want to miss a fraudulent case, while on the other hand, we do not want to falsely classify someone as fraudulent. In the next section 4.4.4, we tried to improve the classification results using different data rebalancing techniques.

4.4.4 Experiment II: Results Using Data-Level Solutions

One of the biggest issues in fraud detection, confirmed by the results presented in the previous section, is that there are significantly fewer fraudulent cases than normal cases. The imbalanced dataset leads to unintended model performances, such as classifying most customers as low-risk to achieve almost perfect accuracy. Therefore, the problem is how to improve the identification of the minority class rather than achieve better overall performance. In this section, we repeated the experiment from the previous section but balanced the training dataset prior to training the machine learning models.

As the literature review in Chapter 2 suggested, several data- and algorithmic-level approaches exist to deal with the imbalanced class problems. The proposed FDS framework focuses on data-level solutions, which refers to balancing the data by either oversampling the minority class, undersampling the majority class or using a hybrid solution wherein both approaches can be applied. In this section, we investigated how the classification performance was affected when SMOTE or ADASYN was used to achieve a balanced dataset with an approximately 50:50 split between the AML and Normal groups.

Table 4.5 and Table 4.6 show the results for when the dataset was balanced using SMOTE and ADASYN. In contrast with the results of the first experiment in Table 4.4, the machine learning models classified the samples of the minority-positive class with higher accuracy and a noticeable improvement in the recall score of approximately 18%. Although the specificity performance decreased, with a drop of 3–4 % on average, it remained at a high level, with most techniques achieving a score above 90% with either SMOTE or ADASYN. To summarise, with oversampling, we managed to achieve good classification performance on both the true positive and true negative rates, while in the experiment conducted in the previous section, there was a bias towards the majority class.

Regarding the results, note that the models underperformed in terms of precision, which affected the overall F1 score. This could be explained by the fact that precision is more sensitive to FPs compared to specificity since the number of TPs in the precision numerator is much smaller

Table 4.5: Classification results ($mean \pm std$) when training set is balanced with SMOTE: accuracy, recall, specificity, precision and F1.

	Performance Metrics				
Algorithms	Accuracy	Recall	Specificity	Precision	F1
LR	0.9130 ± 0.0060	0.8677 ± 0.0154	0.9169 ± 0.0064	0.4731 ± 0.0201	0.6120 ± 0.0168
RF	0.9429 ± 0.0038	0.8650 ± 0.0189	0.9503 ± 0.0036	0.6195 ± 0.0168	0.7217 ± 0.0126
XGB	0.9517 ± 0.0022	0.8107 ± 0.0242	0.9641 ± 0.0020	0.6668 ± 0.0295	$\mathbf{0.7314 \pm 0.0242}$
NB	0.9361 ± 0.0043	0.7372 ± 0.0323	0.9522 ± 0.0039	0.5547 ± 0.0205	0.6324 ± 0.0163
SVM	0.9151 ± 0.0046	0.9099 ± 0.0152	0.9155 ± 0.0043	0.4891 ± 0.0164	0.6361 ± 0.0166
MLP	0.9301 ± 0.0083	0.8580 ± 0.0250	0.9366 ± 0.0094	0.5562 ± 0.0373	0.6741 ± 0.0295

Table 4.6: Classification results ($mean \pm std$) when training set was balanced with SMOTE: accuracy, recall, specificity, precision and F1.

	Performance Metrics				
Algorithms	Accuracy	Recall	Specificity	Precision	F1
LR	0.9000 ± 0.0049	0.9078 ± 0.0117	0.8993 ± 0.0060	0.4547 ± 0.0054	0.6058 ± 0.0031
RF	0.9371 ± 0.004	0.8801 ± 0.0146	0.9422 ± 0.0044	0.5766 ± 0.0083	0.6967 ± 0.0076
XGB	0.9477 ± 0.0034	0.8158 ± 0.0162	0.9596 ± 0.0020	0.6450 ± 0.0143	$\mathbf{0.7203 \pm 0.0116}$
NB	0.9305 ± 0.0033	0.7589 ± 0.0213	0.9457 ± 0.0036	0.5544 ± 0.0126	0.6406 ± 0.0132
SVM	0.8937 ± 0.004	0.8980 ± 0.0168	0.8933 ± 0.0029	0.4284 ± 0.0165	0.5800 ± 0.0183
MLP	0.9270 ± 0.0024	0.8577 ± 0.0275	0.9332 ± 0.0029	0.5386 ± 0.0117	0.6615 ± 0.0128

than the number of TNs in the specificity numerator.

Now, compared with the two oversampling approaches, better performance was observed when we oversampled the training set with SMOTE, with the differences in several cases being indistinguishable. In both experiments, RF and XGBoost outperformed the other models, with F1 scores of 72% and 73%, respectively.

The SMOTE and ADASYN algorithms depend on the dataset. If there is severe data imbalance, as in the present case, these techniques may not be able to help if the variations within the minority class and the similarities between the two classes are very high. To verify whether these two groups of players can be observed in real-world datasets, we used visualisation methods that project multi-dimensional data points into the low-dimensional space such that the structural properties of the data are preserved. We chose the t-distributed stochastic neighbour embedding (t-SNE) method, as it is one of the most popular methods for projections into newly created dimensions based on the principle of non-linear mapping [59]. The visualisations of the training set before oversampling, after SMOTE and after ADASYN are presented in Figure 4.2, wherein

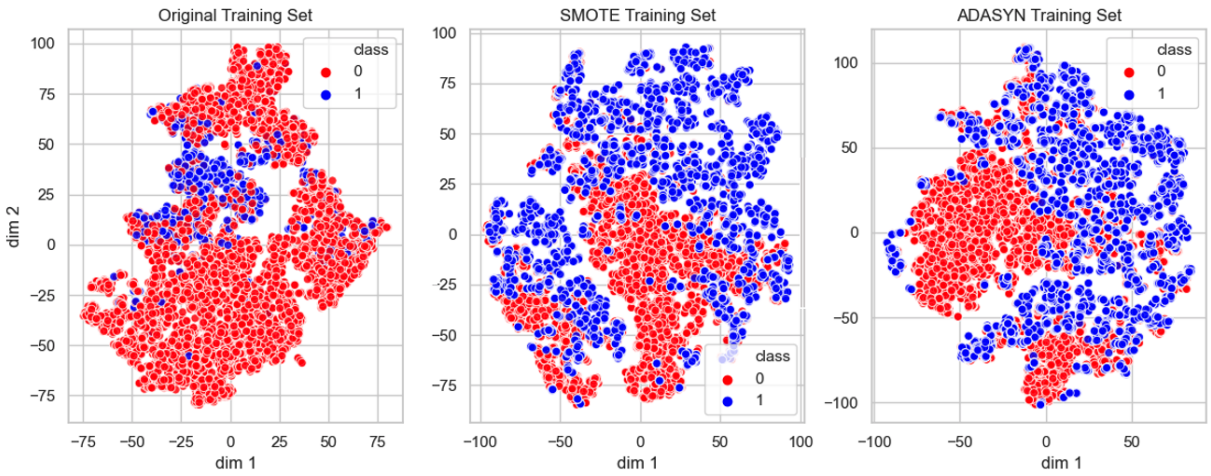


Figure 4.2: Oversampling techniques visualisation of the training set.

it can be seen that the three examples were of different nature. In the original training set, the positive class was at a high percentage, separable from the negative class with few exceptions. When SMOTE and ADASYN were applied to balance the training set, the overlap rate was a bit higher, which was expected due to the nature of our data. Finally, another observation, is the difference in shape between the SMOTE and ADASYN projections from Figure 4.2. This is due to the random behaviour of the t-SNE algorithm [59]. T-SNE has a cost function that is not convex, in which different parameter initialisation results to different results [59].

Even though SMOTE can reduce bias towards the majority class to some extent, it comes with several limitations. As stated by [145], SMOTE has some blindness in synthesising new instances, which could result in overlap between the two classes [146]. Meanwhile, ADASYN is based on the idea of adaptively generating minority data samples according to their distributions: more synthetic data are generated for minority class samples that are harder to learn compared to those that are easier to learn. The ADASYN method can not only reduce the learning bias introduced by the original imbalance data distribution but also adaptively shift the decision boundary to focus on those difficult samples. After creating the samples, ADASYN adds a random small bias to the points, making them not linearly correlated to their parents. Even though this is a small change, it increases the variance in the synthetic data [147].

Nevertheless, as the results showed, using ADASYN does not guarantee better classification

performance from SMOTE. In the next section, we investigate whether further improvements could be achieved in the classification when a hybrid approach is considered, meaning both undersampling and oversampling techniques are applied.

4.4.5 Experiment III: Results Using a Hybrid Data-Level Technique

There are a few drawbacks associated with the use of over- or undersampling techniques for tackling the class imbalance problem. When undersampling is implemented, potentially useful data could be discarded, whereas with oversampling, the size of the dataset is increased, adding computational cost. However, simultaneously, we could introduce instances that are mixed with the majority samples' data, similar to the situation in the previous section. In this experiment, we investigated a hybrid sampling approach called proportional oversampling, which tries to mitigates the drawbacks originating from oversampling and undersampling approaches [148].

Instead of trying to balance the existing dataset, we randomly selected individuals from the majority class (negative instances) at different proportions and oversampled instances of the minority class (positive instances) using either ADASYN or SMOTE until we achieved a perfectly balanced dataset. With a hybrid approach, a modest amount of oversampling can be applied to the minority class to improve the bias towards these examples whilst also applying a modest amount of undersampling to the majority class to reduce the bias on that class. We used random undersampling to reduce the majority class samples by proportion with ratios equal to 0.1–0.5 (undersampling until minority class was 10–50% from the majority class).

Figure 4.3 and Figure 4.4 show how the recall, precision, specificity and F1 score varied for each algorithm when the undersampling ratio changed. It was observed that the recall increased as the undersampling ratio increased, with the highest recall score achieved by XGBoost when combined with ADASYN for a score of 95%; however, at the same time, the precision reduced significantly. The best F1 score was achieved by XGBoost at a ratio of 0.3 with SMOTE (72.27%) and a ratio of 0.1 with ADASYN (72.53%). The final results were not conclusive as to whether the addition of undersampling improved the classification performance. Similar to Section 4.4.4, the average classification performance of the two best undersampling ratios of 0.3

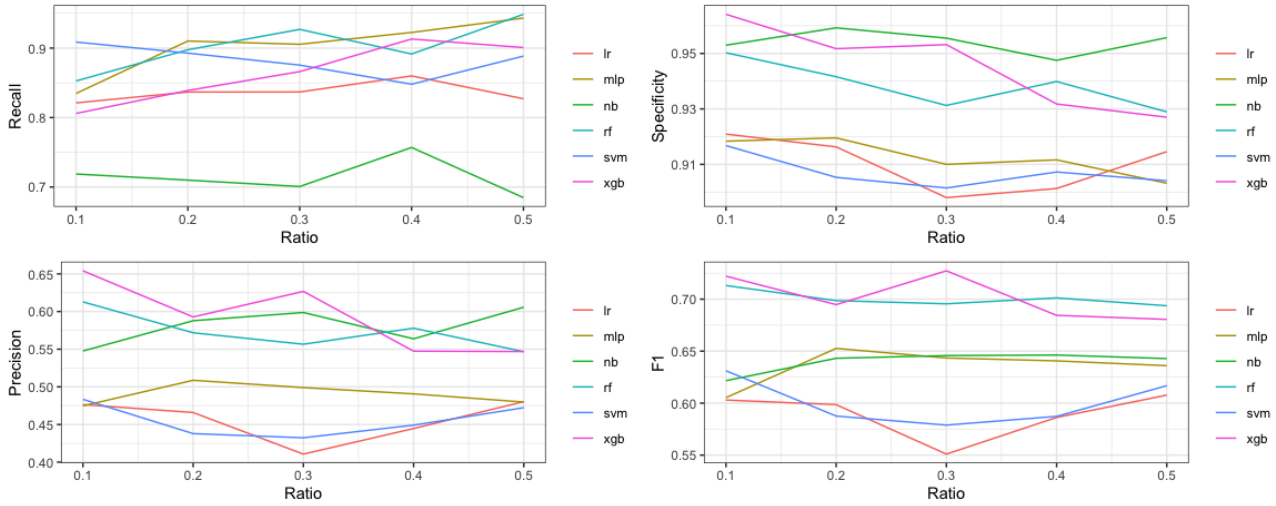


Figure 4.3: Results with different ratios of random undersampling and SMOTE. The recall increased as the undersampling ratio increased, while the precision and specificity decreased. The best overall F1 score was achieved by XGBoost and then RF when the undersampling ration was set to 0.3, meaning the minority class should be 30% of the majority class.

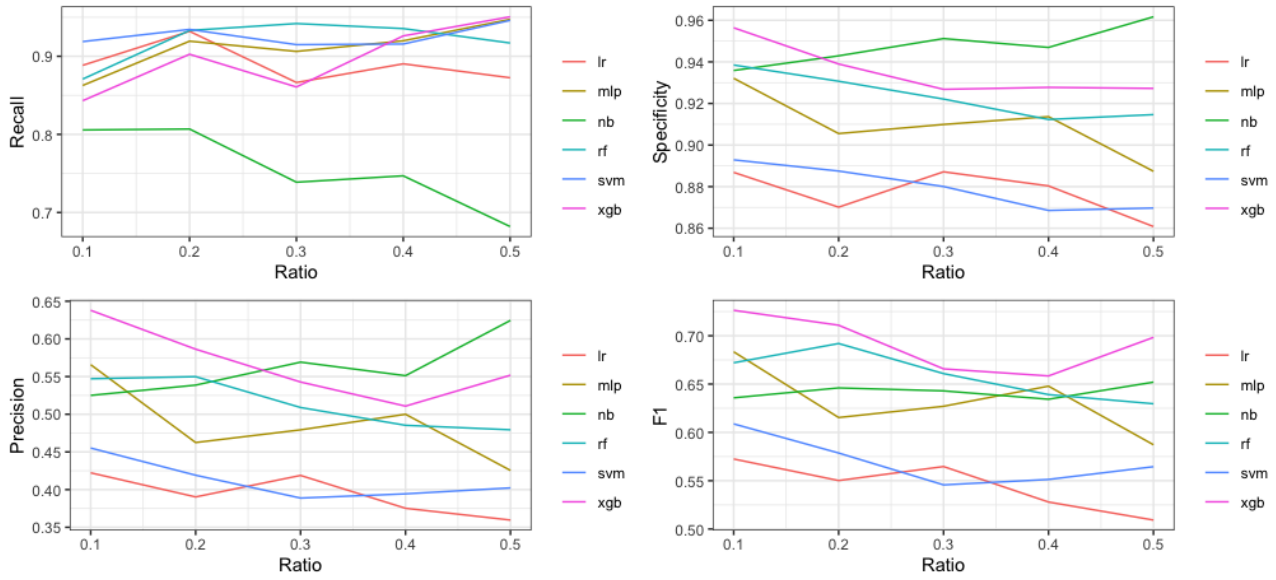


Figure 4.4: Results with different ratios of random undersampling and ADASYN. The recall increased as the undersampling ratio increased. Then, we removed more samples with random undersampling, and, simultaneously, the precision and specificity decreased. The best overall F1 score was achieved by XGBoost and RF when the undersampling ratio was set to 0.1, meaning the minority class should be 10% of the majority class before oversampling.

for SMOTE and 0.1 for ADASYN was investigated further in Table 4.7 and Table 4.8. When undersampling was used prior to oversampling, the algorithms could predict high-risk players with higher accuracy; however, there was a noticeable drop in the classification accuracy of the low-risk group (lower precision).

Table 4.7: Classification results ($mean \pm std$) when training set was balanced with ADASYN and undersampling with a ratio of 0.1: accuracy, recall, precision and F1.

	Performance Metrics				
Algorithms	Accuracy	Recall	Specificity	Precision	F1
LR	0.8960 ± 0.0022	0.9022 ± 0.0104	0.8954 ± 0.0022	0.4421 ± 0.0185	0.5932 ± 0.0165
RF	0.9346 ± 0.0037	0.8809 ± 0.0137	0.9395 ± 0.0040	0.5733 ± 0.0203	0.6944 ± 0.0168
XGB	0.9482 ± 0.0037	0.8184 ± 0.0122	0.9597 ± 0.0028	0.6429 ± 0.0207	$\mathbf{0.7200 \pm 0.0161}$
NB	0.9317 ± 0.0057	0.7768 ± 0.0161	0.9456 ± 0.0056	0.5619 ± 0.0287	0.6517 ± 0.0216
SVM	0.8945 ± 0.0016	0.9137 ± 0.0108	0.8929 ± 0.0021	0.4247 ± 0.0104	0.5798 ± 0.0096
ANN	0.9301 ± 0.0083	0.8580 ± 0.0250	0.9366 ± 0.0094	0.5562 ± 0.0373	0.6741 ± 0.0295

Table 4.8: Classification results ($mean \pm std$) when training set was balanced with SMOTE and a ratio of 0.3: accuracy, recall, precision and F1.

	Performance Metrics				
Algorithms	Accuracy	Recall	Specificity	Precision	F1
LR	0.9037 ± 0.0034	0.8482 ± 0.0239	0.9086 ± 0.0041	0.4524 ± 0.0245	0.5897 ± 0.0232
RF	0.9296 ± 0.0035	0.9243 ± 0.0156	0.9301 ± 0.0035	0.5553 ± 0.0192	0.6937 ± 0.0187
XGB	0.9386 ± 0.0036	0.8899 ± 0.0161	0.9428 ± 0.0042	0.5749 ± 0.0245	$\mathbf{0.6980 \pm 0.0165}$
NB	0.9326 ± 0.0043	0.7343 ± 0.0315	0.9504 ± 0.0058	0.5720 ± 0.0255	0.6425 ± 0.0194
SVM	0.8937 ± 0.0040	0.8980 ± 0.0168	0.8933 ± 0.0029	0.4284 ± 0.0165	0.5800 ± 0.0183
ANN	0.9276 ± 0.0069	0.8797 ± 0.0152	0.9317 ± 0.0082	0.5310 ± 0.0206	0.6619 ± 0.0149

It is also noteworthy that the machine learning techniques improved their recall score when ADASYN was used to balance the training data whilst their precision score decreased. This supported the initial assumption that ADASYN helps in the classification of the minority class. We could conclude that overall producing synthetic data can assist in the improvement of the classification performance in supervised learning involving the imbalance class problem.

In terms of examining the quality of data generated from SMOTE and ADASYN, investigation were carried out in Chapter 5 in section 5.7, where we examined the ability of synthetic data generation techniques to generating new synthetic instances and not replicating the existing ones. To achieve that the statistical test of Wilcoxon rank-sum [149] and Kolmogorov–Smirnov tests [150] were examined.

Table 4.9: The detection flags from the rules-based automated system were compared with the best performing machine learning models. Both RF and XGBoost outperformed the rule-based system in all performance indicators.

	Performance Metrics				
Algorithms	Accuracy	Recall	Specificity	Precision	F1
Rule-Based System	0.9148	0.8185	0.9234	0.4856	0.6096
RF + SMOTE	0.9429	0.8650	0.9503	0.6195	0.7217
XGB + SMOTE	0.9517	0.8107	0.9641	0.6668	**0.7314**

4.5 Rule-Based System Comparison

To assess the quality of the results produced by the machine learning algorithms in Section 4.4.5, we compared them with the detection flags, as these were generated by a rule-based system. As described in Section 3.1 a false positive by the automated system was *defined as a case wherein a player is flagged by the rule-based system but passes all appropriate money laundering checks and avoids an internal risk report.* Similarly, a false negative was *defined as a case wherein the AML process generates an internal risk report for a player who was not detected by the automated rule-based system.* To evaluate the effectiveness of the machine learning algorithms in fraud identification, we compared the results with the detection flags from the rule-based system provided by Kindred Group on the test set we used to evaluate the performance of our models. The performance indicators of the rule-based system are reported in Table 4.9 and were compared against the top two performing machine learning models from the previous sections.

Table 4.9 shows that the supervised learning framework with either XGBoost or RF outperformed the rule-based system in all categories, with better recall, precision, specificity and F1 score. The F1 score in both machine learning models showed an improvement of 13% for XGBoost and 12% for RF in comparison with the rule-based system. The results indicated that with a supervised learning framework, we could improve the identification rate for both high-risk (true positive rate) and low-risk players (true negative players).

4.6 Interpretability in Fraud Detection

Machine learning systems are now being used for automated decision-making in areas such as security, finance, autonomous vehicles, robotics and healthcare. However, the inner workings of state-of-the-art machine learning systems are frequently referred to as a black box. The size and complexity of learned model calculations are considered to be beyond the capacity of human understanding [151]. Whilst it is important to develop a consistent solution with strong prediction or classification abilities, increasingly, in sensitive business applications (e.g. where human safety is concerned or where decisions can materially impact an individual), it is interesting to determine how a model provides these results, both at the model level (global explanations) and sample level (local explanations) regarding, for instance, which variables are engaged the most, the presence of correlations and the possible causation relationships.

Techniques for explaining and interpreting machine learning models are evolving, with explainable artificial intelligence (XAI) being an emerging field within the machine learning community with considerable advances in state-of-the-art in recent years. Earlier, we discussed a new method called TREPAN [15] that is able to extract human-readable logic rules from a neural network trained to predict self-exclusion. In addition, it has been argued [152] that single predictions can be explained by counterfactuals stating the minimum changes needed for an observation to change its classification. An example of a counterfactual explanation is the following: you would have received a loan if your annual salary were US$50,000 instead of US$42,000. Therefore, two types of explainability exist, i.e. local (specific sample) and global (general explanation of the model). Since this research was at an experimental stage, we provide several insights at the global level which includes techniques such as weights from LR, RF feature importance and SHAP [153].

In this section, we focused on using SHAP global view to provide some interpretability to our machine learning results. The SHAP values were derived from the concepts of cooperative game theory and local explanations [153]. Given a set of players, cooperative game theory shows how well and fairly to distribute the payoff amongst all payers that are working in coordination. The analogy here is that players are equivalent to independent features, and the payoff is the

difference between the average prediction of the instance minus the average prediction of all instances. The SHAP values for each feature represent the changes in the expected model prediction when conditioning on that feature. For each feature, the SHAP values explain the contribution to explain the difference between the average model prediction and actual prediction of the instance. This has been used to provide both local and global explainability in machine learning models. Thus, we examined its ability to provide global explanations. Here, we preferred to use SHAP rather than the RF feature importance from Chapter 3. The main reasoning behind this choice was that permutation importance does not tell us how each feature matters. For example, if a feature has medium permutation importance, this could mean it has either a large effect on a few predictions but no effect in general or a medium effect for all predictions. On the contrary, a SHAP summary plot can provide a view of the feature importance and what is driving it. Figure 4.5 shows the SHAP summary plot produced on our best machine learning model. The results from Figure 4.5 and Figure 3.9 look similar, in terms of showing similar importance results (the order of importance is the same in both cases). Since, a SHAP summary plot is another way to describe the effect of each feature in the classification task, the similarity between the two Figures can provide validity in the importance scores.

We can observe from Figure 4.5 that the total deposit, deposit over the number of active days ratio, total bet and win amount and withdrawal deposit ratio had a significant impact on the model's classification ability. This could explain to some degree why the industry struggles to develop AML detection systems with high specificity or TN; the behaviours of high-value players, i.e. those who deposit the gamble typically in the 90_{th} percentile of all players, are difficult to distinguish from those TP cases. This is a sensitive area for the industry in that a large proportion of revenue is often driven by these players.

This suggests that the industry will be required to develop more sophisticated models that incorporate a wider range of data, including data not typically held within gambling operator systems, e.g. details about the players' occupations, incomes and wealth. These are sensitive types of information to ask players, but it is not inconceivable to expect gambling operators to one day have to collect data and implement affordability assessments on high-value players in the same manner as financial institutions do when making decisions on mortgages given that

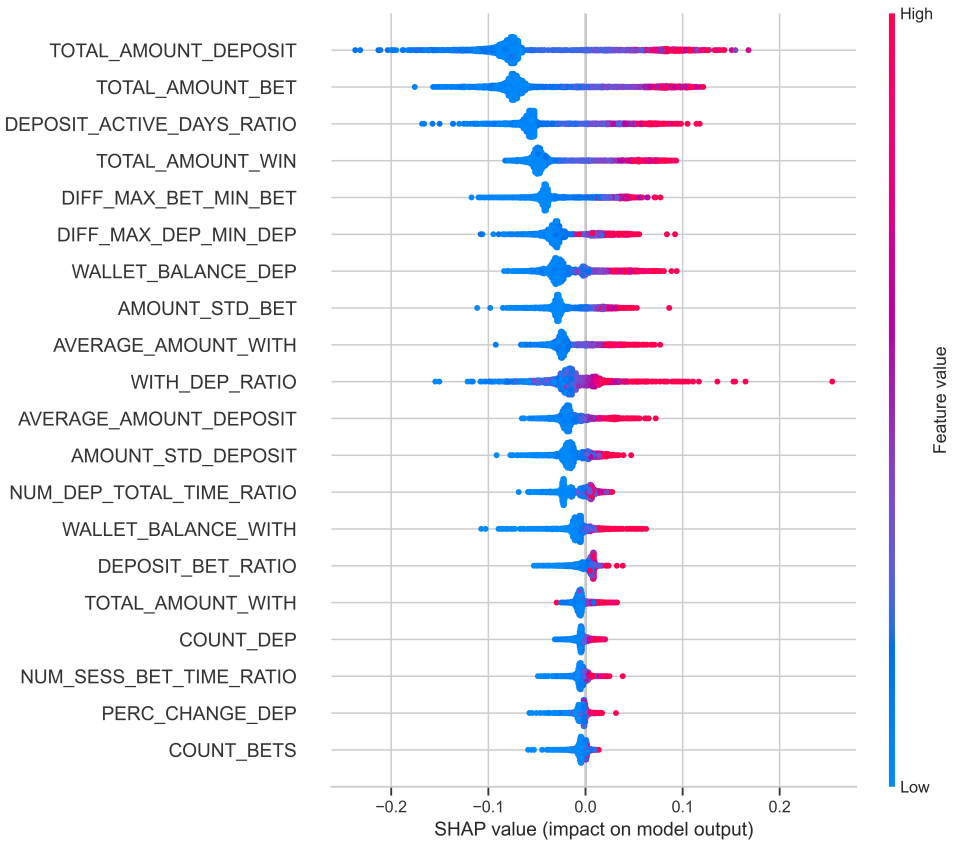


Figure 4.5: This is a SHAP summary plot. The vertical axis shows what feature is represented. The colour red or blue shows whether that feature was high or low, respectively, for that row on the dataset. Finally, the horizontal axis shows whether the effect of that value caused a higher or lower prediction.

the monthly expenditures can be similar in both cases. The United Kingdom (UK) regulator, the Gambling Commission, recently announced increased industry efforts to strengthen the oversight of schemes for high value players, i.e. VIP schemes, wherein such customers must pass thorough checks relating to spending, safer gambling and enhanced due diligence before becoming eligible for high-value customer incentives [154].

Model explainability also points us towards a strong overlap in suspicious player behaviour with behaviours associated with gambling related harm. Features such as AMOUNT_STD_DEPOSIT, and TOTAL_AMOUNT_BET and AVERAGE_AMOUNT_DEPOSIT, have been found in the past to be some of the most important features for predicting self-exclusion [14]. Indeed, all of the high-profile regulatory cases in the UK in recent years have highlighted both AML and problem gambling related behaviours [7]. For example, there is considerable research evidence to support that increasing deposits and staking can be associated with behaviours consistent

with problem gambling [155, 156, 157], which can include the following patterns: i) need for players to increase the amount of their wagers to achieve the desired excitement previously experienced at lower levels of wagering, ii) unsuccessful attempts to cut back or control gambling, iii) chasing wins or losses (i.e., wagering or betting more to try to win back what they have lost), and iv) suffering negative financial consequences (i.e. financial losses), that are likely to increase with higher wagering amounts in the long run.

In addition, several of the features developed in our model (Table 3.7), including fail transactions and the intensity of transactions, can also be associated with problematic play [12]. This points to the operational challenges and opportunities for the industry to harmonise and improve systems and processes to detect problematic play by combining human and analytical resources to assess both types of cases in parallel.

4.7 Summary

The growth of online gambling in the last decade has increased the risk of the industry becoming a channel for fraud, specifically money laundering. The development of an adaptive, real-time monitoring system that can be facilitated by self-learning models to detect money laundering and fraud is becoming one of the key priorities of gambling operators. In this chapter, we examined the ability of machine learning algorithms trained by a new set of features to find patterns related to suspicious money laundering cases.

A major limitation of anti-money laundering processes is that financial crime agencies provide limited or no feedback to operators regarding the money laundering cases they submit. Therefore, obtaining high quality ground-truth money laundering labelled data is impossible. Thus, building highly accurate profiles for money launderers is a difficult task. As mentioned earlier, IRRs are used as the ground truth to represent high-risk money laundering cases. However, many of these IRRs are the result of detections from the rule-based system, based on pre-defined thresholds, which could explain the most important features from Figure 4.5.

Furthermore, we should note that the data here were drawn from a single gaming platform. As

the volume of data is growing faster than ever, data protection laws are becoming increasingly tighter [7]. A challenge of this is that players and potential criminals are able to use different platforms from different operators. Even if they are flagged by some operators, they can still proceed to gamble using accounts with other operators. Data sharing between operators regarding flagged cases could significantly assist in building customer profiles.

The results obtained were compared with the current system of our research partner. The main priority during the experiment was to reduce the number of cases the system could not detect initially even though they were reported eventually through different monitoring elements. Therefore, improving the first level of monitoring is critical to reducing the money laundering risk.

Highly imbalanced data, undersampling and oversampling were evaluated together with the machine learning techniques. The XGBoost tree was the strongest performer with the highest overall F1. Regarding which model performed best in detecting money laundering, a trade-off existed between the true positive and true negative rates. If a model with the highest specificity were selected, this would mean we obtained less false positives, while if a model with higher recall were selected, then we would have less false negatives. This is something companies need to decide on.

Although considerable work has been accomplished in fraud detection, extensive empirical studies exploring the application of fraud detection algorithms to money laundering are few, and none have been published for gambling. This is a relatively new and a promising area for future study. The absence of high-quality labelled data in the industry makes this a challenging task. If national crime agencies are not prepared to share more data with the industry, regulators should consider progressing discussions on industry-wide data-sharing schemes. The need for more research is critical due to the increasing proliferation of crimes in online gambling, which affect millions of customers and threaten to drastically diminish trust in the existing gambling infrastructure, systems and platforms.

Finally, given the evidence of overlap between problem gambling and anti-money laundering cases, the industry should explore changes to its corporate culture, structure and processes

to enable greater sharing of data and experiences amongst departments, which may still be separate, especially the anti-money laundering and responsible gambling teams.

Chapter 5

Enhancing Classification in Fraud Detection with GANs

5.1 Introduction

In Chapter 4 we showed that supervised algorithms can accurately predict high-risk money laundering cases. However, more improvements could be applied to the overall identification rate and the quality of synthetic data. A direct approach to the data generation process would be the use of a generative model that captures the actual data distribution [91] for generating synthetic data. Generative adversarial networks (GAN) are a recent method that uses neural networks to create generative models [3]. As previous studies have shown, GANs can be used effectively as an oversampling method to produce high-quality synthetic data [147]. In contrast with other generative techniques, GANs are able to parallelise sample generation with sample classification. Further, they make no assumption about distribution and variational bounds. Finally, GANs make no use of Markov chain or maximum likelihood estimation [3, 158].

In this chapter, we propose a GAN-based approach called synthetic data generation GAN (SDG-GAN), which, as the empirical results show, can be a powerful tool for tackling the imbalanced class problem on structured data by generating new high-quality instances. In Section 5.2 an overview of GANs is provided, where in Section 5.3 we introduce our approach.

Our method is validated in Section 5.4 and 5.5 via experiments on benchmark datasets (Credit Card Fraud, Breast Cancer Wisconsin, Pima Diabetes) from different disciplines before applying it for generating new synthetic data for online gambling players in Section 5.6. Our method is evaluated in terms of its classification performance when combined with the classification models from Chapter 4, namely LR, RF, MLP and XGBoost.

5.2 Generative Adversarial Networks

GANs are generative models based on a game-theoretic scenario in which a generator (G) network is competing against a discriminator (D) [3]. The generator, with noise variable Z as input, generates fake samples with distribution p_g that match the true data distribution, p_{data}. However, the discriminator network is trained to distinguish the real samples (drawn from the training data) and fake samples generated from G.

A common analogy in the literature for GANs [159] is to think of one network as an art forger and another as an art expert. The forger, known in the literature as the generator, G, creates forgeries with the aim of making realistic images. The expert, known as the discriminator, D, receives both forgeries and real images and aims to tell them apart. Both are trained simultaneously and in competition with each other.

Typically, the discriminator model is trained to maximise its ability to distinguish real input

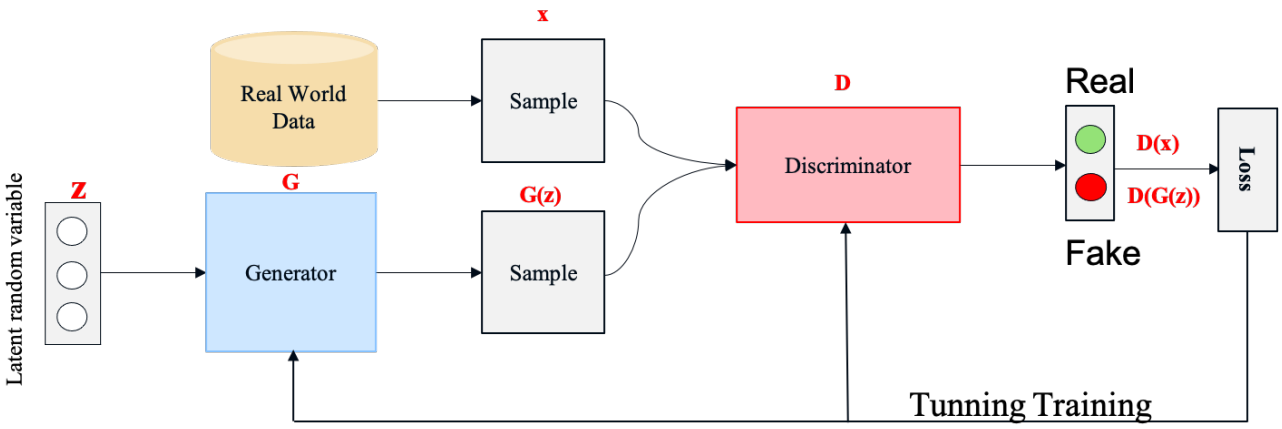


Figure 5.1: Generative Adversarial Network Architecture [3]

data from fake data. The generator tries to fool the discriminator by producing better fake samples. Mathematically, the generator and discriminator play a min-max two-player game with value function V(G,D) [3]:

$$\begin{aligned} \min_G \max_D V(G,D) = E_{x\sim p_{data}}[log(D(x))] \\ + E_{z\sim p_z}[1 - log(D(G(z)))] \end{aligned} \tag{5.1}$$

where E is the expectation, p_{data} is the real data distribution and p_z is a noise distribution. The training of a GAN could be characterised as an optimisation process for both the generator and discriminator. The output of the generator is defined as p_g. As equation (5.1) suggests, GANs aim to minimise the Jensen–Shannon divergence between the data distribution p_{data} and the generative distribution p_g with perfect minimisation reached when $p_g = p_{data}$. The optimisation equations for the generator and the discriminator are defined respectively as follows:

$$\min_G E_{z\sim p_z}[1 - log(D(G(z)))] \tag{5.2}$$

$$\begin{aligned} \max_D E_{x\sim p_{data}}[log(D(x))] \\ + E_{z\sim p_z}[1 - log(D(G(z)))] \end{aligned} \tag{5.3}$$

Although GANs are very promising for synthetic data generation, training a GAN is challenging [3] and often unstable which could lead to the following 'symptoms' [160, 161, 162]:

- Difficulties in making both the discriminator and generator converge [160].
- Collapse of the generator model by producing similar samples from different inputs [161].
- The discriminator converging quickly to zero [162], providing no reliable path for gradient updates to the generator.

Researchers have considered several approaches to overcome such issues. They have been experimenting with architectural changes [161], different loss functions [163] and both. Our SDG-GAN tries to eliminate the above problems by combining a different loss function and

architecture compared to the original vanilla GAN [3]. In Section 5.2.1, we give an overview of conditional GANs (cGANs) [164] and in Section 5.3, we explain how they were used to build the SDG-GAN architecture.

5.2.1 Conditional GANs

Conditional GANs [165] are a simple extension of the original GAN framework, which conditions the generator on class labels to generate output for a specific class [164]. The conditioning is achieved by feeding the class label y into both the discriminator and generator as additional input. Thus, the generator estimates the distribution of $p_{X|y}$, and the discriminator learns to estimate $D(X, y) = P(fake|X, y)$. The modification of the generator and discriminator with the conditional rule allows for the generation of samples belonging to a specific class. Furthermore, the conditional discriminator ensures that the generator does not ignore class labels [164]. Formally, the objective value function between generator G and discriminator D is the min-max in equation (5.4).

$$\begin{aligned} \min_G \max_D V(G, D) = E_{x \sim p_{data}}[log(D(x|y))] \\ + E_{z \sim p_z}[1 - log(D(G(z|y)))] \end{aligned} \tag{5.4}$$

During cGAN training, the discriminator is trained first with batches of only real features, $y_{real} = 1$ and then with batches of only fake ones, $y_{fake} = 0$ before the generator training continues through the GAN model. The GAN model assumes that the generated features will always be real, $y_{gan} = 1$. As cGAN is an extension of the original generative adversarial framework, it exhibits the same problematic behavior, i.e. mode collapse and unstable training, due to the vanishing gradient problem [166].

5.3 Synthetic Data Generation GAN

In the SDG-GAN framework the generator and discriminator of SDG-GAN are both feedforward networks with a MLP architecture. The generator of a regular GAN aims to generate fake data that are close to the real distribution. The discriminator of a regular GAN is used to identify whether an input is real or fake from the generator.

The process of generating new instances of the minority class requires training the GAN to estimate the distribution of the data. When the training phase is completed, new synthetic data can be generated utilising the generator's abilities. The cGAN architecture of estimating the conditional distribution, $p_{x|y}$, is adapted in our method to generate the minority class samples. Instead of regular loss, feature matching loss is adapted by the SDG-GAN. Feature matching loss was introduced by [161] as a method for improving GAN training.

Here, we propose a GAN architecture based on cGANs. The generator is a feedforward neural network that tries to learn the actual data distribution. In contrast with a cGAN generator, we use a feature matching technique to train the generator. Feature matching changes the cost function for the generator to minimise the statistical differences between the features of the real data and generated data. This changes the scope of the generative network from fooling the opponent to matching features in the real data. The objective function of feature matching loss is defined as follows:

$$||E_{x\sim p_{data}} f(x) - E_{z\sim p_z(z)} f(G(z))||_2^2 \tag{5.5}$$

where $f(x)$ is the feature vector extracted by an intermediate layer in the discriminator. Feature matching addresses the instability of GANs by specifying a new objective for the generator that prevents it from overtraining. Instead of directly maximising the output of the discriminator, the new objective requires the generator to generate data that match the statistics of the real data, while we use the discriminator only to specify the statistics we think are worth matching. Specifically, we train the generator to match the expected value of the features on

an intermediate layer of the discriminator. This is a natural choice of statistics for the generator to match because by training the discriminator, we ask it to find the features that are most discriminative of real data versus data generated by the current model [161].

To oversample an imbalanced dataset, we first trained the SDG-GAN's generator with imbalanced samples to estimate the data distribution. Once the training was completed, we could oversample the data by specifying to the generator how many new synthetic instances of the minority class we wanted to produce. We used a cGAN structure to estimate the conditional distribution, $p_{X|y}$, which allowed us to sample the minority class explicitly by conditioning the generator on the minority class label, $X_{new} = G(z, y = y^{minority})$.

The discriminator was trained similarly to a regular GAN discriminator. As with regular cGAN training, the objective had a fixed point where G exactly matched the distribution of the training data. We had no guarantee of reaching this fixed point in practice, but our empirical results indicated that feature matching is indeed effective in situations wherein a regular GAN becomes unstable [161]. Thus, we achieve the following objective function:

$$\min_G \max_D \underbrace{||E_{x\sim p_{data}} f(x|y) - E_{z\sim p_z(z|y)} f(G(z))||_2^2}_{FM_{Loss}} + E_{x\sim p_{data}}[log(D(x|y))] \tag{5.6}$$

where FM is the feature matching loss and the rest of the objective function is the binary cross entropy between true class label $y \in (0, 1)$ and the predicted class probability.

5.3.1 Hyperparameter Settings

Our proposed method has many hyperparameters that need to be tuned in order to achieve optimal performance. After experimenting and trying different hyperparameters, the hyperparameters in Table 5.1 have been chosen since they produced the best results. Future work could include optimising those hyperparameters for the oversampling task. The noise parameter distribution was set to be a Gaussian distribution with size dimensions set to 50. The dropout ratio was set to 0.2 on both discriminator's and generator's hidden layers. Batch size is 64 and

number of epochs was set to 100. In terms of activation function rectified linear unit (ReLU) was used for the hidden layers where sigmoid for the output layer of discriminator and tanh for the ouptut layer of the generator. Adam optimiser was selected for the training [167].

5.4 Experimental Design

To evaluate SDG-GAN as an oversampling method to tackle binary classification problems in imbalanced data, we compared the performance of the classification algorithms when combined with SDG-GAN and other state-of-the-art oversampling methods, e.g. SMOTE [61], ADASYN [62] and B-SMOTE [63], and other GAN-based oversampling architectures, e.g. cGAN.

In Section 5.4.1, we introduce the publicly available datasets used as part of the evaluation process. In Section 5.6, we apply our method to the real-world gambling dataset we produced in Chapter 3 and examine whether it improves the classification performance of the machine learning algorithms. A quality assessment of the new synthetic data is performed in Section 5.7 with the help of Kolmogorov's and Wilcoxon's data replication tests. The following hypotheses need to be met to describe our method as successful:

- H_0: The use of SDG-GAN will create completely new synthetic data without replicating existing data, i.e. it will learn to generate new data based on learning the distributions.
- H_1: The use of SDG-GAN to augment imbalanced datasets will improve the algorithmic performance in baseline experiments on the benchmark imbalanced datasets.

Table 5.1: SDG-GAN hyperparameters settings

Hyperparameter	**Value**
Learning Rate	$1x10^{-4}$
Optimiser	$Adam(a = 5x10^{-4}, \beta_g = 0, \beta_g = 0)$
Epochs	100
Batch size	64
Generator layers sizes	(Noise, 128), (128,64), (64, data size)
Discriminator Layer sizes	(data size, 128), (128,64), (64,32), (32, 1)
Activation function in hidden layers	Leaky ReLU
Noise Distribution p_z	N(0,1)
Noise size	50

- H_2: The use of SDG-GAN will improve the algorithmic performance of classification algorithms in the real-world gambling dataset.

For H_0, Kolmogorov and Wilcoxon are used to calculate the degree of similarity between the new synthetic and original data. Then, H_2 and H_3 are tested by combining the original and synthetic datasets with the four classification algorithms, i.e. LR, RF, XGBoost and MLP, in Section 5.5.

5.4.1 Benchmark Datasets

We evaluated our method on the different benchmark imbalanced datasets presented in Table 5.2. The IR was defined as the imbalance ratio between the minority and majority classes. We used data from different sectors to examine the range of applications for our method. We selected the Credit Card Fraud Dataset from Kaggle [168] and the Pima Diabetes [169] and Breast Cancer Wisconsin (Diagnostic) Datasets from the UCI Machine Learning Repository [170], an online resource containing several datasets for machine learning purposes.

The rationale behind using the benchmark datasets was so that the results of this study could be easily compared to similar studies carried out previously and in the future. Moreover, it was decided that all datasets should describe a binary classification problem and contain numeric features to be in the same format as our gambling data.

The Credit Card Fraud Dataset contains transactions made by credit cards in September 2013 by European cardholders. It presents transactions that occurred over two days, with 492 frauds out of the 2,492 transactions. The Wisconsin Breast Cancer Dataset includes features computed from a digitised image of a fine needle aspirate of a breast mass. The features describe characteristics of the cell nuclei present in the image. The purpose of the dataset is to classify a diagnosis as positive or negative. The PIMA Diabetes Dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. Its objective is to diagnostically predict whether a patient has diabetes based on certain diagnostic measurements included in

the dataset. Several constraints were placed on the selection of these instances from a larger database; in particular, all patients are females at least 21 years old of Pima Indian heritage.

Note that before these datasets were used, their attribute values were scaled to be in interval $[0, 1]$ by the min-max method to make the range of all attributes the same, preventing any one of them from dominating the others due to its scale. This reduced the range of values that the generator had to produce as well. With regard to implementation, all standard oversampling method tests were implemented using the '*imblearn.over_sampling*' module in Python, similar to Chapter 4. We used the default hyperparameter settings for SMOTE and its variants, i.e. $kneighbours = 5$. For cGAN, we primarily used the hyperparameter settings that we set for SDG-GAN method, as seen in Table 5.1.

5.5 Results

For each dataset, we present the classification results observed after 10 runs for each oversampling technique and classification algorithm. The results in this section represent the average scores during those 10 runs. Similar to the process of [147], we split the data into testing and training sets. The training set included 80% of the total population of the samples of each class and the testing set the other 20% of the data. The data were shuffled to ensure reliable distribution in the sets.

In the SDG-GAN, given an imbalanced training dataset, we first calculated the imbalance difference between the classes in the dataset. Then, a set of noise vectors with a dimension of 50 was used as the input for the generator. We trained the network generator by optimising

Table 5.2: UCI datasets used in this thesis. There are three different sectors (B = business, L= life sciences). Number of features, number of instances, imbalance ratio

ID	Data Set	Sector	#Features	#Instances	IR
1	Credit Card Fraud	B	30	2,492	1:4.07
2	PIMA Diabetes	L	8	768	1:1.87
3	Breast Cancer	L	30	569	1:1.68

Table 5.3: Real-world Gambling Dataset

ID	Data Set	Sector	#Features	#Instances	IR
1	Gambling Fraud	B	31	4,700	1:2.97

the generator using the loss equation (5.5). Real and synthetic data were then used as input for the discriminator D to output a probability value for evaluating the authenticity of the input data. Finally, the simulation samples generated through SDG-GAN were bonded with the original samples to enhance and balance the training dataset, which was then fed into the machine learning model for training.

5.5.1 Results of Benchmark Datasets

Table 5.4, Table 5.5 and Table 5.6 show the results observed for the three imbalanced public datasets of Credit Card Fraud, Breast Cancer and Pima Diabetes. We compare the five oversampling techniques in combination with four classification algorithms from Chapter 4. The performance of each classification method is measured in terms of recall, precision and F1 score.

For the Credit Card Fraud Dataset in Table 5.4, the highest F1 score was achieved when SDG-GAN was combined with RF for a score of 91.31%. In Table 5.5 for the Breast Cancer Dataset, cGAN combined with XGBoost outperformed the rest of the methods with F1 score of 91.95%. Similarly with the Credit Card, in the Pima Diabetes Dataset, SDG-GAN in combination with RF produced the best results with an F1 score of 70.80% as Table 5.6 indicates. This was a significant improvement of $\approx 5\%$ compared to when no oversampling was used and an improvement of $\approx 2\%$ than the second-best combination between MLP and ADASYN. Another observation that could be drawn from the results was that when the standard oversampling techniques were used i.e. SMOTE, ADASYN, there was a drastic improvement in the classification of the minority class with better overall recall compared to precision (in the majority of cases). However, simultaneously, there was a huge drop in the classification accuracy of the majority class. This was supported by the increase of the recall score in the Credit Card Fraud Dataset prior to the use of any oversampling method; on average, the recall was 85% and the

precision 94%. When SMOTE was used, the recall score increased significantly, while the precision decreased. However, this was not the case when SDG-GAN was used for oversampling, whereby we saw a more robust improvement in the classification metrics, as Table 5.4 and Table 5.6 show.

The mean rankings of the F1 score per classifier across all datasets are presented in Table 5.7. No one oversampling technique performed best across all classification methods and datasets. However, the SDG-GAN method performed consistently well and managed to achieve the highest overall mean rank score (2.6).

Among the oversampling methods, SMOTE produced the second-best results, outperforming ADASYN and B-SMOTE. This indicated that the more recent variations of SMOTE do not

Table 5.4: Credit Card detection results: recall, precision and F1 measure

Algorithms	**Metrics**	**W/O**	**SMOTE**	**ADASYN**	**B-SMOTE**	**cGAN**	**SDG-GAN**
LR	Recall	0.8142	0.8571	0.8667	0.8495	0.8144	0.8090
	Precision	**1.0000**	0.9545	0.7865	0.9320	0.9875	0.9863
	F1	0.8975	0.9032	0.8246	0.8888	0.8926	0.8889
RF	Recall	0.8984	0.8694	0.9288	0.8894	0.8453	0.9208
	Precision	0.9170	0.9586	0.8309	0.9106	0.9647	0.9055
	F1	0.9076	0.9116	0.8771	0.8999	0.9010	**0.9131**
XGB	Recall	0.8973	0.9163	0.9087	0.8776	0.8559	0.9053
	Precision	0.9112	0.8959	0.8787	0.8600	0.9694	0.9122
	F1	0.9042	0.9060	0.8935	0.8687	0.9091	0.9087
MLP	Recall	0.8191	0.8830	0.9087	0.8761	0.8454	**0.9487**
	Precision	0.9390	0.8384	0.8536	0.9082	0.9879	0.8315
	F1	0.8750	0.8601	0.8803	0.8919	0.9111	0.8862

Table 5.5: Breast Cancer detection results: recall, precision and F1 measure

Algorithm	**Metrics**	**W/O**	**SMOTE**	**ADASYN**	**B-SMOTE**	**cGAN**	**SDG-GAN**
LR	Recall	0.8095	0.8888	0.9048	0.9302	0.9067	0.8837
	Precision	0.9189	0.9302	0.8636	0.8888	0.8863	0.9500
	F1	0.8608	0.9091	0.8837	0.9090	0.8966	0.9157
RF	Recall	0.8604	0.9069	0.8666	0.9069	0.8604	0.8809
	Precision	0.8809	0.8478	0.9069	0.8667	0.9024	0.8604
	F1	0.8706	0.8764	0.8863	0.8863	0.8809	0.8706
XGB	Recall	0.8524	0.9262	0.9143	0.9119	0.9524	0.8571
	Precision	0.8802	0.8282	0.8426	0.8567	0.8889	0.9767
	F1	0.8637	0.8742	0.8754	0.8821	**0.9195**	0.9130
MLP	Recall	0.8604	0.9069	**0.9381**	0.9302	0.9069	0.8604
	Precision	**0.9487**	0.8863	0.7940	0.8888	0.8863	0.9737
	F1	0.9024	0.8965	0.8578	0.9090	0.8965	0.9137

Table 5.6: Pima Diabetes Dataset results: recall, precision and F1 measure

Algorithm	Metrics	W/O	SMOTE	ADASYN	B-SMOTE	cGAN	SDG-GAN
LR	Recall	0.6181	0.7455	0.7272	0.7091	0.6727	0.6727
	Precision	**0.6939**	0.5694	0.5479	0.5652	0.6851	0.6379
	F1	0.6538	0.6456	0.6250	0.6290	0.6788	0.6549
RF	Recall	0.6727	0.7636	0.7454	0.6727	0.6545	0.7272
	Precision	0.6379	0.6086	0.5775	0.6271	0.6101	0.6897
	F1	0.6548	0.6774	0.6507	0.6491	0.6315	**0.7080**
XGB	Recall	0.6727	0.7091	0.7636	0.7455	0.6727	0.6545
	Precision	0.5781	0.6094	0.5753	0.5775	0.6066	0.5902
	F1	0.6218	0.6555	0.6562	0.6508	0.6379	0.6207
MLP	Recall	0.6182	0.7818	**0.7818**	0.7091	0.6727	0.6727
	Precision	0.6938	0.6142	0.5890	0.6094	0.6491	0.6852
	F1	0.6538	0.6880	0.6718	0.6555	0.6607	0.6788

necessarily outperform their predecessor, mirroring previous findings in the credit scoring literature [164]. Considering the mean ranking results from Table 5.7, we could address the second hypothesis, H_1, stating that the use of SDG-GAN improves the classification performance in experiments on benchmark imbalanced datasets.

Table 5.7: Summary Rank Results For F1 score

	Overall	Classifier			
Method	Mean Rank	Logistic Regression	Random Forest	XGBoost	Multilayer Perceptron
SDG-GAN	**2.6**	2.7	2.3	3.3	2.0
W/O	4.3	3.3	4.0	5.0	4.7
SMOTE	2.9	2.0	2.7	3.3	3.7
B-SMOTE	3.6	4.0	3.7	3.7	3.0
ADASYN	4.3	5.3	4.0	3.7	4.3
cGAN	3.1	2.7	4.3	2.0	4.0

5.6 SDG-GAN in Online Gambling

After the success of our proposed method on the benchmark datasets, we applied our SDG-GAN technique for generating synthetic players' data to tackle the imbalanced class in the gambling dataset from Chapter 3. In contrast with the experiments from Chapter 4, wherein we included around 15,000 samples in our player distribution, we decided to further reduce the noise in the dataset and discard any individuals who deposited very small amounts in total

Table 5.8: Gambling Dataset results

Algorithm	Metrics	W/O	SMOTE	ADASYN	B-SMOTE	cGAN	SDG-GAN
LR	Recall	0.6842	0.8907	0.8745	0.9109	0.6541	0.7206
	Precision	0.8942	0.8209	0.8120	0.7840	0.8788	0.8900
	F1	0.7752	0.8544	0.8421	0.8427	0.7500	0.7964
RF	Recall	0.9245	0.9338	0.9249	0.9367	0.9245	0.9004
	Precision	0.8546	0.8389	0.8328	0.8223	0.8556	**0.8943**
	F1	0.8881	0.8838	0.8764	0.8761	0.8887	**0.8973**
XGB	Recall	0.9195	0.9449	0.9492	0.9576	0.8923	0.9322
	Precision	0.8645	0.8479	0.8327	0.8278	0.8722	0.8627
	F1	0.8912	0.8938	0.8871	0.8880	0.8821	0.8961
MLP	Recall	0.8189	0.9671	0.9588	**0.9712**	0.8213	0.8601
	Precision	0.8805	0.7655	0.7767	0.7540	0.8816	0.8636
	F1	0.8486	0.8545	0.8582	0.8489	0.8807	0.8619

(<£500). The reason of the chosen £500 threshold, was decided after communicating with our industrial partners. They suggested that players who have not exceeded a £500 deposit threshold, have an insignificant money laundering risk. The final metrics of our experimental dataset are available in Table 5.3. Since we used a subset of the dataset from Chapter 4, the performance metrics of the rule-based system (Table 4.9) based on the test set changed and needed to be re-calculated , with the new F1 score equal to 84.76% on average (from the 10 runs).

We used SDG-GAN as part of the supervised learning framework from Figure 4.1 for oversampling the minority AML class. Similar to the benchmark dataset case experiments, we evaluated the effectiveness of our approach for practical applications against the standard oversampling techniques and a GAN-based approach introduced in this chapter. Table 5.8 presents the classification performance results for the gambling fraud dataset.

Similar to the experiments on the Credit Card Fraud and Diabetes Datasets, the performance of SDG-GAN was superior, with the highest F1 measure and precision at 89.73% and 89.43%, respectively. As Table 5.8 shows, SDG-GAN combined with XGBoost and RF outperformed the other oversampling techniques. However, when combined with LR, we did not expect it to improve the classification performance. Overall, the SDG-GAN results showed it can effectively estimate even complex data distributions. Furthermore, the results from Table 5.8 supported the final hypothesis, H_2, stating that the use of SDG-GAN could improve the identification

rate of risk of money laundering in online gambling. Comparing the new classification results with SDG-GAN and the rule-based system, there was a significant F1 score improvement of around 5%. Overall, with our oversampling method, we managed to reduce the number of both false positives and false negatives compared to the other techniques and enhanced the ability of the classification algorithm to distinguish the AML and Normal groups' classes.

5.7 Assessment of Synthetic Data Quality

To evaluate the quality of the synthetic data, statistical replication tests were performed. In general, GANs can be over-trained [161], which could potentially lead to replication of the original data. To ensure we avoided this issue, we examined whether SDG-GAN was learning the original data distribution rather than just memorising it. Following the approach suggested by [171] we calculate the euclidean distance between the new synthetic data and its nearest neighbour in the training and test sets. Subsequently, the distance distribution was compared via the statistical Wilcoxon rank-sum [149] and Kolmogorov–Smirnov tests [150].

Both tests are non-parametric significance tests for determining whether two independent samples are drawn from the same population with the same statistical distribution. Both output a p-value that, if high, will signify that two samples are drawn from the same population, e.g. that SDG-GAN replicates the training data instead of creating new data. A low p-value will prove that the models are generating unique data. Regarding the implementation, all statistical tests were carried out using the 'scipy.stats' module in Python.

The null hypothesis of the Wilcoxon rank-sum test was defined as that the data from the two distributions were the same, indicating that the SDG-GAN technique was replicating the original data rather than creating new data. The alternative hypothesis was defined as a notable difference between the two distributions, concluding that new data were being created. In the Kolmogorov–Smirnov test, the null distribution was calculated under the null hypothesis that the samples were drawn from the same distribution. On the other hand, the alternative hypothesis was that there was a big difference between the two distributions, which means, the

Table 5.9: Statistical test results for all the methods and datasets. Small p-values indicate the rejection of the null hypothesis that the new data replicate the original data.

Dataset	Method	Wilcoxon Statistic	Wilcoxon P-Value	K-S Statistic	K-S P-Value
Breast Cancer	ADASYN	814	$1.47e^{-19}$	0.5777	$2.73e^{-23}$
	SMOTE	436	$1.62e^{-22}$	0.5839	$7.33e^{-24}$
	cGAN	1006	$3.87e^{-18}$	0.5772	$2.74e^{-23}$
	B-SMOTE	1001	$3.56e^{-18}$	0.772	$2.75e^{-23}$
	SDG-GAN	915	$8.38e^{-19}$	0.5772	$2.75e^{-23}$
Diabetes	ADASYN	132	$6.79e^{-36}$	0.7465	$3.77e^{-58}$
	SMOTE	133	$6.89e^{-36}$	0.7605	$1.07e^{-60}$
	cGAN	612	$4.83e^{-33}$	0.7417	$2.54e^{-57}$
	B-SMOTE	612	$4.83e^{-33}$	0.7417	$2.54e^{-57}$
	SDG-GAN	677	$1.15e^{-32}$	0.7417	$2.54e^{-57}$
Credit Card	ADASYN	1045	$6.35e^{-64}$	0.7725	$5.82e^{-118}$
	SMOTE	1072	$6.47e^{-65}$	0.9500	$3.97e^{-200}$
	cGAN	1659	$5.49e^{-62}$	0.7725	$5.82e^{-118}$
	B-SMOTE	1631	$4.48e^{-62}$	0.7725	$5.82e^{-118}$
	SDG-GAN	1988	$5.82e^{-61}$	0.7725	$5.82e^{-118}$
Gambling Data	ADASYN	8500	$5.29e^{-155}$	0.7444	$4.07e^{-271}$
	SMOTE	4643	$6.88e^{-160}$	0.7490	$2.64e^{-275}$
	CGAN	15168	$9.52e^{-147}$	0.7440	$2.78e^{-271}$
	B-SMOTE	15582	$3.04e^{-146}$	0.7430	$2.76e^{-270}$
	SDG-GAN	15094	$7.73e^{-147}$	0.7430	$2.76e^{-270}$

oversampling techniques were producing new synthetic data. The outcome of the statistical tests for all the datasets are summarised in Table 5.9.

All p-values of the Wilcoxon rank-sum test from Table 5.9 were small (< 0.05). That enables us to reject the null hypothesis of the Wilcoxon Rank Sum test, and not reject the alternative hypothesis. Consequently this means that the new generated data from SDG-GAN, as well from SMOTE, B-SMOTE, ADASYN and cGAN are new synthetic data and not a replication of the training data. In addition, the p-value for every Kolmogorov–Smirnov test, is very small again (< 0.05). Similarly to the first statistical test, this allows us to reject the null hypothesis, and not reject the alternative hypothesis that SDG-GAN is generating new synthetic data rather than replicating training data.

Now, we can address the first hypothesis H_0, and summarised that generative adversarial models are learning new representations instead of replicating training data. The experimental results suggested that GANs are capable of generating new, unique data that is similar to input data

for various numeric datasets. They also showed that in certain circumstances the synthetic data can be used to augment imbalanced datasets and improve algorithmic performance.

5.8 Summary

In this chapter, we introduced SDG-GAN, an architecture based on GANs for generating synthetic data. Our method was compared against popular oversampling techniques i.e. SMOTE, B-SMOTE and ADASYN as well as other adversarial network architecture that has been used for generating new data i.e. cGANs. We evaluated the ability of SDG-GAN to produce high-quality synthetic data by comparing the algorithmic performance of four machine learning classification algorithms when combined with our method on three public imbalanced datasets and a real-world gambling fraud dataset. We found that the SDG-GAN oversampling compared favourably to the other oversampling methods and achieved the highest overall rank, as Table 5.7 shows. Our method outperformed SMOTE, ADASYN, B-SMOTE and cGAN on three out of the four examined imbalanced datasets, with the best performance achieved when it was combined with RF in two out of the three experiments.

In the real-world gambling dataset, the application of SDG-GAN helped improve the identification rate by improving the F1 score by 5% compared to the rule-based system and around 0.4% compared to the other oversampling techniques. Finally, to examine the quality of the new generated synthetic data, the statistical tests of Wilcoxon rank-sum and Kolmogorov–Smirnov were carried out. The results of the tests confirmed that there was no replication in newly generated synthetic datasets.

Chapter 6

Semi-Supervised GANs for Fraud Detection

6.1 Introduction

In the previous two chapters we showed that supervised learning could be effective in identifying high-risk for money laundering players in online gambling. At first in Chapter 4 we examined different discriminative classification algorithms together with oversampling techniques in order to identify the high-risk players. The supervised learning techniques showed promising results. Then in Chapter 5 we tried to improve further the results from Chapter 4 by improving the synthetic data generation module of the supervised learning framework Figure 4.1. To achieve that we introduced SDG-GAN, a new architecture based on GANs for synthetic data generation. Our method was compared with the best classification methods from 4, and the empirically results showed improvement in the classification performance of the machine learning algorithms.

As stated before in this thesis, in fraud detection problems, the fraudulent cases tend to be far fewer than the non-fraudulent ones (an issue referred to in the literature as an 'imbalanced dataset'), which leads to difficulties in the training of classification algorithms. In most cases, such classification algorithms seek to maximise accuracy and as a result they become biased

towards the majority class.

Classification models, such as LR, RF, MLP, are typically discriminative models, i.e. via the use of a certain feature set, they try to select the most appropriate class. This is, essentially, the root cause of the problem of the bias caused by the data imbalance, as the algorithm does not have a notion of 'how' the data are produced, yet it focuses on the objective measure of discrimination (e.g. accuracy). A way of alleviating this problem is to use models that aim to also understand the underlying generative process, as done for example by generative networks. Gaussian Mixture Models (GMMs) have formed the backbone of a variety of generative models, including Hidden Markov Models, employed with this objective [172], yet they come with Gaussian distribution assumptions and require much effort to be deployed in classification problems. Such models have been used together with clustering techniques to provide the required classification algorithm [173].

GANs allowed for a more generic approach with the advantages of combining end-to-end both generative and discriminative techniques. As we showed with SDG-GAN, they can be a powerful tool in the imbalanced class problem at the data level by generating high quality synthetic data. By extending the traditional framework of GANs to allow for the discriminator to perform classification [161], semi-supervised GANs (SSGANs) have shown potential in the recent literature particularly at learning from unstructured data such as images or sound [174]. Nevertheless, research regarding the application of GANs to structured data has been very limited. It has been shown that an SSGAN generalises from a small number of training examples much better than a comparable, fully-supervised classifier [108, 175]. In this chapter we examine whether SSGANs can be used at the algorithmic level in order to improve the classification of imbalanced datasets.

We argue that SSGANs can provide a powerful and versatile framework for tackling supervised learning from imbalanced and sparse structured data. We validate this claim empirically by applying SSGANs to different domains suffering from the same data imbalance difficulty. We conduct experiments on the benchmark datasets of Credit Card Fraud, Breast Cancer Wisconsin and Pima Diabetes introduced in Chapter 5. Finally, similar to our experimental approach of

Chapter 5, we apply the proposed semi-supervised framework on the Gambling Fraud Detection dataset which is related with money laundering. We compare our results with those of classical discriminative techniques, namely RF, XGBoost, LR and MLP, trained in conjunction with SMOTE [61] and ADASYN [62]. The results show that our framework outperforms the other models even when these are combined with elaborate oversampling methods.

More specifically, in this chapter we introduce a system architecture based on semi-supervised generative adversarial networks and sparse auto-encoders (SAE) and we apply it to a fraud detection system and other classification tasks with imbalanced data. During the training phase our approach is divided into two parts: first, the data are encoded into a latent representation (vector space) using the sparse auto-encoder. Then, that feature representation extracted from the auto-encoder is used to train the complementary SSGAN (SSGAN-c). The contributions of this chapter are summarized as follows:

- We propose a new architecture for imbalanced data classification which does not require oversampling techniques to produce good classification results.

- Our results on the benchmark datasets are promising; our method outperforms logistic regression, random forest, XGBoost and multi-layer perceptron with improvements on F1 score for all the datasets that were examined.

- We apply the proposed SSGAN architecture to a real-world problem of money laundering in online gambling, obtaining better classification results than an existing anti-money laundering detection system. The F1 score was improved by 3.64%.

The remainder of this chapter is organised as follows: Section 6.2 describes the proposed semi-supervised GAN model in detail. Section 6.3 presents the experimental results. Section 6.4 discusses the application of the model to money laundering detection in gambling.

6.2 SSGAN for Fraud Detection

6.2.1 Framework Description

The structure of the proposed framework is illustrated in Figure 6.1. It consists of two parts: a sparse auto-encoder and a complementary generative adversarial network. In this architecture, the sparse auto-encoder includes two encoding layers and two decoding layers. During the encoding phase the input data are projected into a higher dimension, while in the decoding phase the network tries to reconstruct the input data from the sparse representations of the data. Mapping the data onto a higher dimension during encoding seeks to increase the distance between positive and negatives samples as Figure 6.2a and Figure 6.2b illustrate.

The data representations extracted from the SAE are used as input to the generative adversarial network. Our GAN adopts a complementary generator which tries to match the data representations from a Gaussian random noise in order to generate new complementary samples. Together with the real representations the generated samples are used to train a discriminator model. After training is complete, the discriminator is used to distinguish and detect the high-risk for money laundering cases.

The pseudo-code of training SSGAN-c is shown in Algorithm 1. Given a training dataset T

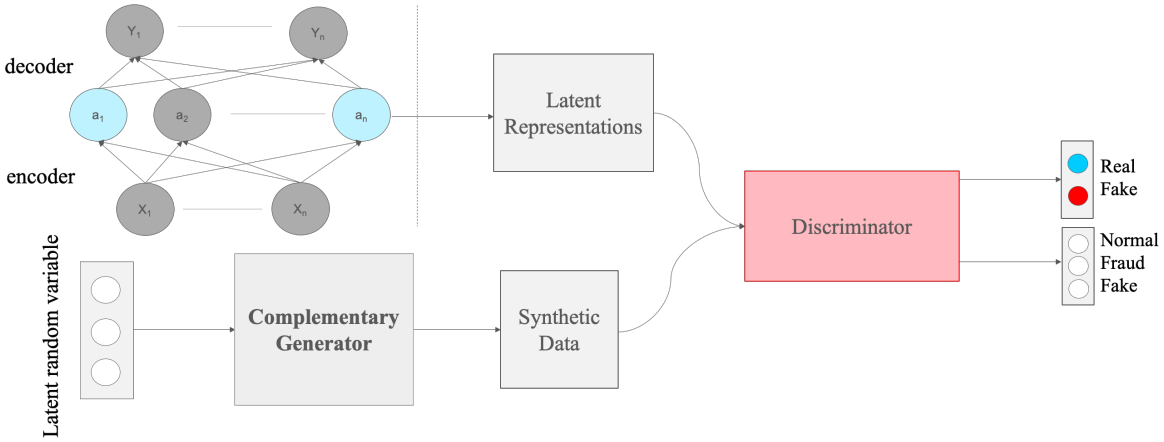


Figure 6.1: Architecture of the proposed system (SSGAN-c) showing (on the left) a sparse Autoencoder mapping the data onto a higher-dimensional vector space. The output of the encoder is used as input to the generative adversarial network (on the right). After training, the discriminator of the GAN is able to classify the data as fraud or normal.

Algorithm 1: Training Complementary SSGAN

Inputs : Training dataset $T = X_1, X_2, X_3, ...$
Training Epochs for auto-encoder,
$Epoch_{AE}$ and $Epoch_{GAN}$

Outputs: A trained auto-encoder and complementary SSGAN

```
parameters initialisation for spare auto-encoder and complementary SSGAN;
j ← 0 ;
while j < Epoch_AE do
    reconstruct the players data with auto-encoder;
    optimize parameters of the auto-encoder to reduce loss function;
    project the the data into higher dimension space through the encoding stage;
    output latent representations;
    j ← j+1 ;
end
while j < Epoch_GAN do
    optimise the discriminator D ;
    optimise the generator G;
end
return trained sparse auto-encoder and complementary SSGAN;
```

that contains the feature vectors of n number of gambling players, we first train the auto-encoder model in order to project the data into a higher dimensional space during the encoding phase. Then using those representations, we train the complementary SSGAN. At the end of the process our model is able to perform a binary classification task.

6.2.2 Sparse Auto-encoders for Latent Representation

The framework's sparse auto-encoder consists of a feedforward neural network whose hidden layer is larger than the size of the input layer and whose target output is by definition equal to the input vector [176]. The output of the hidden layer within the auto-encoder represents the encoding of the input x into a sparse latent feature representation. This type of neural network tries to learn a function $h_{W,b}(x) \approx x$ in order to reproduce an output x' that is similar to x [177].

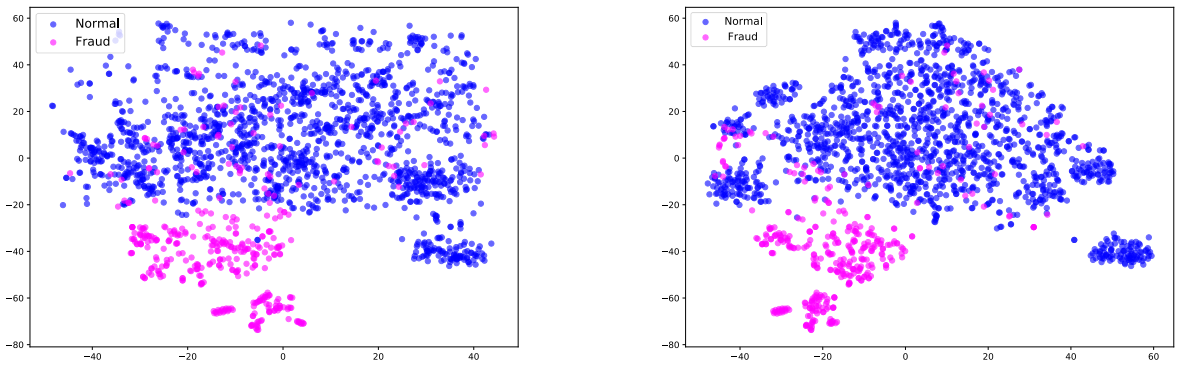
Extending the idea of the original auto-encoder, a sparse auto-encoder incorporates to the reconstruction error a sparse penalty term $\Omega(h)$ w.r.t. the hidden layer h [178, 179]. This penalty on the activation of the units of a neural network seeks to make the representation

sparse with the objective of producing more robust and generalised features [180]. The sparsity term can be imposed on the output layer of the encoder or on a hidden layer or bottleneck. In our sparse auto-encoder, we applied the L1 regularisation which enforces sparsity by allowing some activations to become zero. The loss function of a sparse auto-encoder is defined in equation (6.1):

$$L(x, g(f(x))) + \Omega(h) \tag{6.1}$$

where $g(h)$ is the output of the decoder and $h = f(x)$ is the output of the encoder. The penalty term $\Omega(h)$ can be further expressed as $\Omega(h) = \lambda \sum_i \left| a_i^{(h)} \right|$. The loss function penalises the absolute value of the vector of activation functions a in the hidden layer for an observation i, scaled by a tuning parameter λ.

The choice of an sparse auto-encoder over the original auto-encoder is supported by [178]. In that paper, the authors suggest that using a sparse auto-encoder enables robust feature extraction from the input. In addition, projecting the data to higher dimensional spaces is more likely to result in an easier classification task [181]. In this chapter, the data representations extracted from the hidden layers of the auto-encoder are denoted by $\tilde{x}$.



(a) Original data

(b) Representation data

Figure 6.2: Figure 6.2a and Figure 6.2b show the original and representation training data distribution for the Credit Card Fraud dataset in 2D space using t-SNE.

6.2.3 Training the complementary Generator of SSGAN

Although GANs are very promising for new data generation, due to the vanishing gradient problem, training GANs could be really unstable. However, this can be improved when the model architecture and hyper-parameters are carefully selected [182]. Building on the generator we used in our SDG-GAN framework where we applied feature matching loss function to provide stability during training, we extend that approach and we introduce further improvements in the generator.

GANs can be extended to semi-supervised learning by defining another output in the discriminator. The first output of the discriminator only classifies data as real or fake, while the second output classifies the data by the class that they belong. The idea is that whether the data are real or fake, the classifier has to determine whether it can classify the samples into their true class. if it can, then the data are probably real.

Inspired by the work of [183] and [174], we implement a complementary generator. Our generator is a two layer feed forward neural network that tries to learn the distribution of the representations (output of the encoder) and not the actual data distribution. The new generated samples have the same dimension as the latent representations and are defined by $n = G(z)$.

By following the approach of [174], the complementary generator with output p_g tries to learn the distribution $p^*(n)$ which is defined as:

$$p^*(n) = \begin{cases} \frac{1}{r} * \frac{1}{p(n)} & if\ p(n) > \tau\ and\ n \in B_{\tilde{x}} \\ \\ C & if\ p(n) \leq \tau\ and\ n \in B_{\tilde{x}} \end{cases} \tag{6.2}$$

where r is a normalisation term, and $B_{\tilde{x}}$ is the feature space of the extracted feature representations, C is a constant and τ is the threshold for separating low and high density data. As a result, the generator now is trained in order to converge its distribution (p_g) to the new complementary distribution p^*. Using the definition of the KL divergence:

$$\begin{aligned} KL_{(p_g \;||\; p_g^*)} = -H(p_g) + E_{n\sim p_g} log\ p(n)\ \mathbb{I}[p(n) > \tau] \\ + E_{n\sim p_g}(\mathbb{I}[p(n) > \tau]\ log\ r - \mathbb{I}[p(n) \leq \tau]\ log\ C) \end{aligned} \tag{6.3}$$

In the above equation, $\mathbb{I}$ denotes the indicator function and H the entropy function. As it is stressed by [174] the final term of equation (6.3) would not add any further information and can be ignored. The generator also adapts the feature matching loss [161] that we introduced as part of the generator of SDG-GAN in order to bring the generated representations closer to the real representations. The final objective function of the complementary generator is the following:

$$\begin{aligned} \min_G\ -H(p_g) + E_{n\sim p_g} log\ p_{data}(n)\mathbb{I}[p_{data}(n) > \tau] \\ + \left\| E_{n\sim p_g f(n)} - E_{\tilde{x}\sim p_{data}} f(\tilde{x}) \right\|_2^2 \end{aligned} \tag{6.4}$$

where the last term of equation (6.4) describes the feature matching loss and $f(\tilde{x})$ is defined as the output of the hidden layer in the discriminator.

One of the main disadvantages of training a generative adversarial network is the *mode collapse* scenario. It is also known as the problem that occurs when the generator learns to map several different input z values to the same output point [184]. This problem is directly related to the entropy distribution of generated features and is a sign of low entropy. Therefore, to improve further our generator, from [183, 174], we adopted a pulling away term (PT) which was introduced in [185] to increase the generator's entropy, defined as:

$$L_{PT} = \frac{1}{N-1} \sum_{i}^{N} \sum_{j\neq i}^{N} \left(\frac{f(n_i)^T f(n_j)}{\|f(n_i)\|\ \|f(n_j)\|}\right)^2 \tag{6.5}$$

where N is the size of the mini-batch and $f(n)$ is the output of the hidden layer of the discriminator.

6.2.4 Training the Discriminator

Following the architecture of [161], the output of the discriminator is mapped onto a softmax classifier. Assuming that there are K possible classes in the data, semi-supervised learning is performed by including the new (fake) samples from the generator in our data and labeling them with a new class $K+1$. The dimension of the discriminator output is increased to $K+1$. Moreover, an additional output is added to the soft-max classifier in order to distinguish the real and fake samples. Our discriminator loss function can be described as follows:

$$\begin{aligned} L = E_{\tilde{x},y\sim p_{data}(\tilde{x},y)}\ [log\ p_{model}(y|\tilde{x})]+ \\ E_{\tilde{x}\sim G}\ [log\ p_{model}(y=K+1|\tilde{x})] \end{aligned} \tag{6.6}$$

where $p_{model}(y = K+1|\tilde{x})$ is defined as the probability that x is fake and $p_{model}(y|\tilde{x})$ as the probability that x belongs to a real class. The loss function in (6.6) is divided into supervised loss $L_{supervised}$ and unsupervised loss $L_{unsupervised}$:

$$L_{supervised} = E_{\tilde{x},y\sim p(\tilde{x},y)}\ [log\ p_{model}(y|\tilde{x}, y < K+1)] \tag{6.7}$$

$$\begin{aligned} L_{Unsupervised} = E_{x\sim p_{data}(\tilde{x})}\ [1 - log\ p_{model}(y=K+1|\tilde{x})] \\ +E_{\tilde{x}\sim G}[log\ p_{model}(y=K+1|\tilde{x})] \end{aligned} \tag{6.8}$$

where $L_{supervised}$ is the typical supervised loss and $L_{unsupervised}$ is the loss generated from the GAN. A main contribution of this work is to highlight the classification ability of GANs in supervised learning tasks on structured data, including the imbalanced class problem, as discussed in the next section.

6.3 Experimental Results

Our proposed method was evaluated through three different sets of experiments: (1) We compared the SSGAN framework with four classification machine learning algorithms: logistic regression, random forest, XGBoost and multi-layer perceptron. We trained these methods with the imbalanced datasets of Breast Cancer and Diabetes that we introduced in Chapter 5. In addition, we generated results when these three algorithms were combined with oversampling techniques of SMOTE, ADASYN and SDG-GAN. (2) We investigated the effect of the sparse auto-encoder on the results by training the benchmark algorithms and our framework with the original data of Credit Card Fraud and with the latent representations. We also compared our method with a semi-supervised GAN trained with a regular generator (SSGAN-r). (3) We applied our framework to real-world data of Gambling Fraud dataset that was used in Chapter 5 and we demonstrate the value of the framework in that application domain with a comparison of results.

6.3.1 Hyperparameters Settings

Table 6.1: SSGAN-c hyperparameters settings

Hyperparameter	**Value**
SAE Learning Rate	$1x10^{-5}$
SAE Penalty	L1 regularisation
SAE Epochs	300
SAE Latent Dimensions	65
Encoder layer sizes	(input,32), (32, latent dim)
Decoder layer sizes	(latent dim, 50), (50, data size)
SSGAN Learning Rate	$1x10^{-3}$
SSGAN Optimiser	*Adam*
SSGAN Batch size	64
Generator layers sizes	(Noise, 100), (100, latent dim)
Discriminator Layer sizes	(latent dim, 64), (64, latent dim)
Discriminator Output	(latent dim, 1), (latent dim, nclasses+1)
Activation function in hidden layers	Leaky ReLU
Noise Distribution p_z	N(0,1)
Noise size	50

Similar to Chapter 5 and the process of tuning SDG-GAN, after experimenting with different

hyperparameters, we selected the hyperparameters in Table 6.1 since they produced the best results. The proposed framework consists of two networks, one SAE and the SSGAN, both networks hyperparameters need to be defined. In Table 6.1 we present first the hyperparameters of SAE and then the hyperparameters of SSGAN.

Unfortunately there is a fine line between sufficiently complex and too complex. During the auto-encoder tuning we wanted to reduce the validation loss from the reconstruction but at the same time we are trying to avoid overfitting the model. The dimension of the latent dimensions between the encoder and decoder part is 65, and the training epoch is 300. The learning rate is set to $1x10^{-5}$.

In the complementary SSGAN model, both discriminator and generator are feedforward neural networks. Specifically, the discriminator contains two layers. Further, the discriminator has two outputs with one with sigmoid activation function for recognising the fake and real samples, while the second output has a softmax activation with size equal to the number of classes+1. The generator takes the dimension of noise as input. The output layer of the generator has the same dimension as the latent representations from the encoder which is 65 in our experiments. These parameters were applied in all experiments for the different datasets.

6.3.2 Results and Comparison

The presented results illustrate the mean value and standard deviation for accuracy, recall, precision and F1 score on 10 different runs. In all the experiments the training and testing set ratio is set to 80% and 20% respectively. Table 6.2 and Table 6.3 show the results obtained for the Breast Cancer and Pima Diabetes datasets. It is evident that our framework achieves the best performance for both datasets with F1 scores 92.27% and 69.04% for the Breast Cancer and Diabetes datasets, respectively. Table 6.2 shows that the F1 value is increased by 3.88% when all the algorithms are trained with an imbalanced dataset. Although, the discriminative models improved their performance when they were combined with oversampling techniques (c.f. increase in their recall score), still they are outperformed by our method by 2.92% on the F1 score.

For the Diabetes dataset, the classifiers performed poorly due to the high intersection between the negative and positive samples. However, our method enhanced the F1 score by 5.65% on imbalanced training. Again, when ADASYN and SMOTE were combined with LR, RF and MLP, the recall value is increased significantly but precision is decreased. This suggests that when oversampling is used, classification algorithms are able to identify better the minority class, still their performance related to the majority class is reduced.

We further evaluate the proposed method on the Credit Card Fraud dataset. The algorithms are trained with the original data, data extracted using Principal Component Analysis (PCA) as a baseline, and representation data from the sparse auto-encoder. The results are reported in Table 6.4. The performance of SSGAN is improved significantly when the representations from the auto-encoder are used to train the model with an increase of 3.76% of recall and 2.31% of the F1 score. This validates our choice to use the extracted features from the sparse auto-encoder as input to the GAN framework. Table 6.5 shows the results of the discriminative models in combination with ADASYN and SMOTE for the Credit Card dataset. Importantly, the SSGAN framework continues to achieve the best F1 score of 92.31%. In Table 6.4 we also show the results when we train the SSGAN with a regular generator (SSGAN-r) as opposed to the complementary generator (SSGAN-c) used in our framework. A regular SSGAN has

Table 6.2: Breast Cancer detection results ($mean \pm std$): accuracy, recall, precision and F1 measure

Breast Cancer				
Method	**Accuracy**	**Recall**	**Precision**	**F1**
LR	0.8959 ± 0.0093	0.8053 ± 0.0345	0.9095 ± 0.0279	0.8534 ± 0.0185
LR + SMOTE	0.9114 ± 0.0175	0.8762 ± 0.0358	0.8813 ± 0.0387	0.8778 ± 0.0242
LR + ADASYN	0.9005 ± 0.0196	$\mathbf{0.9448 \pm 0.0300}$	0.8182 ± 0.0289	0.8766 ± 0.0238
RF	0.9134 ± 0.0110	0.8891 ± 0.0337	0.8802 ± 0.0230	0.8839 ± 0.0152
RF + SMOTE	0.9187 ± 0.0181	0.9232 ± 0.0353	0.8674 ± 0.0379	0.8935 ± 0.0239
RF + ADASYN	0.9052 ± 0.0245	0.9392 ± 0.0283	0.8230 ± 0.0316	0.8771 ± 0.0277
XGB	0.9044 ± 0.0209	0.8810 ± 0.0522	0.8653 ± 0.0426	0.8714 ± 0.0289
XGB + SMOTE	0.8974 ± 0.0147	0.8810 ± 0.0532	0.8483 ± 0.0270	0.8629 ± 0.0227
XGB + ADASYN	0.8886 ± 0.0257	0.9286 ± 0.0384	0.8030 ± 0.0421	0.8603 ± 0.0300
MLP	0.9157 ± 0.0309	0.8732 ± 0.0761	0.8889 ± 0.0528	0.8782 ± 0.0475
MLP + SMOTE	0.9093 ± 0.0294	0.9207 ± 0.0324	0.8468 ± 0.0599	0.8800 ± 0.0251
MLP + ADASYN	0.8871 ± 0.0264	0.8894 ± 0.0487	0.8361 ± 0.0739	0.8578 ± 0.0288
SSGAN-c+SAE	$\mathbf{0.9227 \pm 0.0193}$	0.9113 ± 0.0270	$\mathbf{0.93485 \pm 0.0238}$	$\mathbf{0.9227 \pm 0.0193}$

Table 6.3: Diabetes detection results ($mean \pm std$): accuracy, recall, precision, F1 measure

Pima Diabetes				
Methods	**Accuracy**	**Recall**	**Precision**	**F1**
LR	0.7656 ± 0.0204	0.5074 ± 0.0598	$\mathbf{0.7455 \pm 0.0483}$	0.6013 ± 0.0440
LR+SMOTE	0.7604 ± 0.0298	0.6685 ± 0.0315	0.6598 ± 0.0601	0.6626 ± 0.0333
LR+ADASYN	0.7370 ± 0.0335	0.7444 ± 0.0785	0.6006 ± 0.0385	0.6638 ± 0.0406
RF	0.7688 ± 0.0193	0.5741 ± 0.0603	0.7145 ± 0.0405	0.6339 ± 0.0386
RF+SMOTE	0.7442 ± 0.0236	0.7037 ± 0.0530	0.6203 ± 0.0339	0.6582 ± 0.0324
RF+ADASYN	0.7357 ± 0.0363	0.7741 ± 0.0567	0.5965 ± 0.0451	0.6727 ± 0.0420
XGB	0.7383 ± 0.0191	0.5963 ± 0.0560	0.6376 ± 0.0356	0.6142 ± 0.0318
XGB + SMOTE	0.7591 ± 0.0301	0.6870 ± 0.0738	0.6467 ± 0.0361	0.6652 ± 0.0505
XGB + ADASYN	0.7253 ± 0.0133	0.6759 ± 0.0470	0.5960 ± 0.0184	0.6325 ± 0.0232
MLP	0.7513 ± 0.0409	0.5741 ± 0.0824	0.6748 ± 0.0780	0.6166 ± 0.0679
MLP+SMOTE	0.7591 ± 0.03551	0.7435 ± 0.0725	0.6357 ± 0.0483	0.6834 ± 0.0467
MLP+ADASYN	0.7351 ± 0.0476	$\mathbf{0.8000 \pm 0.0880}$	0.5907 ± 0.0531	0.6786 ± 0.0610
SSGAN-c+SAE	$\mathbf{0.7906 \pm 0.0321}$	0.6515 ± 0.0535	0.7381 ± 0.0428	$\mathbf{0.6904 \pm 0.0210}$

Table 6.4: Credit card fraud detection results ($mean \pm std$): accuracy, recall, precision and F1 measure

Credit Card Fraud					
Input	**Method**	**Accuracy**	**Recall**	**Precision**	**F1**
Original Data	SSGAN-c	0.9629 ± 0.0032	0.8297 ± 0.0230	0.9874 ± 0.0135	0.9005 ± 0.0096
	SSGAN-r	0.9424 ± 0.0367	0.8109 ± 0.0627	0.9041 ± 0.1105	0.8538 ± 0.0829
	Logistic Regression	0.9577 ± 0.0110	0.7972 ± 0.0523	$\mathbf{0.9954 \pm 0.0046}$	0.8853 ± 0.0328
	Random Forest	0.9667 ± 0.0013	0.8393 ± 0.0086	0.9924 ± 0.0032	0.9086 ± 0.0041
	XGBoost	0.9705 ± 0.0036	0.8707 ± 0.02780	0.9790 ± 0.0145	0.9212 ± 0.0112
	MLP	0.9629 ± 0.0014	0.8264 ± 0.0259	0.9888 ± 0.0159	0.9000 ± 0.0087
PCA	SSGAN-c	0.9381 ± 0.0298	0.8113 ± 0.0353	0.8426 ± 0.1209	0.8426 ± 0.0642
	SSGAN-r	0.9577 ± 0.0054	0.8001 ± 0.0351	0.9821 ± 0.01380	0.8822 ± 0.0179
	Logistic Regression	0.9409 ± 0.0275	0.7737 ± 0.0565	0.9224 ± 0.1003	0.8400 ± 0.0696
	Random Forest	0.9519 ± 0.0102	0.7990 ± 0.0354	0.9513 ± 0.0319	0.8681 ± 0.0290
	XGBoost	0.9484 ± 0.0166	0.8274 ± 0.0338	0.9082 ± 0.0599	0.8650 ± 0.0392
	MLP	0.9152 ± 0.0444	0.8000 ± 0.0356	0.8081 ± 0.1540	0.7968 ± 0.0842
Latent Representations	SSGAN-c	$\mathbf{0.9707 \pm 0.0019}$	0.8673 ± 0.0125	0.9869 ± 0.0165	$\mathbf{0.9231 \pm 0.0047}$
	SSGAN-r	0.9158 ± 0.0323	0.6404 ± 0.1886	0.9139 ± 0.0686	0.9158 ± 0.1551
	Logistic Regression	0.9232 ± 0.0124	0.6230 ± 0.0571	0.9911 ± 0.0117	0.7633 ± 0.0416
	Random Forest	0.9349 ± 0.0128	0.6970 ± 0.0682	0.9690 ± 0.0170	0.8089 ± 0.0488
	XGBoost	0.9611 ± 0.0043	0.8363 ± 0.0258	0.9655 ± 0.0163	0.8959 ± 0.0130
	MLP	0.9619 ± 0.0016	0.8334 ± 0.0058	0.9706 ± 0.0066	0.9151 ± 0.0331

the same architecture as the original GAN model with the addition of an extra output in the discriminator. Our framework improves consistently on the regular generator.

In order to examine the sensitivity level of the sparse auto-encoder in our experiments we altered the hidden dimension size from 20 up to 80 neurons and the effect in the overall performance can be found in Figure 6.3. Overall, the model was robust with the performance remaining stable

Table 6.5: Credit Card Fraud detection results in conjunction with oversampling ($mean \pm std$): accuracy, recall, precision, F1 measure. Comparing with the results of Table 6.4, our proposed architecture achieves the highest F1 measure.

Methods	Accuracy	Recall	Precision	F1
LR + SMOTE	0.9691 ± 0.0055	0.8837 ± 0.0286	0.9577 ± 0.0151	0.9189 ± 0.0156
RF + SMOTE	0.9826 ± 0.0043	0.8620 ± 0.0346	0.9521 ± 0.0184	0.9045 ± 0.0245
XGB + SMOTE	0.9683 ± 0.00370	0.8848 ± 0.0203	0.9526 ± 0.0163	0.9172 ± 0.0100
MLP + SMOTE	0.9682 ± 0.0086	0.8858 ± 0.0283	0.7956 ± 0.0778	0.8353 ± 0.0376
LR + ADASYN	0.9154 ± 0.0147	0.9172 ± 0.0210	0.7272 ± 0.0465	0.8103 ± 0.0285
RF + ADASYN	0.9677 ± 0.0042	0.8561 ± 0.0279	0.9769 ± 0.0200	0.9120 ± 0.0136
XGB + ADASYN	0.9619 ± 0.0094	0.9081 ± 0.0294	0.9021 ± 0.0348	0.9045 ± 0.0229
MLP + ADASYN	0.8760 ± 0.0168	$\mathbf{0.9394 \pm 0.0262}$	0.6269 ± 0.0365	0.7517 ± 0.0328

with small variations across different dimensions. Precision had a small variation throughout all the different experiments due to the large number of non-fraudulent cases in the training set. On the other hand, recall had a significant increase when the dimension changed from 20 to 30 ($\approx 10\%$) and a small increase from 60 to 65 ($\approx 2\%$). Then, when dimension size changes from 65 to 80 a small decrease in recall was observed. Mapping the original data to a higher dimension allows data to be separated more easily. Nevertheless, if the dimension is too high it can lead to overfitting and information redundancy [186]. Finally, we also noted that the trend of F1 score follows the trend of recall as fluctuations occurred in the same examples for both metrics.

Focusing on the semi-supervised GANs, the SSGAN-c has better performance compared to SSGAN-r as shown in Table 6.4. The discriminator of SSGAN-c, which is trained on real and complementary data, can classify more effectively the positive and negative cases since better recall and precision scores are achieved compare to SSGAN-r.

The training behaviour of the two models was further investigated and the progress of the F1 score on the Credit Card dataset is presented in Figure 6.4a and Figure 6.4b. In Figure 6.4, the regular SSGAN shows an inability to converge during training, while the SSGAN-c framework converges. The reason for this is that during the training phase the complementary GAN focuses on the classification task of predicting the correct class whilst the regular GAN focuses on generating better fake samples [183].

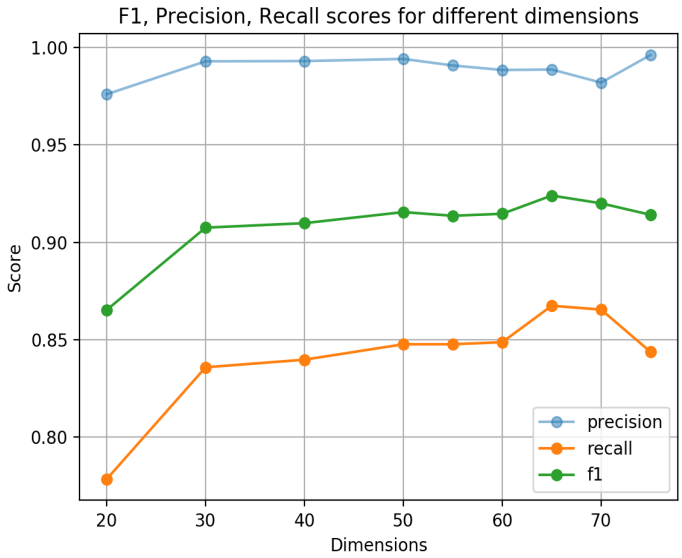


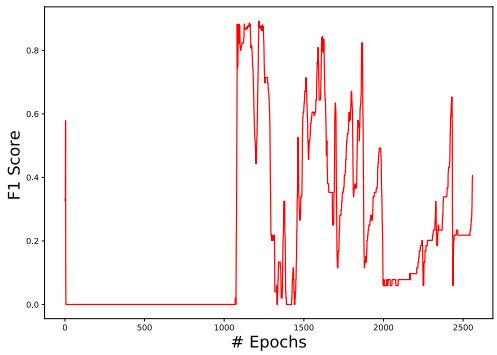
Figure 6.3: Latent Representation dimensions. The performance of SSGAN-c is improved when we increased the dimensions from 20 to 30 with the highest performance observed when latent dimensions equal to 65.

6.4 Application of SSGAN to the Detection of Money laundering in Online Gambling

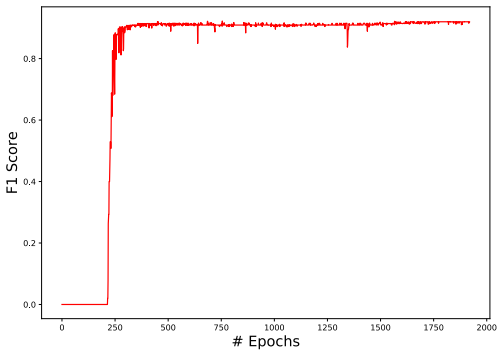
Finally, SSGAN-c is tested on the real-world Gambling Fraud dataset. Our system targets to improve the first level of monitoring (identification rate of rule-based system). The dataset, as Table 5.3 shows, contains 4,700 samples, 3,500 of which are non-fraudulent players and the remaining 1,200 players are flagged for potential money laundering and further investigation. The F1 value of the rule-based system as this is calculated on the test set for all the 10 evaluation runs using the IRR labels and the detection flags of the system is 86.91%.

Table 6.6 outlines the comparative results obtained for the Gambling Fraud dataset. The SSGAN-c framework achieved F1 score of 89.85%, which yields a 3.64% ($\approx$ 20 cases) improvement on the company's current detection system and an 0.52% ($\approx$ 3 cases) improvement in comparison with the other methods. This is an indication that our fraud detection system can be applied to the detection of high-risk for money laundering players in online gambling and improve the overall identification rate.

In this chapter we presented results on similar experiments with Chapter 5 in terms of the



(a) F1 of SSGAN-r



(b) F1 of SSGAN-c framework

Figure 6.4: Fig. 6.4a and Fig. 6.4b show the F1 score progress during training of a regular SSGAN and our complementary SSGAN framework.

datasets that have been used and the methods that we compared our framework against. The overall classification results on the datasets presented in this chapter are slightly different compare to the results from Chapter 5. The random effect when algorithms are trained as well as the random split between training and testing sets were the reasons why this difference is observed. Since SSGAN-c managed to have the best overall performance on the gambling dataset we analysed further the results on the gambling dataset. In total 17 false negatives and 49 false positives were observed out of 940 testing samples.

Table 6.6: Gambling Fraud detection results (*mean* ± *std*): accuracy, recall, precision, F1 measure

Gambling Fraud				
Methods	**Accuracy**	**Recall**	**Precision**	**F1**
LR	0.8733 ± 0.0091	0.6103 ± 0.0296	0.8751 ± 0.0224	0.7187 ± 0.0234
LR+SMOTE	0.9089 ± 0.0103	0.8482 ± 0.0206	0.8164 ± 0.0225	0.8318 ± 0.0185
LR+ADASYN	0.9000 ± 0.0068	0.8960 ± 0.0152	0.7671 ± 0.0171	0.8264 ± 0.0104
RF	0.9424 ± 0.0059	0.9095 ± 0.0212	$\mathbf{0.8781 \pm 0.0115}$	0.8933 ± 0.0116
RF+SMOTE	0.9361 ± 0.0071	0.9458 ± 0.0062	0.8355 ± 0.0146	0.8872 ± 0.0106
RF+ADASYN	0.9347 ± 0.0057	$\mathbf{0.9569 \pm 0.0139}$	0.8256 ± 0.0165	0.8862 ± 0.0091
XGB	0.9393 ± 0.0049	0.9012 ± 0.0141	0.8748 ± 0.0183	0.8875 ± 0.0083
XGB + SMOTE	0.9441 ± 0.0091	0.9186 ± 0.0173	0.8768 ± 0.0190	0.8972 ± 0.0164
XGB + ADASYN	0.9435 ± 0.0055	0.9332 ± 0.0135	0.8652 ± 0.0161	0.8978 ± 0.0095
MLP	0.9194 ± 0.0100	0.8419 ± 0.0289	0.8534 ± 0.0245	0.8472 ± 0.0195
MLP+SMOTE	0.9203 ± 0.0060	0.9372 ± 0.0248	0.7984 ± 0.0167	0.8618 ± 0.0103
MLP+ADASYN	0.9219 ± 0.0039	0.9526 ± 0.0133	0.7946 ± 0.0118	0.8663 ± 0.0060
SSGAN-c+SAE	$\mathbf{0.9437 \pm 0.0051}$	0.9308 ± 0.0157	0.8672 ± 0.0170	$\mathbf{0.8985 \pm 0.0088}$

Expanding further our analysis on the false positives and false negatives, we produced the plots in Figure 6.5 where we show the histograms of the most important features as these are defined by the SHAP analysis in Figure 4.5. We randomly selected 1,000 positive samples (high-risk samples), 1,000 randomly selected negative samples together with all the false positives and the false negatives generated from SSGAN-c. The red and black dashed line in Figure 6.5 corresponds to the false negatives and false positives mean values respectively. Firstly, looking at the false positives we can see that features values lie mostly within the distribution of the positive samples as the black dashed line indicates. Then secondly, for the false negatives it can be observed that there are features where the mean values was positioned closer to the low-risk class and other cases closer to the high-risk class.

6.5 Summary

In this chapter we proposed a GAN-based system architecture for detecting fraud in online gambling. Our system consists of a complementary generative adversarial network and a sparse auto-encoder. First, we used the auto-encoder to extract new data representations which were

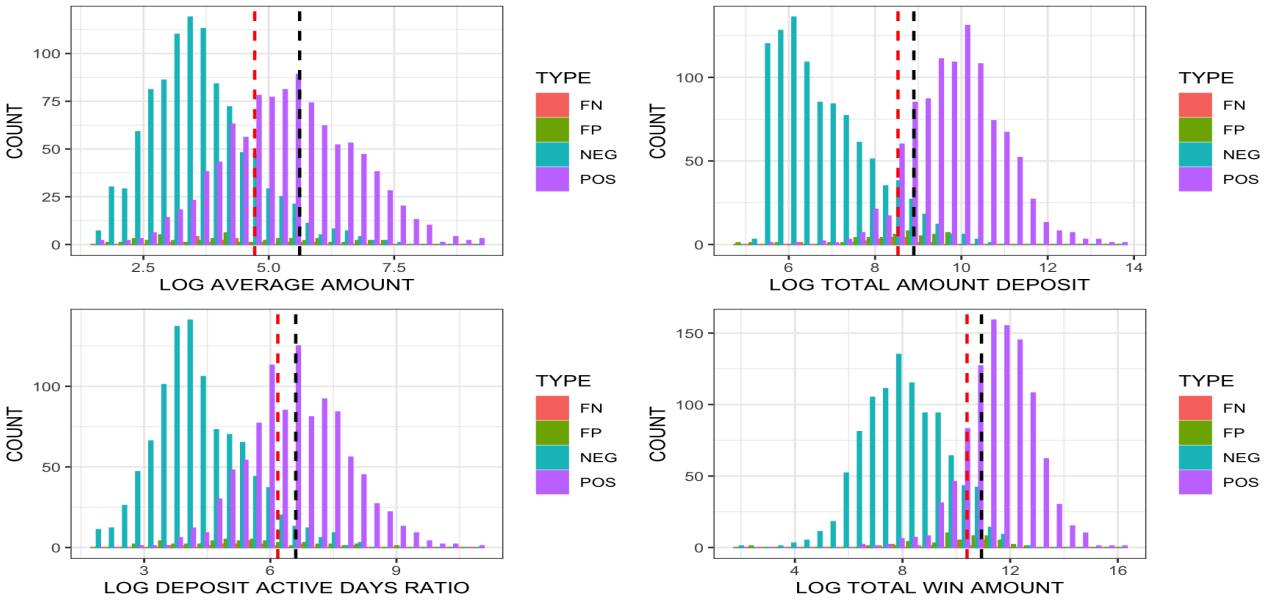


Figure 6.5: Histogram plots in log scale of the most important features from the SHAP analysis in Figure 4.5. The black dashed line represents the mean value of the false positives and the red dashed line the mean value of the false negatives.

then used to train our GAN model. A series of experiments were performed to evaluate the proposed system architecture against popular discriminative models such as logistic regression and random forest, both on their own and in conjunction with data balancing techniques of SMOTE and ADASYN. Following a similar experimental process with Chapter 5, experiments were performed on three publicly available datasets and on the real-world gambling dataset. We demonstrated that our system outperforms the other classification methods by achieving the highest F1 score. In addition, when our system was compared with the existing rule-based system of our research partners, better results were achieved by SSGAN-c. Overall, the results showed that complementary semi-supervised GANs can be a useful versatile framework for tackling supervised problems with imbalanced and sparse structured data. In future work, results could be compared with other deep network models including with the use of sparse coding [187].

Chapter 7

Anomaly Detection

7.1 Introduction

Although the results with supervised learning showed good classification performance, it was not possible to discover new trends and patterns related to the risk for money laundering. In this chapter, we are introducing a monitoring system for generating real-time alerts. The system was used to detect and classify anomalies on players' bets and transactions. The output of our system is a probability score indicating whether a particular set of action is an anomaly or not. The smallest the score, the higher the chance of the specific action to be an anomaly. The The system was evaluated empirically with the assistance of player risk experts from Kindred Group. Due to time limitations, our partners were only able to review five individual cases of anomaly detection, and their feedback is presented in this chapter.

Anomaly detection (AD) is a significant research problem with various application domains. Many techniques have been developed specifically for certain applications, while others are more generic[94]. One of the things that stood out during the literature review in Chapter 2 was that on multivariate time series data, LSTM has been proven to be efficient in detecting complex relationships. Laptev, Amizadeh and Flint[188] built an AD system that separates forecasting, AD and alerting into three separate components. The alerting component uses machine learning to select the most relevant anomalies for each consumer. Chauhan and Vig [189] utilised a

deep recurrent neural network architecture with LSTM units to develop a predictive model for healthy ECG signals. They further utilised the probability distribution of the prediction errors from these recurrent models to indicate normal or abnormal behaviour. The authors in [190] developed an online multi-task detection algorithm based on LSTM for action recognition and task prediction. Konstantinos et al. [191] combined online sequential extreme learning machines (OS-ELM) and restricted Boltzmann machines (RBMs), aiming to classify critical infrastructures' network flow for AD. Pankaj et al. [192] applied an LSTM encoder–decoder framework to learn and reconstruct normal time series; using the reconstruction error, they detected any anomalies. Malhotra et al. [118] used a stacked LSTM network for prediction and a Gaussian distribution for AD in a time series. In the anomaly detection field, multivariate Gaussian has been extensively used for abnormal event detection [193], [194], [195].

A gambling time series signal could provide information about the transactions and betting data of a player. These data could be used for predicting the player's next action, i.e. deposit, withdrawal or bet, and predict the action's amount. However, there are many obstacles that are preventing machine learning algorithms from being successful when applied to gambling-related time series. Players' actions can vary, and there is not a particular order that could occur. In this chapter, we propose and examine a hybrid method based on an encoder–decoder LSTM network with an Attention mechanism (LSTM-ATT) and multivariate Gaussian for AD on multivariate time series data. The LSTM-ATT is employed to capture temporal patterns and predict the next action of a player (meaning bet, deposit or withdrawal) together with the amount of the action. The predictions from the temporal network are then compared with the actual values of the amount and the type of action, and the respective classification and regression errors are calculated. Finally, these errors vectors are modelled to fit a multivariate Gaussian distribution $N = N(\mu, \Sigma)$, which is used to assess the likelihood of anomalous behaviour.

Section 7.2 introduces the background theory behind the anomaly detection framework. Section 7.3 presents different threads that exist in online gambling and could lead to money laundering. Section 7.4 introduces the architecture of the proposed AD framework. Section 7.5 explores the raw dataset used in the experiments in this chapter. Sections 7.6 and 7.7 shows the experimental process and results from the AD system. Finally, we show the evaluation of the system that

was conducted with the assistance of field experts from Kindred in Section 7.8.

7.2 Background theory

Recurrent neural networks are a class of neural networks that are naturally suited for processing time series sequential data [196]. Similar to standard neural networks, RNNs are composed with input, hidden and output layers. The difference is that in this architecture each neuron is assigned to a fixed time step. The neurons in the hidden layer are also forwarded in a time-dependent direction. The input and output neurons are connected only to the hidden layers, with the same assigned time step. In an RNN, the information cycles through a loop and takes into consideration both the current input and what it has learned from the past inputs.

The architecture of an RNN is shown in Figure 7.1; the input vector, X(t), at each time step is connected to the hidden layers through a weight vector, U. At the same time, the hidden layer neurons are connected to the neurons of different time steps by a weight matrix, W. Finally, the neurons of the hidden layer are connected to the output by a weight matrix, V. All the weight matrices remain constant at each time step. It can be concluded that RNN cells share the same weights and parameters across multiple time steps and that their hidden state is updated as follows:

$$h_t = \begin{cases} 0, & t = 0 \\ \phi(Wh(t-1) + Ux(t))), & otherwise \end{cases} \tag{7.1}$$

where ϕ is a nonlinear function transformation, i.e. a logistic sigmoid or tanh function. During the training phase of an RNN, the goal is to optimise the weight matrices of U, W and V in order to generate the best output, y(t). Although recurrent neural networks have been successful in many tasks, such as text generation and speech recognition, it is hard for RNNs to learn a long-term sequence due to the vanishing and exploding gradient problem that propagates through their multiple layers. That means that the parameters in the hidden layers either

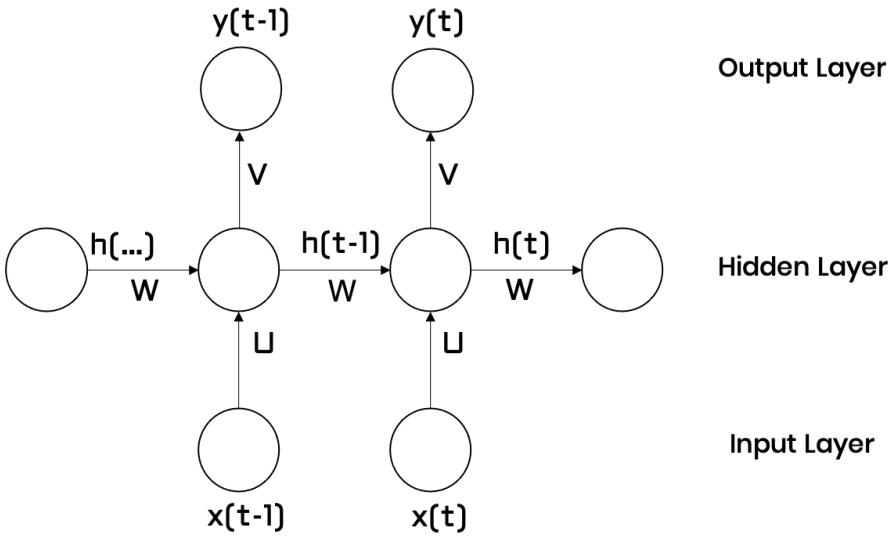


Figure 7.1: Illustration of recurrent neural network architecture

do not change that much or lead to numeric instability and chaotic behaviour. To address this problem, different extensions of recurrent neural networks have emerged, such as LSTM networks, and Gated Recurrent Unit (GRU) networks and Attention mechanism [115].

Long short-term Memory networks

Long short-term memory networks have been developed to overcome the vanishing gradient problem in RNN by replacing the hidden layer of RNN with an LSTM unit which is composed from a cell state, an input gate, a forget gate and an output gate [197].

- The cell state is considered as the memory of the network.
- The forget gate is responsible for deciding what information from the previous time steps is important.
- The input gate decided what information from the current time step is important to add.
- The output is responsible for deciding the output value of the current time step.

A traditional LSTM unit is shown in Figure 7.2 and is composed of a cell with an input gate, output gate and forget gate. At each time step, the gates control which operation is performed by the cell; the internal LSTM equations are defined in equations (7.2)–(7.7):

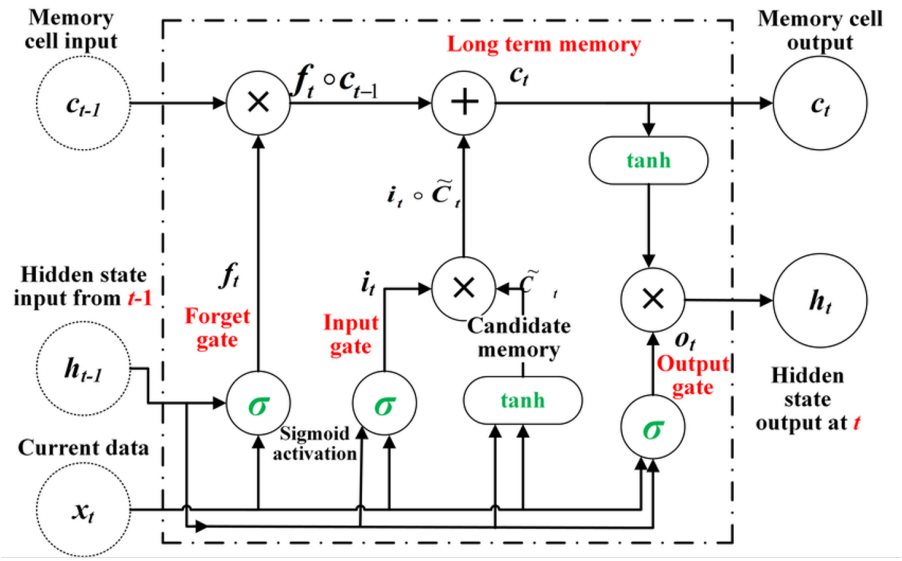


Figure 7.2: Example of an LSTM unit [4]

$$i_{i,j} = \sigma(W_f[h_{t-1}, x_t] + b_{(i)}) \tag{7.2}$$

$$f_{i,j} = \sigma(W_f[h_{t-1}, x_t] + b_{(f)}) \tag{7.3}$$

$$o_{i,j} = \sigma(W_o[h_{t-1}, x_t] + b_{(o)}) \tag{7.4}$$

where i_t represents the input gate, f_t the forget gate and o_t the output gate. Furthermore, σ is the sigmoid function, W_x is the weight vector of the gate, and h_{t-1} is the output of the previous LSTM block. Finally, x_t is the input of the current time step and b_x the biases for the respective gates.

The input gate in equation (7.2) controls what information is going to be stored in the cell state. Similarly, the forget gate controls what information is kept and what information is forgotten from the cell state doing a reset operation. Finally, the output gate is used to provide the activation to the final output of the LSTM block doing a *read operation*. To get the memory vector for the current time step, (c_t), the cell candidate is calculated and defined as $\tilde{c}_t$.

$$\tilde{c}_t = tanh(W_c[h_{t-1}, x_t] + b_{(c)}) \tag{7.5}$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \tag{7.6}$$

$$h_t = o_t * tanh(c_t) \tag{7.7}$$

Equation 7.6 shows that at any time step, our cell state knows what it needs to forget from the previous state and what it needs to consider from the current time step.

7.2.1 Encoder-Decoder Architecture

Expanding the idea of a classical LSTM, encoder–decoder architecture for time series prediction has been introduced [198]. Encoder–decoder architectures have mainly been used for Seq2Seq tasks in natural language processing (NLP). At the same time, multi-step time series forecasting can also be treated as a Seq2Seq task, for which the encoder–decoder model can be used. An encoder–decoder LSTM is a model that consists of two sub-models: a model called the encoder that reads the input sequences and compresses them to a fixed-length internal representation and an output model called the decoder that interprets the internal representation and uses it to predict the output sequence as presented in Figure 7.3. The encoder processes an input sequence, x and encodes the entire sequence to a context vector. The context vector, c_T, which is then passed to the decoder, includes the hidden state produced by the encoder and contains all the encoded information from the previous hidden representations and previous inputs.

$$c_T = f(Wh(t-1) + Ux(t)) \tag{7.8}$$

where f is the chosen activation function of the LSTM unit, W the weight of the hidden state of the encoder. Similar to the encoder, the decoder could be composed of several LSTM units.

The decoding phase is initialised with a time step and a dummy input, s_{init}. Each LSTM unit receives a hidden state, $S(t-1)$, from the previous unit and produces an output, $y^{(t)}$, as well as its own hidden state, $s(t)$.

$$s_T = f(Wh(t-1)) \tag{7.9}$$

Finally, the output of the decoder, $y^{(t)}$, is generated by applying the affine transformation followed by the function that suits the specific tasks (e.g. a softmax function for a classification

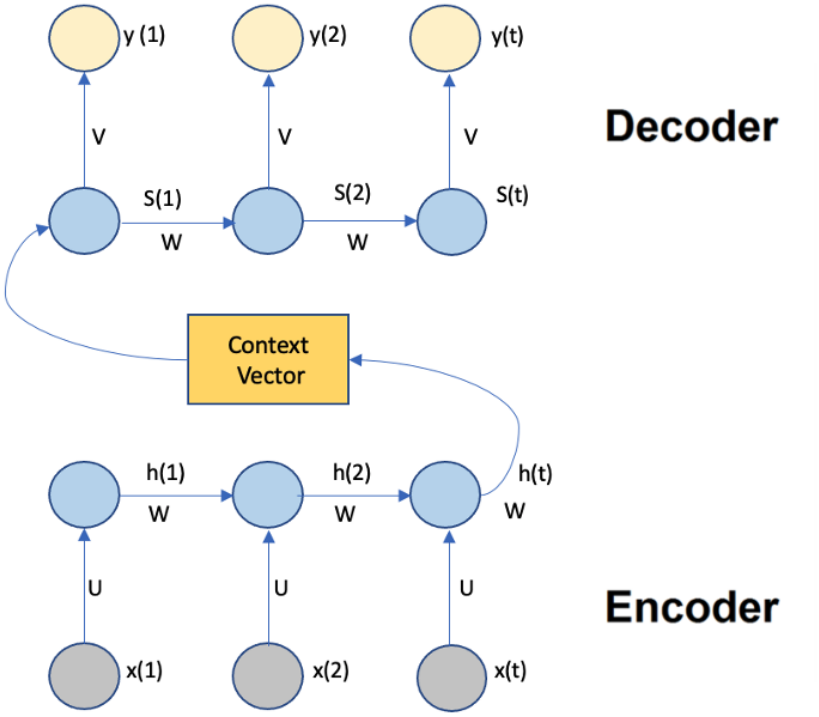


Figure 7.3: Encoder-Decoder Architecture

task or tanh for a regression task).

$$y(t) = f(Vs(t)) \tag{7.10}$$

This encoder–decoder architecture for LSTM has proven to be effective for small length sequences, but when the sequence length increases, it is hard for the model to compress a long sequence into a single context vector without forgetting information. This explains why studies have shown that the performance of this model drops as the size of the input sequence increases [199].

The Attention mechanism has been introduced to resolve this problem that arises in the encoder–decoder architecture of recurrent neural networks. The concept of the attention is that instead of attempting to learn a single vector representation, which the architecture of Figure 7.3 does, it keeps around context vectors for each input sequence and references these vectors at the decoding state. Therefore, the input can be expressed in an optimal manner since the Attention mechanism pays greater attention to certain factors when processing the data [199] [200]. As mentioned, in using the Attention mechanism, what mainly changes compared to a standard encoder–decoder model is that a different context vector is computed at each time

step as the input to the decoder. At first, in every combination of time step j of the encoder and time step t of the decoder, alignment scores, $\beta(j,t)$, are computed with the following weighted sum:

$$\beta(j,t) = V_a tanh(U_a s(t-1) + W_a h(j)) \tag{7.11}$$

where V_a are the weights from the decoder, U_a the weights of the encoder and W_a the weights of the hidden state of the encoder. Then, in every time step, $\beta(j,t)$ is normalised using the softmax function, and the attention weight ($\alpha(j,t)$) is defined as:

$$a(j,t) = \frac{exp(\beta(j,t))}{\sum_{j=1}^{t} exp(\beta(j,t))} \tag{7.12}$$

Finally, the new context vector, c(t), is formed using the attention weights and the hidden state from the encoder:

$$\sum_{j=1}^{t} a(j,t)h(j) \tag{7.13}$$

Following the above process at every time step, we can select the relevant information from a sequence of players' actions, update the input feature and the hidden state of the encoder successively and generate the most relevant short-term features [200].

7.3 Money Laundering Risk in Online Gambling

The AD framework in this chapter aims to identify potential threads and risks in customers' behaviour across two parameters: a) betting risk and b) payment risk. The risk assessment that was published by the Gambling Anti-Money Laundering Group (GAMLG) lists the main risk factors that remote gambling operators should focus on for improving their AML procedures [127]. Also, the GAMLG has provided guidance on a range of customer, product, payment and employee risk areas that should be assessed, e.g. withdrawing without play. These are essential for the small operators who often struggle to be able to spend as much as larger operators on compliance research and development. Using the insights from the GAMLG in Table 7.1,

Table 7.1: Behavioural analysis for AML detection. We define suspicious flags that could occur in online gambling and increase the risk of money laundering. Our system tries to implement AD in monitoring granular-level player data.

Domain	**Type**	**Description**
Deposit Threshold	Absolute	Flags whether a deposit exceeds an absolute threshold.
Spend Threshold	Absolute	Flags whether spend exceeds an absolute threshold.
Near Threshold	Absolute	Flags whether a player reaches within a set% of the limit/threshold for Deposit and Spend.
Deposit/Spend Withdrawal Ratio	Ratio	Spots player who have a very low withdrawal ratio.
Deposit Threshold Ratio	Ratio	Spot players who deposit just below threshold.
Spend Threshold Ratio	Ratio	Spot players who spends just below threshold to avoid being flagged.
Suspicious Play Check	Machine Learning	Based on Patterns of Known money launderers, however predicting very rare events is very hard due to the lack of data.
Anomaly Check (Player)	Machine Learning	Spot Anomalies based on what is considered normal.
Anomaly Check Branch	Machine Learning	Spot Anomalies based on what is considered normal in a branch, game or game type
Affordability Check	Machine Learning	Using internal and external data to derive affordability

we present a list of different flags that could be observed in online gambling and could be potentially linked to money laundering risk [7].

Our AD system tries to exploit anomalies in terms of anomaly checks wherein the system flags events that are considered abnormal in relation to a player's activity. Suspicious play checks are considered to be addressed with the supervised learning approaches we investigated in the previous chapters. Finally, an affordability analysis could have been very useful in for evaluating the activity in online gambling, but related data were not available.

7.4 Anomaly Detection Framework

In this section, we introduce the hybrid LSTM-ATT and Gaussian estimation framework for AD. The framework is a two-step process wherein, at the start, we predict a player's next

action in terms of its type and amount and then investigate whether the particular action is an anomaly. Initially, raw features are extracted for each player, including the amount, type and results of the executed action together with the balance after the action is completed and the time from the last action. Then, an encoder–decoder LSTM-based framework is proposed to predict the individual's next action amount and type. In the encoder step, the LSTM and the Attention mechanism are implemented to get the most relevant features and encode them into a context vector. Then, in the decoder step, the LSTM decoder decodes the most relevant features to predict the amount of a player's next action (regression task) and the type of the next action (classification task). Using a similar approach as [118], classification and regression reconstruction errors from the output of the LSTM-ATT are then used to fit a multivariate Gaussian distribution in order to find any abnormal behaviours. During the training phase of the system, it is assumed that the training data include only normal behavioural patterns.

The proposed hybrid framework of Figure 7.4 consists of two major components: a) an LSTM-ATT network for time series prediction and b) a Gaussian estimation step for AD. The first component aims to achieve adaptive learning on the temporal dependency features of the multivariate time series data and assist in the identification of anomalies in players' behaviour through its prediction. The first LSTM cell is used to encode the hidden representations of the time series as the temporal context vector. The Attention mechanism is used to select relevant encoder hidden states across all time steps with more accuracy so as to improve our model's representation ability of dynamic multivariate time series data. In this way, the Attention mechanism assigns different importance to the different elements of the input sequence and gives more attention to the more relevant inputs. The other LSTM cell is used to decode the hidden representation for predicting both the amount and type of a player's actions. Through this end-to-end process, hidden long-term dependent features and non-linear correlation features can be learned from the raw multivariate data.

In contrast with more traditional approaches where neural networks focus on a single task, our LSTM-ATT tries to optimise both the a) regression task of the action amount prediction and b) the classification task of the action type prediction through hard parameter sharing. Hard parameter sharing is defined as the method of sharing the hidden layers between all tasks; it

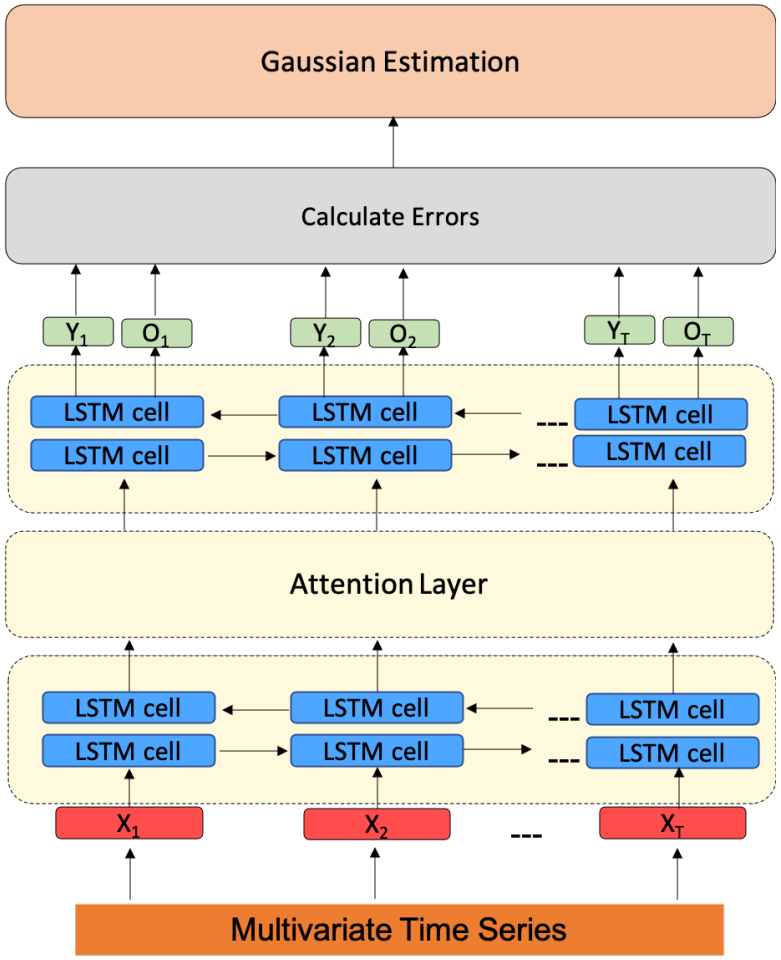


Figure 7.4: LSTM-ATT framework for multi-task time series prediction and anomaly detection. Y_T represents the classification task and O_T the regression task.

reduces the risk of overfitting [201]. Multi-task learning (MTL) studies have shown that by sharing information between related tasks, we allow the model to better generalise the original task [202]. Multi-task learning has been very popular recently, with a lot of success in different areas of machine learning, from NLP [203] to speech recognition [204].

Subsequently, the prediction errors from the LSTM-ATT are modelled using a Gaussian distribution. The mean and variance of the distribution are estimated and fit a multivariate Gaussian. Finally, the output of the framework is a probability, p, that indicates the likelihood of a specific action being an anomaly.

7.5 Gambling Raw Dataset

In contrast to the experiments that were carried out in the previous chapters of the thesis wherein we used the aggregated dataset we built in Chapter 3, for the AD task a granular level (every bet, transaction are recorded in a sequential form), a gambling dataset was provided by Kindred group. Due to constraints on data sharing, our research partners provided us data with only the period from December 2018 to the end of January 2019. The dataset contained every betting and transaction record of every player that played in the two-month interval between 1 December 2018 and January 2019 in Unibet. The dataset contained 10 columns, including the player ID, the action type, the action reference, the bonus and cash balance after the action, the timestamp of the action and the amount of the related action.

To capture the short-term and long-term relationships of the gambling activities, the following features were included in the features' space: a) The action that a player has executed, b) The amount of the action and c) the wallet balance after each event was added as well. To give emphasis in the action type, d) the outcome of each action was recorded (amount and result) as Table 7.2 shows. For example, when a player wins a big bet, he might withdrawal his winnings. e) Another feature was the time difference between consecutive events. This could help the model distinguish between continuous events, meaning where the events were only a few seconds apart or isolated events.

As mentioned in Section 7.4, the problem was formulated into a sequence prediction problem and could be defined by $S = (T_1, T_2, ..., T_l)$, where T_i is a vector of the time series and represents a temporal vector and l is the length of each sequence. Thus, two supervised temporal sequences were defined for the two outputs of our multi-task framework as $D_1 = (S_i, Y_i)_{i=1}^{n}$ and $D_2 = (S_i, V_i)_{i=1}^{n}$, where Y_i is the output of the regression task and V_i the label of the classification output. The final dataset was processed into slices of smaller sequences; a snapshot of the data can be seen in Table 7.2. A player sequence included the action type, action amount, action result, wallet balance and time from the last action.

In a multivariate time series, the values of the variables can be on a different scale. To reduce

Table 7.2: Data sequence of actions of a player's activity

Action Type	Action Amount	Action Result	Amount Result	Balance	Time Diff
Stake	10	Win	18	18	0
Deposit	5	Success	5	23	5 min
Stake	5	Lose	-5	18	20 min
Withdraw	5	Success	-5	13	5 sec
Stake	5	Win	25	33	30 min

any bias that could affect our prediction model, we pre-processed the data with the maximum–minimum normalisation in both the training and testing sets, similar to our previous experiments. One important thing that was highlighted in the granular dataset was the imbalance between action types, where betting actions were dominating the rest (deposits and withdrawals) in the standard player time series. Similar to other imbalance problems, this would negatively affect the performance of the prediction task. Since resampling the minority class would not preserve the structure of the time series, we penalised the loss function of our algorithm using specified weights for each class.

$$w_i = \frac{N_m}{N_i} \tag{7.14}$$

where N_m is the total population of the majority class and N_i is the total population of the class for which we calculated the weight. This means that the majority class has $w_m = 1$.

7.6 Experiments

In this section, we evaluate the proposed framework on five different players' datasets, where each player's dataset represents a multivariate time series. Since the proposed AD framework is a two-step process, i.e. time series forecasting and AD, it requires a two-step evaluation. a) In step one, we evaluate the forecasting capabilities of the LSTM-ATT network in terms of both regression and classification tasks. The LSTM, GRU and standard LSTM encoder–decoder models are implemented for comparison. b) In step two, we evaluate the anomalies generated by the system with the assistance of the compliance team of Kindred; their testimonies were

recorded and are presented in Section 7.8.1.

All the players we included in the case studies presented in this section could be part of four different groups of players (we mainly focused on false positives and false negatives) that were generated from the results produced by SSGAN-c in Chapter 6. This will allow us to investigate further the behaviour of those specific players:

- True positives from SSGAN-c – the players who have an IRR report and whom our SSGAN framework classified correctly.
- True negatives from SSGAN-c – the normal players who were correctly identified by our framework.
- False positives from SSGAN-c – this category describes the normal players our system classified as high-risk for money laundering. However, those cases did not have an IRR.
- False negatives from SSGAN-c – this category describes the high-risk players with IRR report, that our system missed and categorised as normal.

We split the players' datasets in chronological order into training validation and testing (hold-out) sets (60% used for training and 40% for testing). This way, the model could simulate a real-world situation wherein it could only see past events to predict future events. The open-source machine learning library Scikit-learn and deep learning framework PyTorch were used to implement the benchmark methods and our proposed framework. There were five input nodes in the neural network of the system, and each took input from the features in Table 7.2: i) event type, ii) event amount, iii) result of the event, iv) wallet balance and v) time from the last event. At the other end of the neural network, there were two output nodes for the classification (event type) and regression (amount of action). The classification output node used a 'softmax' activation function, whereas the regression node used a 'tanh' activation function. The model was trained with the loss function of categorical cross-entropy for classification and mean squared error for regression.

7.6.1 Comparison Models and Evaluation Metrics

The LSTM, GRU and LSTM encoder-decoder (Seq2Seq) models were implemented for comparison. These models can be described as follows:

1. LSTM is an extension of RNN, which has been used to solve sequential problems.
2. GRU is an extension of RNN, with a simpler structure compared to LSTM.
3. Seq2Seq is a classic sequence-to-sequence model (encoder–decoder architecture).

Information regarding the hyperparameter settings of these methods is provided in the following section. We used the root mean square error (RMSE) and mean absolute error (MAE) as the model error evaluation metrics for the regression task and the classification accuracy of each class (action type) for the classification task. Finding the absolute value was important because it did not allow for any form of cancellation of the error values. However, sometimes, large error values could occur, which could change drastically the final results. For this reason, the RMSE was taken into consideration. Specifically, assuming N was the total number of data points, $\hat{x_t}$ and x_t denoted the predicted value and the true value, respectively, at time t, and we could calculate the MAE and RMSE with equations (7.15) and (7.16), respectively:

$$MAE = \frac{1}{N}\sum_{i}^{n}|x_i - \hat{x_i}| \tag{7.15}$$

$$RMSE = \sqrt{\frac{\sum_i^n (x_i - \hat{x_i})^2}{N}} \tag{7.16}$$

7.6.2 Hyperparameter Settings

Before training any neural network model, it is essential to set appropriate hyperparameter values. These parameters cannot be inferred while training the model, as they correspond to the model selection task and influence the speed of the learning process. Hyperparameters pertaining to the model selection task include the architecture and size of the network. In

addition, the mini-batch size and drop-out and learning rates are some of the hyperparameters that affect the speed and quality of the learning process. After testing and examining different hyperparameters combinations for the proposed architecture, meaning different number of neurons, different number of layers, different learning rate values the hyperparameters of Table 7.3 have been chosen since they produced the best results.

In terms of the model parameter settings, we tried to keep the baseline models consistent so that the same configurations were shared in both the baseline and our model. For the baseline deep learning models (e.g. GRU and LSTM), we set the parameters of each model as in Table 7.3. We selected one hidden layer with 100 neural units; the MSE was chosen as the regression loss function and the cross-entropy as the classification function. Furthermore, the Adam function was chosen as the model optimiser. The batch size was set to 64. The training process was repeated for 100 epochs. For the Seq2Seq model and our model, we used an LSTM as a hidden layer of the encoder and another LSTM as the hidden layer of the decoder.

Table 7.3: Hyperparameters settings for sequential models

Method	**Parameter**	**Value**
Baseline Models	Hidden Layers	1
	Hidden Layers Units	32
	Batch Size	64
	Loss Function Regression	Mean Squared Error
	Loss Function Classification	Cross Entropy
	Learning Rate	0.001
	Training Epochs	100
LSTM-ATT and Seq2Seq	Default hidden Layers (encoder)	1
	Default hidden Layers (decoder)	1
	Hidden Layers Units	32
	Batch Size	64
	Function of Attention layer (LSTM-ATT)	softmax
	Loss Function Regression	Mean Squared Error
	Loss Function Classification	Cross Entropy
	Learning Rate	0.001
	Training Epochs	100

7.7 Results of Time Series Forecasting

In this section, we summarise the results of the experiments that analysed the forecasting performance of our model and the baseline techniques. We conducted the experiments with the experimental settings in Section 7.6; we have illustrated the results of the regression task in Table 7.4 and the results of the classification task in Table 7.5. The results show the average performance for different sequence lengths in the range of 2–20. It is clear that our model outperformed the other models regarding the regression task of predicting the amount of a player's next action. The LSTM-ATT was observed to have the smallest MAE and RMSE on three out of the five players' datasets, which was an indication of a robust performance over different players' behaviours. This suggests higher confidence in our method regarding predicting the amount of the next action of a player. Further analysis showed that the RMSE and MAE of the baseline deep neural network models based on LSTM and GRU were similar in most cases, with similar performance. However, the Seq2Seq model results indicated an inability to capture the temporal behaviour of a player and predict with high accuracy the next action's amount: in four out of the five examples, Seq2Seq achieved the worst performance with the lowest MAE.

Evaluating the performance of the models on the classification task of predicting the correct type of the next action, we observed that the standard Seq2Seq model had better classification accuracy in predicting the next bet, showing a much better performance in identifying when the next deposit would happen. Overall, all four models had good performance in predicting bets, with the average accuracy across the five players equal to 90%. Meanwhile, for predicting withdrawal events, there was not a clear best model since, in different instances, a different model outperformed the others. In terms of predicting deposits, LSTM-ATT achieved the highest accuracy in three out of 5 players. It was noticeable that all the models found it hard to predict when the next withdrawal would occur, with some cases not predicting any withdrawals, i.e. PLAYER 2 and PLAYER 3. However, the main reason that 0% accuracy was observed for withdrawals was that there were cases where we had a very small number of withdrawals in the testing set. Overall, our approach showed the best overall performance in

Table 7.4: Regression task results of our approach and the baseline models for five players.

MODELS	AVERAGE MODEL PREDICTION ERROR FOR DIFFERENT PLAYERS									
	PLAYER 1		PLAYER 2		PLAYER 3		PLAYER 4		PLAYER 5	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
LSTM-ATT	29.27	66.91	**2.70**	**6.79**	**1.74**	**6.74**	4.11	6.64	**2.87**	**7.21**
Seq2Seq	32.75	79.11	10.04	13.63	10.99	14.57	3.55	6.44	20.15	26.65
LSTM	29.63	66.35	2.99	7.32	2.41	8.71	**3.47**	**6.00**	4.55	13.01
GRU	**26.88**	**65.85**	3.72	7.79	3.26	10.93	3.53	6.06	3.63	11.00

Table 7.5: Classification task prediction results on five players.

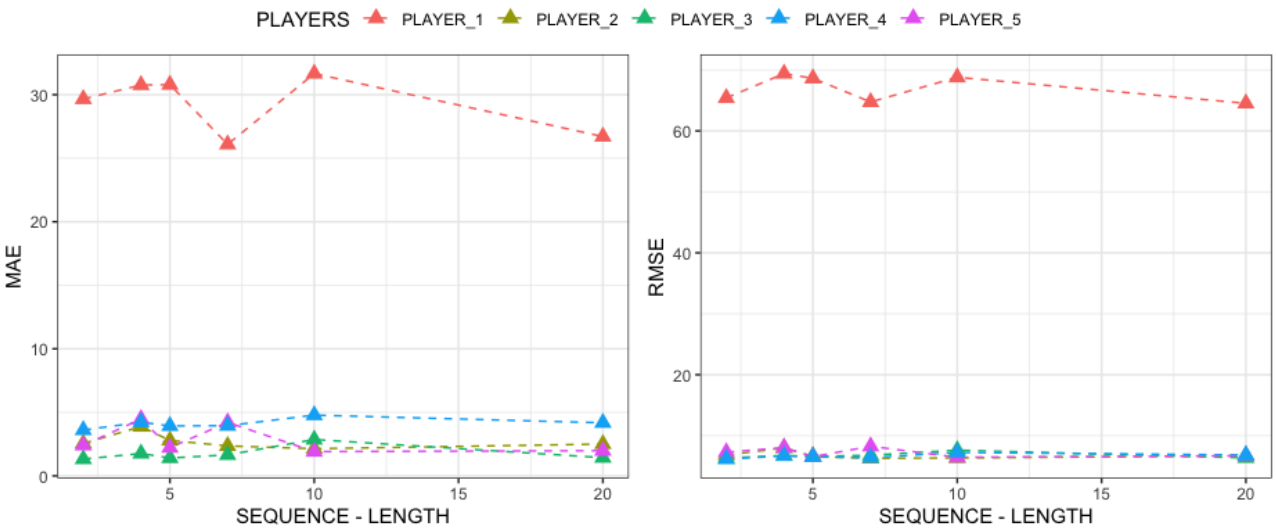
MODELS	AVERAGE CLASSIFICATION ERROR														
	PLAYER 1			PLAYER 2			PLAYER 3			PLAYER 4			PLAYER 5		
	BET	DEP	WITH	BET	DEP	WITH	BET	DEP	WITH	BET	DEP	WITH	BET	DEP	WITH
LSTM-ATT	0.82	**0.58**	0.50	0.88	0.80	0.00	0.98	**0.98**	0.00	**0.91**	0.50	0.87	0.93	**0.71**	**0.64**
Seq2Seq	**1.00**	0.18	**0.62**	**0.97**	0.49	0.00	**0.99**	0.65	**0.89**	0.69	0.40	0.77	**0.96**	0.44	0.54
LSTM	0.87	0.53	0.33	0.86	0.82	0.00	0.96	0.95	0.00	0.71	**0.52**	0.70	0.93	0.71	0.63
GRU	0.87	0.56	0.33	0.86	**0.86**	0.00	0.98	**0.98**	0.06	0.84	0.40	**0.90**	0.95	0.71	0.61

both prediction tasks and an ability to predict with high confidence the next actions of the players, which is really important for our AD framework. In the next step, based on the output from the forecasting step, we attempt to identify abnormalities in these five players' behaviours.

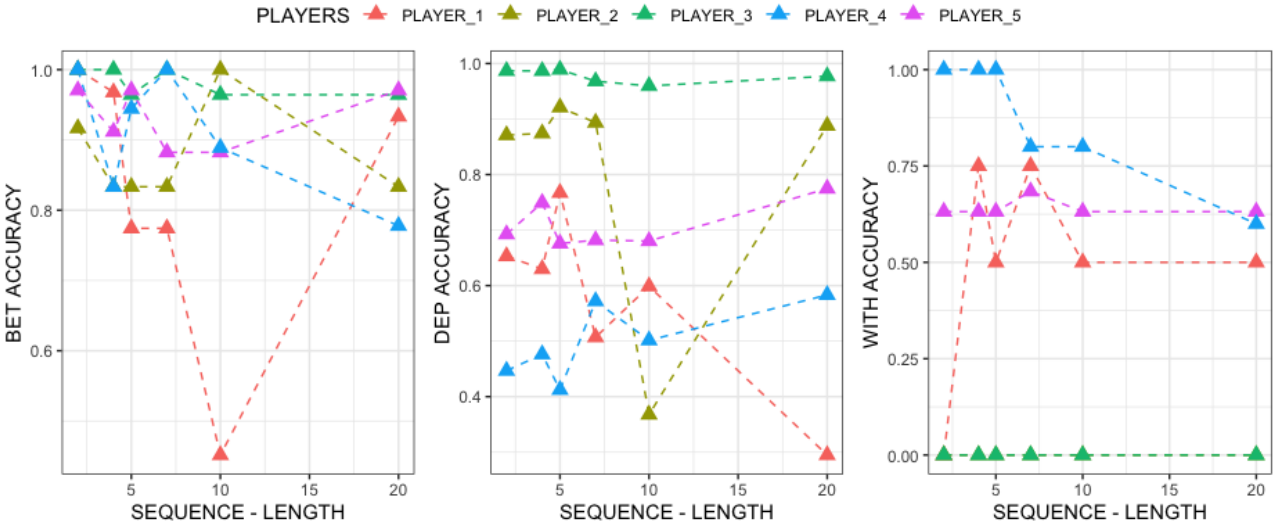
As indicated, the results in Tables 7.4 and 7.5 corresponded to the average performance of the models on the different sequence lengths. In Figure 7.5, we show how the regression errors and the classification accuracy of our model were affected when the length of the input sequence changed. Figure 7.5a illustrates that overall, the regression error remained on the same levels; in some cases, a small increase was noted, while in some other cases, a small decrease was noted. The accuracy plots in Figure 7.5b show more volatility in the performance. The model trained with a sequence length of four was found to have the most balanced performance in terms of both the classification and regression tasks.

7.8 Anomaly Detection in Online Gambling

As part of the research, in Figure 7.4, we proposed an AD system that could be implemented in an online real-time environment for identifying abnormal, potentially illicit activities that



(a) MAE and RMSE of LSTM-ATT when the input sequence length changes



(b) Bet, Deposit and Withdrawal accuracy of of LSTM-ATT when the input sequence length changes

Figure 7.5: Sensitivity analysis of the performance of our model for different sequence length

could be connected with money laundering or any other type of fraud. As depicted in the system, our proposed method involved two stages. The forecasting stage, where we predicted both the amount and type of the next action, and the AD stage. From the predictions, we computed a regression error, $e_r^{(t)}$, for an instance, x^t, as the absolute difference between the actual value and its predicted value, $|Real\ Value - Predicted\ Value|$. Similarly, we computed the classification error, $e_c^{(t)}$, as the difference between the probability of an event being observed (from the prediction model) and the actual label, i.e. $e_c^{(t)} = |[1,0,0] - [0.7,0.3,0]| = [0.3,0.3,0]$.

The prediction error vectors from the training data (assumes only normal instances) were

calculated and they were then used to fit a multivariate Gaussian distribution (assumed to follow a Gaussian distribution) with function $N = N(\mu, \Sigma)$, where μ and Σ are defined by Equation (7.17) and Equation (7.18) respectively:

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x^i \tag{7.17}$$

$$\Sigma = \frac{1}{m}\sum_{i=1}^{m}(x^i - \mu)(x^t - \mu)^T \tag{7.18}$$

where m is the total number of instances, μ is defined as the mean of each feature and Σ is the covariance matrix. These parameters then are used to estimate the probability of an action to be an anomaly $p(x)$. Given a new instance $p(x)$ is calculated as follows:

$$p(x) = \frac{1}{2\pi^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}}\ exp(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)) \tag{7.19}$$

The anomalies would be the data that fall under the low probability areas of the Gaussian, because being in the low probability area, that data is highly unlikely to be distributed in our distribution. An anomaly was flagged if $p(x) < \tau$, where τ is a pre-defined threshold. If $p(x) > \tau$, then the specific instance was considered as normal. The anomalous gambling events predicted could indicate an increased level of money laundering risk under the assumption that normal events share common patterns while anomalies do not. To evaluate the output from the AD framework, money laundering experts reviewed these anomalies in order to identify any potential risks and associations with money laundering. In this research, "anomaly" and "abnormal behaviours" mean actions with unusual data patterns, differing from "risky behaviours". Players with abnormal data patterns are of interest for detection, but they need to be reviewed by domain experts to determine whether they represent any money laundering risks.

Table 7.6 shows some examples of how an anomaly can be categorised and describes the characteristics of each category. The first category describes an anomaly that flags events where the action amount deviates a lot from the regular patterns of a player. A large amount is defined as an amount that deviates from the normal pattern amount. This type of anomaly

Table 7.6: Anomaly description: we define different anomalies' categories. Three different categories have been set to describe the type of anomaly that our system produces. These types are meant to assist the compliance department in evaluating the flags that our system generates.

Category	Key Characteristics	Explanation
Larger amount than expected	More than three standard deviations than usual	Unusual event. May be suspicious
Wallet withdrawal immediately after a deposit without placing a bet	Withdrawal the full balance normally do not follow a deposit.	Possible suspicious behaviour compared to usual play.
Wrong event prediction i.e. Bet instead of deposit	Series of losses reduces balance below recent bet rate. This would normally lead to a deposit.	The player changes strategy which could results to something suspicious.

could implicate suspicious behaviour. The second category corresponds to strange actions, e.g. a withdrawal after a deposit. This type of anomaly is highly correlated with money laundering. In the final category, the anomaly type describes cases when a player's action does not match with their usual pattern of play.

7.8.1 Anomaly Detection Case Studies

Assessing AD performance is not straightforward; as discussed in previous works [205], there exists no established benchmark comprising a set of well-defined scenarios that are considered anomalous by domain experts. In fact, it could be argued that assessing AD performance in this way is not appropriate, as we are biased towards evaluating the system's ability to detect a particular class of anomalous situations from a goal-driven perspective, which stands in conflict with our definition of an anomaly as something previously unknown/ill-defined [205].

Since there were no ground truth labels on the players' actions to optimise our threshold, τ, according to the output probability from equation (7.19), we defined five anomaly levels:

1. Highly Likely: if $p < 0.0001$.

2. Likely: if $p > 0.0001$ *and* $p < 0.001$.

3. Possible: if $p > 0.001$ *and* $p < 0.01$.

4. Less Likely: if $p > 0.01$ *and* $p < 0.1$

5. Highly Unlikely: if $p > 0.1$ *and* $p < 0.25$

The above levels were defined empirically after experimenting with different thresholds. A balance needed to be established so our anomaly detection system wasn't very sensitive to anomalies but at the same time is able to capture abnormal behaviours. The above levels mean the smaller the probability, the higher the probability of a particular flag to be an anomaly. We specified these different levels in order to define different levels of importance.

7.8.2 Kindred Evaluation

In this section, we present the case studies that Kindred's risk team evaluated our system on. Feedback was provided on whether the AD system that we built could be a part of the money laundering detection framework. Due to the limited time that our research partners had available, only five players were reviewed. Each individual case is presented here, and initially, a brief history of each player's betting activity is provided. Then, we analyse the anomalies from the AD system, and the feedback that was provided is presented.

Case Study - Player 1

Player 1 was a false negative prediction from our SSGAN-c framework. Figure 7.6a shows the rolling mean of the actual amount that the player has deposited, withdrawal or used to bet together with the rolling mean of the predicted amount where Figure 7.6b shows the wallet balance during the same period. Consistently, the customer wager an amount below £100, which is on average 12% of the total wallet balance. Table 7.7 presents the anomalies that our AD system detected (all possible anomalies).

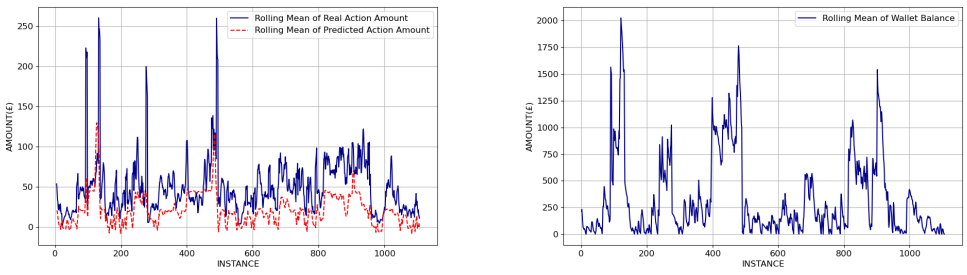
Table 7.7: Player 1 anomalies detected from the anomaly detection system

Date	Amount(£)	Action Type	Action Predicted	Predicted Amount
$2019-01-05\ 16:05:21$	1,000	Withdrawal	Stake	40,09
$2019-01-05\ 16:11:37$	1,000	Withdrawal	Withdrawal	71.13
$2019-01-11\ 22:15:13$	800	Withdrawal	Stake	35.15
$2019-01-13\ 00:54:42$	1,000	Withdrawal	Withdrawal	63.96

1. 1^{st} Anomaly Description: Following a losing a streak of bets worth between £20 and £70, and with wallet balanced of £0.1 the customer deposited £180. Then the customer managed to win a few small bets to take his balance to £400, where at this point he started losing again to force his balance below £100. Finally, the customer managed to win a bet around £1,400 which followed by a series of losing bets until he finally executed a withdrawal of £1,000.

2. 2^{nd} Anomaly Description: The second anomaly observed 6 minutes after the first anomaly. The customer again executed a series of small bets and took his total balance around £2,000. Similarly, to the first anomaly, after his big win the player performed in a short time interval a few more losing bets in order to take his wallet balance just below £1,500, when finally he performed a withdrawal that was flagged by the AD system.

3. 3^{rd} Anomaly Description: On the 2019-01-11 at 21:33:03 pm the customer deposited £250. After, a series of unsuccessful bets, he managed to win a bet around £500. Then he kept playing placing small amount of bets where he lost some money. The he finally won again to take his balance above £1,000. Following the winning bet, he executed a



(a) Rolling Average of the Amount of actual actions and predicted actions

(b) Rolling Average of the Wallet Balance

Figure 7.6: Information of Player 1

few more losing bets (£500-600). Finally, at the end he executed the withdrawal worth £800 leaving his wallet balance with £200.

4. 4^{th} Anomaly Description: The customer performed a few successful bets that took his wallet balance just over £1,500. Subsequently the customer, following the same pattern that led to the previous 3 anomalies he lost a series of small wager bets when finally he decided to withdrawal all his money.

Using the above information, Kindred's Player Risk Officer reviewed the behaviour of Player 1 and was asked to answer the following questions related to the anomalies produced by the AD system where his answers are inside quote-marks:

- Would you consider the behaviour of the player as high-risk for money laundering (a lot of bets below £100, deposits around £200 and withdrawals around £1000) or would you consider the behaviour of the player as a responsible gambling case or do you think the above behaviour could be described as normal?

  "Customers who consistently wager minimal amounts or amounts that are disproportionate to their available balance would be considered higher risk for AML purposes."

- Would you consider the above flags as anomalies or regular withdrawals for this type of players? If these flags are treated as anomalies, do you believe these anomalies can help you detect money laundering?

  "I think that the withdrawal amounts are reflective of the volatility of the product the customer was using (roulette). A customer could place a bet of £100 on roulette which could potentially see them return £3,700. I think slightly less weight should be given to the disparity between bet size and withdrawal size depending upon the product used"

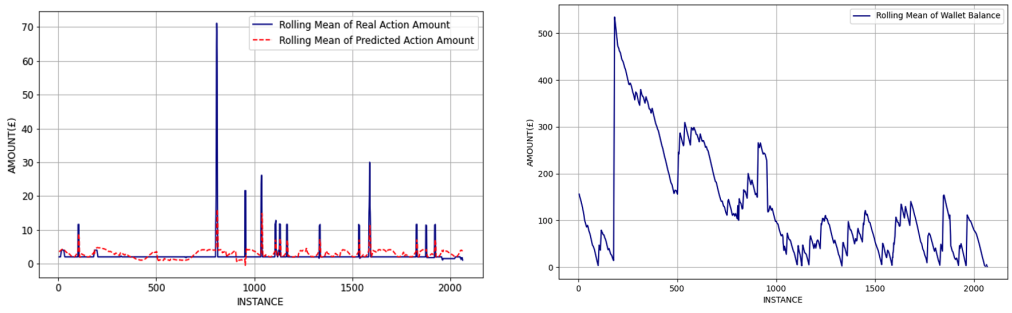
- Would you want to examine the player further? If yes at which point i.e. instantly or after a few more anomalies?

  "Yes, but likely after a few more anomalies."

Case Study - Player 2

Player 2 was a false positive prediction from our SSGAN-c framework. Figure 7.7a shows the rolling mean of the actual amounts that the player has deposited, withdrawal or used to bet together with the rolling mean of the predicted amount. Figure 7.7b shows the real wallet balance. Even though this player does not have an IRR flagged the rule-based system flagged this player on the 27-Mar-19 and on the 20-May of 2019. According to the Figures 7.7a and 7.7b, we can see the player deposits and withdrawals under £100 most of the time. However, we can see that in 2 months period the player has more than 2,000 events.

In total five anomalies have been flagged for Player 2 as Table 7.8 shows (4 out 5 have be categorised as high unlikely anomalies while on the first anomaly have been classified as highly likely by our detection system). Player was registered on Unibet on 11/11/18. During the month of December, he only performs a few small transactions and bets all below £100 from the 02/12/2019 till the 25/01/2019. From the 25 of January until the 31 of January the player was very active with more than 4,000 actions. As we can see from Figure 7.7a the player bets around £2 on average in most of his bets.



(a) Rolling Average of the Amount of actual actions and predicted actions

(b) Rolling Average of the Wallet Balance

Figure 7.7: Information of Player 2

1. 1^{st} Anomaly: Observed on the 27^{th} of January where the player bet £135 which is highly unusual for his pattern of play. Prior to this bet the player was betting around £2 per bet and his wallet balance was around £100. He suddenly made a bet of £50 where he won £74. Then he bet all £135 on his wallet and lost all his money.

Table 7.8: Player 2 anomalies detected from the anomaly detection system

Date	Amount(£)	Action Type	Action Predicted	Predicted Amount
$2019-01-27\ 18:08$	135	Stake	Stake	0.21
$2019-01-27\ 18:53$	83	Deposit	Stake	0.40
$2019-01-27\ 18:54$	83	Stake	Stake	1.57
$2019-01-28\ 10:03$	100	Withdrawal	Stake	0.64
$2019-01-28\ 20:30$	50	Deposit	Stake	0.65

2. 2^{nd} Anomaly: The player deposited £83 after he lost all his money when anomaly after the instance of the 1^{st} anomaly.

3. 3^{rd} Anomaly: The player bet all the £83 he just deposited which is highly unusual with his pattern of play which he makes bet of £2. The player managed to almost double his money. Then he continues to bet £2 and losing money.

4. 4^{th} Anomaly: Observed at a period where the customer's wallet balance was around £220. The individual was being very active and he was betting small amount of £2. After the withdrawal his wallet balance was £120. Then he continues with small bets of £2.

5. 5^{th} Anomaly: The player executed a series of losing bets £2 (casino bet) where at the end he ended up with £61. Then he bet and lost all his money(sports bet) and immediately deposited £50.

Using the above information, Kindred's Player Risk Officer reviewed the behaviour of Player 2 and was asked to answer the following questions related to the anomalies produced by the AD system where his answers are inside quote-marks:

- Would you consider the behaviour of the player as high-risk for money laundering or would you consider the behaviour of the player as a responsible gambling case or do you think the above behaviour could be described as normal?

  "In this instance, his activity could be viewed as a potential responsible gambling (RG) case. Considering the longer activity and turnover generated by the customer the likelihood of money laundering diminishes. Individuals attempting to launder funds will generally try to minimise their losses and are only willing to lose a fixed percentage of their investment.

A progressive anomaly weight approach would be a good indicator in this regard. The other way, meaning the increase in activity and investment made would be a good indicator for a developing RG case."

- Would you consider the above flags as anomalies or regular event for this type of players? If these flags are treated as anomalies, do you believe these anomalies can help you detect money laundering?

  "The indicators could be helpful in a slightly later stage as the customer only really started his activity on the 25/01, unless high deposit values are involved. Note: 'A progressive weight approach would be helpful to distinguish between a latent AML or RG case."

- Would you want to examine the player further? If yes at which point i.e. instantly or after a few more anomalies?

  "Yes, but likely after a few more anomalies."

Case Study - Player 3

Player 3 was a false positive prediction from our SSGAN-c framework. Figure 7.8a shows the rolling mean of the actual amount that the player has deposited, withdrawal or used to bet together with the rolling mean of the predicted amount. Figure 7.8b shows the real wallet balance. Even though this player does not have an IRR flag, the rule-based system flagged this player on the 2019-01-12 and on the 2019-03-10.

In total three anomalies have been detected for Player 3 (all highly likely). The player registered on Unibet on the 25/04/18. In the period of 2 months from December of 2018 until the January of 2019 the player had over 20,000 bets, 69 deposits and only 3 withdrawals. He switched his gambling activity between different products Casino Playngo, Casino Arena and Casino Relax.

1. 1^{st} Anomaly: A withdrawal of £220 has been flagged as an unusual event. The player was betting a very small amounts of £0.3 per bet at Casino Playngo. After he lost all his money he deposited £20. Immediately after the deposit, he won £68 betting at Casino

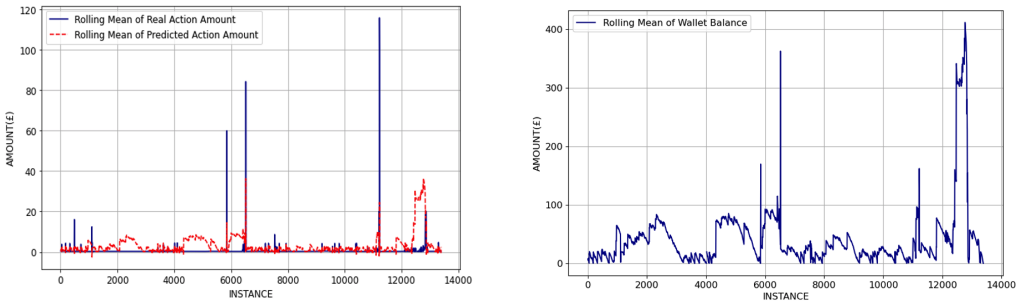
Arena. Then he continued playing at Casino Arena where he was betting £20 per bet. He took his account balance a bit over £220. Then he performed a withdrawal and left his account with £28, where he continued betting an amount £0.20 on Casino Relax.

2. 2^{nd} Anomaly: The player after his first withdrawal continues betting on casino games and betting less that £1 in every bet. After he won a big bet, he withdrawal £420.

3. 3^{rd} Anomaly: The third anomaly observed a few days later. The player deposited £20 at 17:00. He lost all £20 on bets of £0.2. Then he deposited again £20. Finally, he deposited another £20, and kept betting on Casino arena where he won £500 and withdrawals immediately. Then he kept executing bets of £0.2.

Using the above information, Kindred's Player Risk Officer reviewed the behaviour of Player 3 and was asked to answer the following questions related to the anomalies produced by the AD system where his answers are inside quote-marks:

- Would you consider the behaviour of the player as high-risk for money laundering or would you consider the behaviour of the player as a responsible gambling case or do you think the above behaviour could be described as normal?



(a) Rolling Average of the Amount of actual actions and predicted actions

(b) Rolling Average of the Wallet Balance

Figure 7.8: Information of Player 3

Table 7.9: Player 3 anomalies detected from the anomaly detection system

Date	Amount(£)	Action Type	Action Predicted	Predicted Amount
$2019-01-25\ 18:08$	220	Withdrawal	Stake	3.49
$2019-01-25\ 18:53$	420	Withdrawal	Stake	6.48
$2019-01-28\ 10:03$	500	Withdrawal	Stake	1.19

"In this instance, his activity could be viewed as normal. He initially invested a total of £70.00 in five deposits and made some winnings that resulted in a total of 2 withdrawals. He than reinvested a portion of his earnings which explains the increase on 25/01. Same behaviour can be found on 28/01. Note: While the increase in activity can be viewed as normal, his overall activity was somewhat concerning from an RG perspective due to his regular longer evening and night-sessions."

- Would you consider the above flags as anomalies or regular event for this type of players? If these flags are treated as anomalies, do you believe these anomalies can help you detect money laundering?

  "The withdrawal amounts reflect the volatility of the products used by the customer (slots). A customer could win in slots at any given time."

- Would you want to examine the player further? If yes at which point i.e. instantly or after a few more anomalies?

  "Yes, but after a few more anomalies."

Case Study - Player 4

Player 4 was a false negative meaning that the our framework was unable to detect and classified this player as high-risk. The player executed more than 1,000 actions in the month of January. On average he deposited £19 per transaction in total of £1,000. On average he bets £3.5 per bet in a range of 0.1 to £78.4. No detection flags have been raised for this particular player.

1. 1^{st} Anomaly: Initially, the player deposited £50 and then he executed a series of either £5

Table 7.10: Player 4 anomalies detected from the anomaly detection system

Date	Amount(£)	Action Type	Action Predicted	Predicted Amount
$2019-01-17\ 02:29$	220	Deposit	Deposit	3.49
$2019-01-17\ 03:53$	420	Stake	Withdrawal	6.48
$2019-01-17\ 03:54$	500	Deposit	Deposit	1.19
$2019-01-25\ 02:12$	500	Deposit	Deposit	1.19

or £10 bets where he was winning or losing £5 or £10. After around 20 bets he decided to withdrawal £50 (same amount as he deposited).

2. $2^{nd} - 3^{rd}$ Anomaly: Prior to the flagged event, the player performed a series of bets between £5 - £20. Then the player after winning a bet of £40 he took his wallet balance to £78.4. Then he decided to bet all his money and emptied his wallet. The player lost the bet and deposited £50.

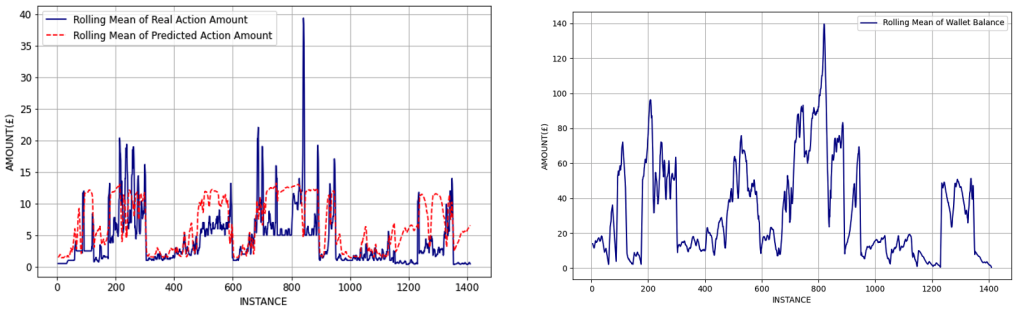
3. 4^{th} Anomaly: The player deposited £50 immediately after he had nothing left on his wallet. Then he started executing very small amount bets prior to the flagged event. Even in his winning bets the amount that he won was insignificant.

Using the above information, Kindred's Player Risk Officer reviewed the behaviour of Player 4 and was asked to answer the following questions related to the anomalies produced by the AD system where his answers are inside quote-marks:

- Would you consider the behaviour of the player as high-risk for money laundering or would you consider the behaviour of the player as a responsible gambling case or do you think the above behaviour could be described as normal?

  "In this instance, his activity could be viewed as a potential RG case."

- Would you consider the above flags as anomalies or regular event for this type of players? If these flags are treated as anomalies, do you believe these anomalies can help you detect



(a) Rolling Average of the Amount of actual actions and predicted actions

(b) Rolling Average of the Wallet Balance

Figure 7.9: Information of Player 4

money laundering?

"The withdrawal amounts reflect the volatility of the products used by the customer (slots). A customer could win in slots at any given time."

- Would you want to examine the player further? If yes at which point i.e. instantly or after a few more anomalies?

  "Yes, but after a few more anomalies. Note: In this case, the customer had previously reversed some withdrawals that he reinvested until he made withdrawal (2019-01-12 04:04). His activity continued after the payment and additional withdrawals were made and cancelled. Additionally, the activity took place during the night-time, overall raising some RG concerns."

Case Study - Player 5

Player 5 was a true negative with no IRR flags but with detection system flags on the 2019-05-24, 2019-06-07 and 2019-06-12 of 2019. During the two month period we investigated the particular player, he performed over 55,000 bets, 72 deposits and 64 withdrawals. His maximum bet amount was £2. and he was managing to win bets over £100 with £1 stake.

In total our framework generated five anomalies in this for this player (1^{st},4^{th},5^{th} as likely while 2^{nd} and 3^{rd} as unlikely). It is evident that was really hard to predict when the player is going to withdrawal his money as the results on Table 7.11 show. This is evident of the huge imbalance that exist in the actions of this player with 55,000 bets and only 64 withdrawals.

1. 1^{st} Anomaly: Observed when the player withdraws an amount of £300 from his account.

Table 7.11: Player 5 anomalies detected from the anomaly detection system

Date	Amount(£)	Action Type	Action Predicted	Predicted Amount
$2019-01-12\ 02:42$	300	Withdrawal	Stake	5.23
$2019-01-12\ 04:04$	200	Withdrawal	Deposit	3.18
$2019-01-25\ 03:56$	200	Withdrawal	Deposit	1.96
$2019-01-25\ 04:25$	400	Withdrawal	Withdrawal	149.11
$2019-01-29\ 06:07$	600	Withdrawal	Stake	31.45

2. 2^{nd} Anomaly: Observed when the player withdraws an amount of £200 from his account his wallet balance after the transaction was £100.

3. 3^{rd} Anomaly: Observed when the player withdraws £200.

4. 4^{th} Anomaly: Bets £1 constantly losing money. The he managed to win 400 and then he withdraws the money immediately.

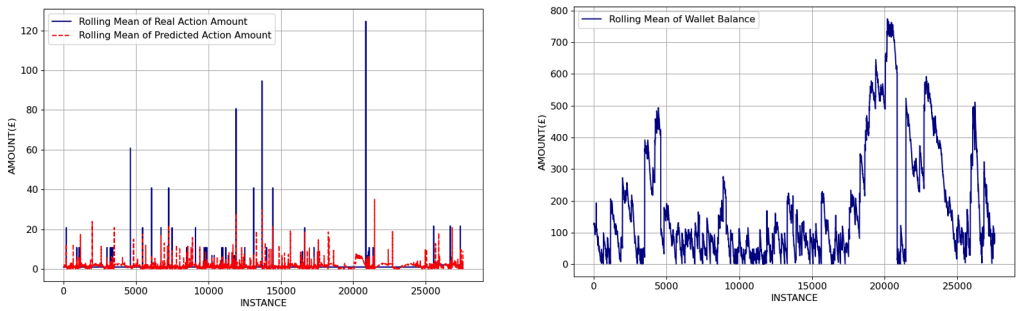
5. 5^{th} Anomaly: Similar pattern results to withdraws £600.

Using the above information, Kindred's Player Risk Officer reviewed the behaviour of customer 3 and was asked to answer the following questions related to the anomalies produced by the AD system where his answers are inside quote-marks:

- Would you consider the behaviour of the player as high-risk for money laundering or would you consider the behaviour of the player as a responsible gambling case or do you think the above behaviour could be described as normal?

  "In this instance, his activity could be viewed as a potential RG or money laundering case. Note: Playing baccarat. Highlighted activity shows that the customer changed his strategy and was trying to double his winnings, lost and deposited again."

- Would you consider the above flags as anomalies or regular event for this type of players? If these flags are treated as anomalies, do you believe these anomalies can help you detect money laundering?



(a) Rolling Average of the Amount of actual actions and predicted actions

(b) Rolling Average of the Wallet Balance

Figure 7.10: Information of Player 5

"Considering the product used (Baccarat) anomalies flagged could result helpful in certain circumstances."

- Would you want to examine the player further? If yes at which point i.e. instantly or after a few more anomalies?

  "Yes, but after a few more anomalies."

The response of our research partners towards the results produced by our AD framework was positive and encouraging. The anomalies above could indicate either money laundering risk behaviour or behaviour related to gambling addiction which as we have seen during our research sometimes could be indistinguishable. For this reason, monitoring for longer periods is necessary, as the experts suggested. To add to the above evaluation, we asked the compliance team to provide us with some general comments regarding our system and how they think it could be improved. The compliance team believed that our system would be useful and supplementary to their existing AML workflow with the condition that it is factored into the AML and business risk assessment and policy. Furthermore, the system could be expanded and take more points into consideration, going from turnover, time, age, account, payment method or the actual product. Moreover, they suggested that a more granular view of the anomalies could potentially have provided more insights regarding a specific event, i.e. AD for deposits, withdrawal or bets.

7.9 Summary

In this chapter, we had two main objectives. The first objective was to predict players' behaviour, with the use of a sequential model being very promising, with good regression and classification results. The second objective was to detect any abnormalities in players' behaviour and decide how these could be effectively used for spotting high-risk money laundering behaviours. We examined five case studies based on the results from Chapter 6. We tried to focus on cases that were wrongly predicted from our SSGAN-c. Results showed that even

though the SSGAN-c was not able to detect and flag for example Player 1, the AD system was able to spot important anomalies in this player's behaviour. Furthermore, the false positive cases from SSGAN-c still could indicate a risk of money laundering and a closer monitoring would be needed before classified those cases as low-risk.

As mentioned in the previous sections, for an unsupervised AD task like the one in this chapter, there is not a systematic and quantitative way of evaluating the performance of the model. At the end, it was decided that the anomalies labelled by the model would be assessed by a domain expert. However, due to the heavy workload needed to inspect each and every anomaly, the analysis was only done on five selected cases. The feedback we received was very positive (useful anomalies could indicate risk for money laundering), as the domain experts characterised the output of the system very promising and valuable. Our research concentrated on identifying anomalies in players' behaviour for customers playing below the rule-based system thresholds because, most of the time, they remain under the radar. This chapter showed promise for future developments of Kindred's model for detecting potentially high-risk money laundering gamblers in online casinos.

Chapter 8

Conclusion

8.1 Summary of Thesis

This thesis started by providing an overview of the current AML processes in the gambling industry and the perceptions of industry stakeholders on the money laundering problem in online gambling. We listed the challenges defined by the stakeholders and presented suggestions on what should be done to strengthen the anti-money laundering processes. In Chapter 2, an investigation was performed on the techniques and technologies used for fraud detection in gambling. The outcome of the investigation led us to the conclusion that anti-money laundering is relatively new for online gambling since most of the gambling industry's focus in the last couple of years has been on the identification of problem gamblers. However, significant research has been undertaken by financial institutions to tackle the fraud and, more specifically, anti-money laundering issues. Supervised and unsupervised methods have been applied with success. Although they could produce good results, supervised techniques face the challenge of imbalanced datasets as well as the limitation of discovering new patterns. On the contrary, unsupervised techniques could potentially assist in the identification of new patterns for money laundering and fraud at the expense of classification performance. These investigation outcomes turned our attention to how we could utilise supervised and unsupervised learning to improve the current processes on fraud detection in the online gambling environment.

In Chapter 3, we presented the data that were used and examined in this research. Kindred provided us with transactional and gaming player data for a one-year period, i.e. March 2018 to April 2019. A statistical analysis of the data was illustrated, as was the detailed process of transforming the raw data into the new global scope of features that formed our experimental dataset.

In this study, diverse methods for identifying fraud were researched, including LR, RF, XGBoost, MLP, NB and SVM. In Chapter 4, we showed the process we followed to build a fundamental supervised learning framework for the identification of high-risk players based on the machine learning techniques mentioned together with the widely used oversampling algorithms SMOTE and ADASYN. Different performances matrices were used to evaluate the performance of the supervised learning framework, such as accuracy, precision, recall, specificity and F1 score. The results suggested that XGBoost had the best overall performance when combined with SMOTE since it achieved the highest F1 measure at a score of 73.14%. Compared with the performance of the rule-based system on the test dataset, we showed that machine learning algorithms could improve the identification rate, with an increase of around 12% for the F1 score. Then, using SHAP, we provided explainability to our models on a global level. The results displayed strong evidence of overlap between problem gambling and anti-money laundering cases.

Generative adversarial networks have gained attention in the area of image and music generation. In addition, they have shown potential in tackling imbalanced class problems due to their capability of reproducing data distributions given sufficient training data samples. In Chapter 5, we introduced a GAN-based architecture network called SDG-GAN for synthetic data generation. We evaluated our approach against the oversampling techniques of ADASYN, SMOTE, B-SMOTE and cGAN. The ability to generate new synthetic data and assist in the imbalanced class problem was evaluated by calculating the algorithmic performance when SDG-GAN was combined with the supervised machine learning algorithms. We used three benchmark datasets, two from the healthcare space (Breast Cancer Wisconsin and Pima Diabetes) and one from the financial credit card fraud space. At the end, oversampling techniques were evaluated on a real-world gambling dataset. In terms of algorithmic performance, our method in combination with

RF produced the best overall results, achieving the highest F1 score. Then, to assess whether the oversampling techniques were generating new synthetic instances rather than replicating the old ones, the statistical Wilcoxon rank-sum and Kolmogorov–Smirnov tests were carried out.

Building on the work from Chapter 5, in Chapter 6, we proposed a new architecture for imbalanced data classification that did not require any oversampling techniques to produce good classification results. Our novel system was based on semi-supervised adversarial networks and sparse auto-encoders. Various experiments were undertaken to evaluate the proposed architecture against the popular discriminative techniques of LR, RF, XGB and MLP in conjunction with data oversampling techniques SMOTE and ADASYN. The results observed from the three benchmark datasets from Chapter 5 and the real-world gambling dataset showed that complementary SSGAN could be a useful versatile framework for tackling supervised problems with imbalanced data.

Although the results achieved using supervised learning were promising, discovering new patterns and trends in relation to money laundering risk was not possible. In Chapter 7, we developed and implemented an adaptive unsupervised learning system, which was used to detect anomalies by predicting the next action of a player in terms of amount and type. Then, the classification and regression errors from the LSTM-ATT network were used to fit a multivariate Gaussian, and the probability of an event being an anomaly was calculated. We performed the anomaly detection on five players and evaluated the results with the assistance of the Kindred compliance team.

8.2 Contributions and Findings

This section presents the thesis's contributions to knowledge through the following questions.

- What are the main challenges of detecting money laundering in online gambling?

The limited feedback on the money laundering cases from the financial crime agencies to the

gambling operators have made the life of the operators difficult in order to gather high quality information on their high risk for money laundering cases. However, the limited feedback is due to the heavy load of cases that the crime agencies are facing. Further, have difficulties on gathering source of funds and affordability information, in order to evaluate each player situation. Even where a full wealth profile is established for a player, their financial circumstances can change significantly due to loss of employment or significant purchases, such as the purchase of property.

In a relatively new area wherein the research is limited, we analysed and summarised different takes from industry experts based on stakeholder interviews that included experts from national crime agencies, regulators, trade associations, suppliers and operators [7]. The following technical recommendations for the industry could be used as guidance that could assist operators and regulators in finding a better approach to the money laundering fraud problem.

Recommendations for Regulators and Crime Agencies

- Develop a single format or technical protocol for submitting STRs and SARs across jurisdictions that enables operators to submit cases using a consistent system whilst also providing feedback on submission quality. This will i) enable the industry to save money on the increasing manual efforts and costs to submit returns and re-invest in improving systems, and therefore, the quality of submissions, and ii) subsequently reduce the load on the agencies by reducing the number of poorer quality cases.

- Continue exploring opportunities to develop a single central database for customers flagged for suspicious gambling activity, to enable enhanced monitoring of flagged customers across the industry. Combining data could have a significant impact on improving compliance processes in the industry, thus raising standards.

Recommendations for Online Gambling Operators

- Develop more sophisticated and cost-efficient methods to improve ongoing monitoring. This would entail building techniques that analyse players' behaviour below the minimum threshold levels required by regulators and avoid relying on increased staff numbers to broaden the monitoring scope. Supervised and unsupervised learning can be used effectively as showed in this thesis.

- Use data to develop more sophisticated behavioural checks and customer affordability segments to support enhanced sources of funds checks throughout the customer lifecycle for higher spenders rather than just at specific points such as regulatory threshold breaches (e.g. a customer depositing over £1.5 K within a 24-hour period)

The findings from the stakeholder interviews suggested that whilst the current systems and processes on registration and at regulatory thresholds are reasonably robust, more focus needs to be given to the ongoing monitoring of customers' behaviours. This indicates that typically only a small percentage of customer behaviours are subject to a detailed and ongoing customer analysis from an AML and proceeds of crime perspective.

- How can a model developed with the supervised learning approach be used to effectively analyse gambling fraud?

The process of improving the existing online gambling framework starts by designing and enhancing new features that allow a user's behavioural pattern to be captured effectively. Following the creation of the scope of new behavioural variables, a supervised learning framework is provided and a detailed comparative analysis of popular supervised learning methods is shown. One thing all the different fields of fraud detection have in common is the level of class imbalance. Generally, only a small percentage of the total number of transactions is actual fraud. In our dataset, only a small percentage of the total population of players had been flagged as a high money laundering risk. Therefore, training a supervised learning algorithm to predict fraud is a process that will result in many false negatives due to the bias towards the majority

class. Oversampling methods have been used extensively for improving the classification task of imbalanced datasets. For instance, SMOTE and ADASYN are two of the most widely used techniques and were applied in our experiments in Chapter 4 to balance the data and improve the classification performance of the machine learning algorithms towards the high-risk class. Furthermore, the results from Chapter 4 suggested an improvement in the overall identification rate compared to the existing rule-based system. Using SHAP, we identified which features were the most important for the supervised algorithms to make a decision, and an overlap with responsible gambling was found.

We extended the exploration towards improving the class imbalance issue in fraud detection, and we proposed a new architecture based on GANs for synthetic data generation. Our method uses the generator of a Generative Adversarial Network to generate high quality synthetic data. The results showed that SDG-GAN plus a context classifier provided a good detection rate across the different classes in the sample data. Building on the architecture of SDG-GAN, in Chapter 6 we introduced a new method for robust classification of binary imbalanced datasets. The novel two part architecture we propose is based on sparse auto-encoders and semi-supervised GANs. The experiments suggested that SAE+SSGAN-c could be a reliable classification method for imbalanced data without the need to apply oversampling in the process, as it achieved the highest performance in comparison with all methods applied.

- How can a model flag new behaviours which could potentially be related to money laundering and learn new patterns?

The supervised learning architecture constructed in this thesis was shown to significantly improve the detection rate of players with a high-risk for money laundering when the results were compared with the original rule-based system. To identify new suspicious behaviours, a two-step anomaly detection framework was proposed. Upon testing our framework on time series of five different players, we received positive feedback from our research partners. The system flagged players a few months in advance than the automated rule-based system, showing promising signs as the feedback from Kindred indicated. Moreover, Kindred stressed that this

system has the potential to be an important tool for flagging players who try to stay far below the predefined thresholds to remain unnoticeable. In an ideal situation, our research partners suggested that since the preliminary results from the case studies were positive, there was a willingness to test the system live.

8.3 Methods Comparison for Imbalanced Classification

In this thesis we tackled the fraud detection problem in online gambling, however our research can be applied in other areas as well, especially on the imbalanced classification problem. Despite more than two decades of continuous development, utilising imbalanced data is still a focus of intense research [206]. Different techniques on the data and algorithmic level were investigated in this thesis and very promising results were achieved.

This section summarises the findings and provides an overview regarding the various techniques we used for classifying money laundering risk in online gambling. Further, we explain how our approach can be extended from fraud detection in online gambling to provide solution to the general imbalanced classification issue.

In Table 8.1 we present all the techniques that were investigated based on their F1 score and average complexity time across all the examined datasets (gambling fraud and the public benchmark datasets of Pima Diabetes, Wisconsin Breast Cancer and Credit Card Fraud). Firstly, it can be noticed that in terms of classification performance our proposed method of SSGAN-c+SAE outperformed other techniques in 3 out of the 4 imbalanced datasets. That verifies the ability of semi-supervised GANs to produce good classification results with few training data. One thing that was consistently noticed, when handling imbalanced data with SMOTE, ADASYN and B-SMOTE, is that, those techniques could suffer from over-fitting and over-lapping classes, which was showed by the huge improvement that was observed in some cases in the recall score and a significant drop in the performance of precision. However, when generative networks approaches were applied, a more robust improvement was achieved. In addition, the results from Table 8.1 suggest that our approaches can be extended and applied

Table 8.1: Summary Results of methods used for tackling imbalanced class problem in terms of F1 score and average time complexity across all the datasets

Methods	F1-Credit Card	F1-Breast Cancer	F1-Pima Diabetes	F1-Gambling Fraud	Average Complexity time (s)
LR+SMOTE	0.9032	0.9091	0.6456	0.8544	**0.08**
LR+ADASYN	0.8246	0.8837	0.6250	0.8421	0.10
LR+BSMOTE	0.8888	0.909	0.6290	0.8427	0.10
LR+CGAN	0.8926	0.8966	0.6788	0.7500	39.12
LR+SDGAN	0.8889	0.9157	0.6549	0.7964	37.93
RF+SMOTE	0.9116	0.8764	0.6774	0.8838	0.69
RF+ADASYN	0.8771	0.8863	0.6507	0.8764	0.75
RF+BSMOTE	0.8999	0.8863	0.6491	0.8761	0.80
RF+CGAN	0.9010	0.8809	0.6315	0.8887	38.21
RF+SDGAN	0.9131	0.8706	**0.7080**	0.8973	42.38
XGB+SMOTE	0.9060	0.8742	0.6555	0.8938	0.47
XGB+ADASYN	0.8935	0.8754	0.6562	0.8871	0.54
XGB+BSMOTE	0.8687	0.8821	0.6508	0.8880	0.57
XGB+CGAN	0.9091	0.9195	0.6379	0.8821	38.03
XGB+SDGAN	0.9087	0.9130	0.6207	0.8961	45.06
MLP+SMOTE	0.8601	0.8965	0.6888	0.8545	2.31
MLP+ADASYN	0.8803	0.8578	0.6718	0.8582	2.36
MLP+BSMOTE	0.9090	0.9090	0.6555	0.8489	2.41
MLP+CGAN	0.9111	0.8965	0.6607	0.8807	41.99
MLP+SDGAN	0.8862	0.9137	0.6788	0.8619	45.57
SSGAN+SAE	**0.9231**	**0.9227**	0.6904	**0.8985**	47.14

into different sectors and not limited to the fraud detection problem in online gambling.

Even though our proposed method of SSGAN-c+SAE achieved the best classification results, it comes with the drawback of the highest training time in comparison with the other approaches, with time 47.14(s) on average. SMOTE and the other density based techniques managed to keep their complexity time under 3(s) on average. Nevertheless, we believe that GANs impressive results and advancement in deep learning techniques, will result to a wider use and acceptance of GANs in the area of fraud detection and more general in the imbalanced classification space [158].

8.4 Future Work

This section discusses future research directions despite the successful results achieved in this thesis. A list of possible future improvements and works for both the gambling community and

researched areas of fraud and anomaly detection are summarised below.

To start, the findings of this thesis related to the money laundering problem in online gambling could be used to guide operators in strengthening their existing processes. The technical recommendations for both operators and regulators from [7] should be considered by the industry to better improve the existing monitoring. In addition, we showed how machine learning could be utilised at both the unsupervised and supervised levels to tackle fraud-related problems in the gambling industry. Further experimentation and live testing are needed to verify the ability of machine learning algorithms to capture high-risk behaviours.

As shown in this thesis, GANs represent one of the most intriguing recent AI techniques [207, 208]. Their concept has enabled the AI industry to take huge leaps in creativity, generating images and sounds that are very close to their natural counterparts. Moreover, GANs could be crucial in generating new data and tackling imbalanced class problems. Previously, AI has been used to analyse, internalise and predict, but with the rise of GANs, AI can now create.

However, the use and application of these methods are limited and the main reason is the inability to provide explanations to the outcome of those systems. In the anomaly and fraud detection field, the absence of a good explanation restricts the user from taking the appropriate actions. Therefore, they have a major downside, as their output is hard to explain. This limitation could make it harder to convince experts – in our case, compliance officers – to trust and use potentially beneficial anomaly detection systems. The output of these systems may include anomalous instances that the domain expert was previously unaware of and, by providing an explanation of the results, could increase the faith of the users in the AI system. Therefore, further research needs to explain the fraud and anomaly detection results better.

Explainable AI, especially explainable machine learning, is essential to understand and trust an artificially intelligent technique. The field of XAI has emerged in research and aims to develop methods wherein the process leading to a model output can be understood by humans. In this thesis, we considered the SHAP method as a model agnostic approach to provide interpretability on a global level. Nevertheless, to make any final decision on a specific case, local explainability should be considered in the process.

Local Interpretable Model-agnostic Explanations (LIME) [209] has been successful in the field as an XAI method applied to images (but not to GANs). This [209] is a technique that explains how the input features of a machine learning model affect its predictions. For instance, for image classification tasks, LIME finds the region of an image with the strongest association with a prediction label. The importance score is produced by capturing the feature interaction between the features and output using a linear model between the features. However, the limitations of LIME include an inability to measure the fidelity of its regression coefficients, which hides the fact that it may often be producing false explanations. Another problem is that LIME does not provide counterfactual explanations. These limitations, however, are solved by CLEAR [210].

The CLEAR technique provides counterfactual explanations based on the advantages of two explanatory methods while simultaneously addressing their drawbacks. The first method was introduced by Wachter et al. [152], who argued that single predictions are explained by 'boundary counterfactuals', which state the minimum changes needed for an observation to 'flip' its classification. The second method was introduced by Riberio et al. [209], who argued for local interpretable model-agnostic explanations, which are created by building a regression model that seeks to approximate the local input–output behaviour of the machine learning system. CLEAR can be applied to both GANs and images and will need to be adapted to, applied to and evaluated on gambling data. An explanation of why an instance is anomalous would enable experts to focus their investigation on the most important anomalies and could increase their trust in the algorithm.

Further another direction for future work is to utilise GANs for anomaly detection on time series. Building on the research from [211], an anomaly detection framework for multivariate time series based on GANs could be developed. Multivariate anomaly detection (MAD)-GANs [211], uses LSTM recurrent neural networks in both generator's and discriminator's architectures. Another example wherein anomaly detection was used with GANs was in a paper published by Schlegh et al. [212]. The authors proposed AnoGAN, an unsupervised learning system to identify anomalies in imaging data. Empirically, the results of the two works mentioned above show that GANs could be a powerful tool for spotting anomalies in spatial temporal data. Therefore, exploiting the idea of adapting our anomaly detection framework from Chapter 7 in a generative

adversarial architecture could help tackle the imbalanced problem of uneven events and improve overall the AD system.

Bibliography

[1] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.

[2] M. S. Kumar, V. Soundarya, S. Kavitha, E. Keerthika, and E. Aswini, "Credit card fraud detection using random forest algorithm," in *2019 3rd International Conference on Computing and Communications Technologies (ICCCT)*. IEEE, 2019, pp. 149–153.

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[4] T. Zebin, M. Sperrin, N. Peek, and A. J. Casson, "Human activity recognition from inertial sensor time-series using batch normalized deep lstm recurrent networks," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2018, pp. 1–4.

[5] S. Zolkaflil, N. Omar, and S. N. F. S. M. Nazri, "Factors influencing the outcome of money laundering investigations," in *Money Laundering and Terrorism Financing in Global Financial Systems*. IGI Global, 2021, pp. 128–156.

[6] G. Mangion, "Perspective from malta: Money laundering and its relation to online gambling," *Gaming Law Review and Economics*, vol. 14, no. 5, pp. 363–370, 2010.

[7] C. Charitou, S. Dragicevic, and A. Garcez, "Raising standards in compliance: Application of artificial intelligence to online gambling

data to identify anomalous behaviours," Jul 2018. [Online]. Available: https://www.city.ac.uk/__data/assets/pdf_file/0014/421106/City-collaborative-Whitepaper-Anti-Money-Laundering-and-Artificial-Intelligence-02July2018.pdf

[8] F. Schneider, "Money laundering and financial means of organised crime: some preliminary empirical findings," *Global Business and Economics Review*, vol. 10, no. 3, pp. 309–330, 2008.

[9] C. Charitou, A. d. Garcez, and S. Dragicevic, "Semi-supervised gans for fraud detection," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.

[10] C. Charitou, S. Dragicevic, and A. d'Avila Garcez, "Synthetic data generation for fraud detection using gans," 2021. [Online]. Available: http://arxiv.org/abs/2109.12546

[11] J. Braverman and H. Shaffer, "How do gamblers start gambling: Identifying behavioral markers for high-risk internet gambling," *European journal of public health*, vol. 22, pp. 273–8, 01 2010.

[12] S. Dragicevic, C. Percy, A. Kudic, and J. Parke, "A descriptive analysis of demographic and behavioral data from internet gamblers and those who self-exclude from online gambling platforms," *Journal of Gambling Studies*, vol. 31, no. 1, pp. 105–132, 2015.

[13] S. Akhter *et al.*, "Using machine learning to predict potential online gambling addicts." 2018.

[14] C. Percy, M. França, S. Dragicevic, and A. Garcez, "Predicting online gambling self-exclusion: an analysis of the performance of supervised machine learning models," *International Gambling Studies*, vol. 16, pp. 1–18, 04 2016.

[15] C. Percy, A. S. d. Garcez, S. Dragičević, M. V. França, G. Slabaugh, and T. Weyde, "The need for knowledge extraction: understanding harmful gambling behavior with neural networks," in *Proceedings of the Twenty-second European Conference on Artificial Intelligence*, 2016, pp. 974–981.

[16] J. H. Wilson, "An analytical approach to detecting insurance fraud using logistic regression," *Journal of Finance and Accountancy*, vol. 1, p. 1, 2009.

[17] D. Yue, X. Wu, Y. Wang, Y. Li, and C.-H. Chu, "A review of data mining-based financial fraud detection research," in *2007 International Conference on Wireless Communications, Networking and Mobile Computing*. Ieee, 2007, pp. 5519–5522.

[18] R. Bhowmik, "Detecting auto insurance fraud by data mining techniques," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 2, no. 4, pp. 156–162, 2011.

[19] S. Khatri, A. Arora, and A. P. Agrawal, "Supervised machine learning algorithms for credit card fraud detection: A comparison," in *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, 2020, pp. 680–683.

[20] C. S. Hilas and P. A. Mastorocostas, "An application of supervised and unsupervised learning approaches to telecommunications fraud detection," *Knowledge-Based Systems*, vol. 21, no. 7, pp. 721–726, 2008.

[21] X. Niu, L. Wang, and X. Yang, "A comparison study of credit card fraud detection: Supervised versus unsupervised," *arXiv preprint arXiv:1904.10604*, 2019.

[22] A. Dal Pozzolo, "Adaptive machine learning for credit card fraud detection," 2015.

[23] A. Sudjianto, S. Nair, M. Yuan, A. Zhang, D. Kern, and F. Cela-Díaz, "Statistical methods for fighting financial crimes," *Technometrics*, vol. 52, no. 1, pp. 5–19, 2010.

[24] P. K. Chan, W. Fan, A. L. Prodromidis, and S. J. Stolfo, "Distributed data mining in credit card fraud detection," *IEEE Intelligent Systems and Their Applications*, vol. 14, no. 6, pp. 67–74, 1999.

[25] H. Wang, W. Fan, P. S. Yu, and J. Han, "Mining concept-drifting data streams using ensemble classifiers," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003, pp. 226–235.

[26] J. Hollmén *et al.*, *User profiling and classification for fraud detection in mobile communications networks.* Helsinki University of Technology, 2000.

[27] L.-T. Huang, M. M. Gromiha, and S.-Y. Ho, "iptree-stab: interpretable decision tree based method for predicting protein stability changes upon mutations," *Bioinformatics*, vol. 23, no. 10, pp. 1292–1293, 2007.

[28] C. Cody, V. Ford, and A. Siraj, "Decision tree learning for fraud detection in consumer energy consumption," in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA).* IEEE, 2015, pp. 1175–1179.

[29] J. Yao, J. Zhang, and L. Wang, "A financial statement fraud detection model based on hybrid data mining methods," in *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD).* IEEE, 2018, pp. 57–61.

[30] V. Jain, M. Agrawal, and A. Kumar, "Performance analysis of machine learning algorithms in credit cards fraud detection," in *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO).* IEEE, 2020, pp. 86–88.

[31] D. S. Rosario, "Highly effective logistic regression model for signal (anomaly) detection," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5. IEEE, 2004, pp. V–817.

[32] M. S. Mok, S. Y. Sohn, and Y. H. Ju, "Random effects logistic regression model for anomaly detection," *Expert Syst. Appl.*, vol. 37, no. 10, pp. 7162–7166, Oct. 2010. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2010.04.017

[33] A. U. S. Khan, N. Akhtar, and M. N. Qureshi, "Real-time credit-card fraud detection using artificial neural network tuned by simulated annealing algorithm," in *Proceedings of International Conference on Recent Trends in Information, Telecommunication and Computing, ITC.* Citeseer, 2014, pp. 113–121.

[34] X. Yu, X. Li, Y. Dong, and R. Zheng, "A deep neural network algorithm for detecting credit card fraud," in *2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*. IEEE, 2020, pp. 181–183.

[35] K. Fu, D. Cheng, Y. Tu, and L. Zhang, "Credit card fraud detection using convolutional neural networks," in *International Conference on Neural Information Processing*. Springer, 2016, pp. 483–490.

[36] A. M. Mubarek and E. Adalı, "Multilayer perceptron neural network technique for fraud detection," in *2017 International Conference on Computer Science and Engineering (UBMK)*. IEEE, 2017, pp. 383–387.

[37] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 2013.

[38] N. K. Gyamfi and J.-D. Abdulai, "Bank fraud detection using support vector machine," in *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. IEEE, 2018, pp. 37–41.

[39] I. Rajak and K. J. Mathai, "Intelligent fraudulent detection system based svm and optimized by danger theory," in *2015 International Conference on Computer, Communication and Control (IC4)*. IEEE, 2015, pp. 1–4.

[40] V. Mareeswari and G. Gunasekaran, "Prevention of credit card fraud detection based on hsvm," in *2016 International Conference on Information Communication and Embedded Systems (ICICES)*. IEEE, 2016, pp. 1–4.

[41] G. G. Sundarkumar, V. Ravi, and V. Siddeshwar, "One-class support vector machine based undersampling: Application to churn prediction and insurance fraud detection," in *2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*. IEEE, 2015, pp. 1–7.

[42] Y. Ma, S. Liang, X. Chen, and C. Jia, "The approach to detect abnormal access behavior based on naive bayes algorithm," in *2016 10th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*. IEEE, 2016, pp. 313–315.

[43] D. D. Arifin, M. A. Bijaksana *et al.*, "Enhancing spam detection on mobile phone short message service (sms) performance using fp-growth and naive bayes classifier," in *2016 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*. IEEE, 2016, pp. 80–84.

[44] D. Rahmawati, R. Sarno, C. Fatichah, and D. Sunaryono, "Fraud detection on event log of bank financial credit business process using hidden markov model algorithm," in *2017 3rd International Conference on Science in Information Technology (ICSITech)*. IEEE, 2017, pp. 35–40.

[45] X. Wang, H. Wu, and Z. Yi, "Research on bank anti-fraud model based on k-means and hidden markov model," in *2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC)*. IEEE, 2018, pp. 780–784.

[46] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," *Decision Support Systems*, vol. 50, no. 3, pp. 602–613, 2011.

[47] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang, and C. Jiang, "Random forest for credit card fraud detection," in *2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC)*. IEEE, 2018, pp. 1–6.

[48] C. Liu, Y. Chan, S. H. Alam Kazmi, and H. Fu, "Financial fraud detection model: Based on random forest," *International journal of economics and finance*, vol. 7, no. 7, 2015.

[49] R. Bauder and T. Khoshgoftaar, "Medicare fraud detection using random forest with class imbalanced big data," in *2018 IEEE international conference on information reuse and integration (IRI)*. IEEE, 2018, pp. 80–87.

[50] J. Hancock and T. M. Khoshgoftaar, "Performance of catboost and xgboost in medicare fraud detection," in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2020, pp. 572–579.

[51] J. Riffi, M. A. Mahraz, A. El Yahyaouy, H. Tairi *et al.*, "Credit card fraud detection based on multilayer perceptron and extreme learning machine architectures," in *2020*

International Conference on Intelligent Systems and Computer Vision (ISCV). IEEE, 2020, pp. 1–5.

[52] G. E. Batista, A. C. Carvalho, and M. C. Monard, "Applying one-sided selection to unbalanced datasets," in *Mexican International Conference on Artificial Intelligence*. Springer, 2000, pp. 315–325.

[53] N. V. Chawla, "Data mining for imbalanced datasets: An overview," in *Data mining and knowledge discovery handbook*. Springer, 2009, pp. 875–886.

[54] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Special issue on learning from imbalanced data sets," *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 1–6, 2004.

[55] R. Batuwita and V. Palade, "Class imbalance learning methods for support vector machines," 2013.

[56] H. Alhakbani, "Handling class imbalance using swarm intelligence techniques, hybrid data and algorithmic level solutions," Ph.D. dissertation, Goldsmiths, University of London, 2019.

[57] N. Japkowicz and M. Shah, *Evaluating learning algorithms: a classification perspective*. Cambridge University Press, 2011.

[58] J. Stefanowski, "Overlapping, rare examples and class decomposition in learning classifiers from imbalanced data," in *Emerging paradigms in machine learning*. Springer, 2013, pp. 277–306.

[59] K. Napierala and J. Stefanowski, "Types of minority class examples and their influence on learning classifiers from imbalanced data," *Journal of Intelligent Information Systems*, vol. 46, no. 3, pp. 563–597, 2016.

[60] M. Kubat, S. Matwin *et al.*, "Addressing the curse of imbalanced training sets: one-sided selection," in *Icml*, vol. 97. Citeseer, 1997, pp. 179–186.

[61] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[62] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*. IEEE, 2008, pp. 1322–1328.

[63] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: a new over-sampling method in imbalanced data sets learning," in *International conference on intelligent computing*. Springer, 2005, pp. 878–887.

[64] S. García, J. Luengo, and F. Herrera, *Data preprocessing in data mining*. Springer, 2015.

[65] K. McCarthy, B. Zabar, and G. Weiss, "Does cost-sensitive learning beat sampling for classifying rare classes?" in *Proceedings of the 1st international workshop on Utility-based data mining*, 2005, pp. 69–77.

[66] N. V. Chawla, "Data mining for imbalanced datasets: An overview," in *Data mining and knowledge discovery handbook*. Springer, 2009, pp. 875–886.

[67] A. Dal Pozzolo, O. Caelen, and G. Bontempi, "When is undersampling effective in unbalanced classification tasks?" in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2015, pp. 200–215.

[68] G. Batista, R. Prati, and M.-C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *SIGKDD Explorations*, vol. 6, pp. 20–29, 06 2004.

[69] C. X. Ling and C. Li, "Data mining for direct marketing: Problems and solutions." in *Kdd*, vol. 98, 1998, pp. 73–79.

[70] S. Ganguly and S. Sadaoui, "Classification of imbalanced auction fraud data," in *Canadian Conference on Artificial Intelligence*. Springer, 2017, pp. 84–89.

[71] G. E. Batista, A. L. Bazzan, and M. C. Monard, "Balancing training data for automated annotation of keywords: a case study." in *WOB*, 2003, pp. 10–18.

[72] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[73] J. L. Leevy, T. M. Khoshgoftaar, R. A. Bauder, and N. Seliya, "A survey on addressing high-class imbalance in big data," *Journal of Big Data*, vol. 5, no. 1, p. 42, 2018.

[74] G. M. Weiss, K. McCarthy, and B. Zabar, "Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs?" *Dmin*, vol. 7, no. 35-41, p. 24, 2007.

[75] R. V. Rao and V. Patel, "Multi-objective optimization of heat exchangers using a modified teaching-learning-based optimization algorithm," *Applied Mathematical Modelling*, vol. 37, no. 3, pp. 1147–1162, 2013.

[76] P. Cao, D. Zhao, and O. R. Zaïane, "A pso-based cost-sensitive neural network for imbalanced data classification," in *Pacific-Asia conference on knowledge discovery and data mining*. Springer, 2013, pp. 452–463.

[77] P. Cao, D. Zhao, and O. Zaiane, "An optimized cost-sensitive svm for imbalanced data learning," in *Pacific-Asia conference on knowledge discovery and data mining*. Springer, 2013, pp. 280–292.

[78] A. Ali, S. M. Shamsuddin, A. L. Ralescu *et al.*, "Classification with class imbalance problem: a review," *Int. J. Advance Soft Compu. Appl*, vol. 7, no. 3, pp. 176–204, 2015.

[79] Y. Wu, L. Shen, and S. Zhang, "Fuzzy multiclass support vector machines for unbalanced data," in *2017 29th Chinese Control And Decision Conference (CCDC)*. IEEE, 2017, pp. 2227–2231.

[80] P. Shukla and K. Bhowmick, "To improve classification of imbalanced datasets," in *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*. IEEE, 2017, pp. 1–5.

[81] H. Luo, X. Pan, Q. Wang, S. Ye, and Y. Qian, "Logistic regression and random forest for effective imbalanced classification," in *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, vol. 1. IEEE, 2019, pp. 916–917.

[82] K. Li, P. Xie, J. Zhai, and W. Liu, "An improved adaboost algorithm for imbalanced data based on weighted knn," in *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)(*. IEEE, 2017, pp. 30–34.

[83] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 463–484, 2011.

[84] C. Chen, A. Liaw, L. Breiman *et al.*, "Using random forest to learn imbalanced data," *University of California, Berkeley*, vol. 110, no. 1-12, p. 24, 2004.

[85] Y. O. Lee, J. Jo, and J. Hwang, "Application of deep neural network and generative adversarial network to industrial maintenance: A case study of induction motor fault detection," in *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 2017, pp. 3248–3253.

[86] Y. Xian, S. Sharma, B. Schiele, and Z. Akata, "f-vaegan-d2: A feature generating framework for any-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 275–10 284.

[87] E. L. Denton, S. Chintala, R. Fergus *et al.*, "Deep generative image models a laplacian pyramid of adversarial networks," in *Advances in neural information processing systems*, 2015, pp. 1486–1494.

[88] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.

[89] M. Zhai, L. Chen, F. Tung, J. He, M. Nawhal, and G. Mori, "Lifelong gan: Continual learning for conditional image generation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2759–2768.

[90] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He, "Attngan: Fine-grained text to image generation with attentional generative adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1316–1324.

[91] G. Douzas and F. Bacao, "Effective data generation for imbalanced learning using conditional generative adversarial networks," *Expert Systems with applications*, vol. 91, pp. 464–471, 2018.

[92] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional gan," in *Advances in Neural Information Processing Systems*, 2019, pp. 7335–7345.

[93] U. Fiore, A. De Santis, F. Perla, P. Zanetti, and F. Palmieri, "Using generative adversarial networks for improving classification effectiveness in credit card fraud detection," *Information Sciences*, vol. 479, pp. 448–455, 2019.

[94] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009. [Online]. Available: http://doi.acm.org/10.1145/1541880.1541882

[95] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *arXiv preprint arXiv:1901.03407*, 2019.

[96] E. H. Budiarto, A. E. Permanasari, and S. Fauziati, "Unsupervised anomaly detection using k-means, local outlier factor and one class svm," in *2019 5th International Conference on Science and Technology (ICST)*, vol. 1. IEEE, 2019, pp. 1–5.

[97] S. Mahadevan and S. L. Shah, "Fault detection and diagnosis in process data using one-class support vector machines," *Journal of process control*, vol. 19, no. 10, pp. 1627–1639, 2009.

[98] Y. Huang and Q. Zhang, "Identification of anomaly behavior of ships based on knn and lof combination algorithm," in *AIP Conference Proceedings*, vol. 2073, no. 1. AIP Publishing LLC, 2019, p. 020090.

[99] G. Pu, L. Wang, J. Shen, and F. Dong, "A hybrid unsupervised clustering-based anomaly detection method," *Tsinghua Science and Technology*, vol. 26, no. 2, pp. 146–153, 2020.

[100] P. Monamo, V. Marivate, and B. Twala, "Unsupervised learning for robust bitcoin fraud detection," in *2016 Information Security for South Africa (ISSA)*. IEEE, 2016, pp. 129–134.

[101] C.-q. Ma, Y. Li, and X. Hao, "Research of insurance fraud risk based on bayesian networks," in *2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC)*. IEEE, 2011, pp. 3220–3225.

[102] M. I. M. Yusoff, I. Mohamed, and M. R. A. Bakar, "Fraud detection in telecommunication industry using gaussian mixed model," in *2013 International Conference on Research and Innovation in Information Systems (ICRIIS)*. IEEE, 2013, pp. 27–32.

[103] M. Schreyer, T. Sattarov, D. Borth, A. Dengel, and B. Reimer, "Detection of anomalies in large scale accounting data using deep autoencoder networks," *arXiv preprint arXiv:1709.05254*, 2017.

[104] R. Wedge, J. M. Kanter, S. M. Rubio, S. I. Perez, and K. Veeramachaneni, "Solving the" false positives" problem in fraud prediction," *arXiv preprint arXiv:1710.07709*, 2017.

[105] R. Chalapathy, A. K. Menon, and S. Chawla, "Anomaly detection using one-class neural networks," *CoRR*, vol. abs/1802.06360, 2018. [Online]. Available: http://arxiv.org/abs/1802.06360

[106] A. Chouiekh and E. H. I. E. Haj, "Convnets for fraud detection analysis," *Procedia Computer Science*, vol. 127, pp. 133–138, 2018.

[107] Z. Zhang, X. Zhou, X. Zhang, L. Wang, and P. Wang, "A model based on convolutional neural network for online transaction fraud detection," *Security and Communication Networks*, vol. 2018, 2018.

[108] J. Liu, H. Zhu, Y. Liu, H. Wu, Y. Lan, and X. Zhang, "Anomaly detection for time series using temporal convolutional networks and gaussian mixture model," in *Journal of Physics: Conference Series*, vol. 1187, no. 4. IOP Publishing, 2019, p. 042111.

[109] J. Contreras, R. Espinola, F. J. Nogales, and A. J. Conejo, "Arima models to predict next-day electricity prices," *IEEE transactions on power systems*, vol. 18, no. 3, pp. 1014–1020, 2003.

[110] I. Kelilume and A. Salami, "Modeling and forecasting inflation with arima and var: The case of nigeria," *ISSN 2168-0612 FLASH DRIVE ISSN 1941-9589 ONLINE*, p. 62, 2013.

[111] C. Yang, F. Deng, and H. Yang, "An unsupervised anomaly detection approach using subtractive clustering and hidden markov model," *2007 Second International Conference on Communications and Networking in China*, pp. 313–316, 2007.

[112] W. Khreich, E. Granger, R. Sabourin, and A. Miri, "Combining hidden markov models for improved anomaly detection," in *2009 IEEE International Conference on Communications*. IEEE, 2009, pp. 1–6.

[113] N. Görnitz, M. Braun, and M. Kloft, "Hidden markov anomaly detection," in *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ser. ICML'15. JMLR.org, 2015, pp. 1833–1842. [Online]. Available: http://dl.acm.org/citation.cfm?id=3045118.3045313

[114] A. Srivastava, A. Kundu, S. Sural, and A. Majumdar, "Credit card fraud detection using hidden markov model," *IEEE Transactions on dependable and secure computing*, vol. 5, no. 1, pp. 37–48, 2008.

[115] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[116] Y. Heryadi and H. L. H. S. Warnars, "Learning temporal representation of transaction amount for fraudulent transaction recognition using cnn, stacked lstm, and cnn-lstm," in *2017 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*. IEEE, 2017, pp. 84–89.

[117] Y. Ando, H. Gomi, and H. Tanaka, "Detecting fraudulent behavior using recurrent neural networks," in *Computer Security Symposium*, 2016.

[118] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *Proceedings*, vol. 89. Presses universitaires de Louvain, 2015.

[119] S. Wang, C. Liu, X. Gao, H. Qu, and W. Xu, "Session-based fraud detection in online e-commerce transactions using recurrent neural networks," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2017, pp. 241–252.

[120] X. Lp, W. Yu, T. Luwang, J. Zheng, X. Qiu, J. Zhao, L. Xia, and Y. Li, "Transaction fraud detection using gru-centered sandwich-structured model," in *2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design ((CSCWD))*. IEEE, 2018, pp. 467–472.

[121] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "Lstm-based encoder-decoder for multi-sensor anomaly detection," *CoRR*, vol. abs/1607.00148, 2016. [Online]. Available: http://arxiv.org/abs/1607.00148

[122] M. Yadav, P. Malhotra, L. Vig, K. Sriram, and G. Shroff, "ODE - augmented training improves anomaly detection in sensor data from machines," *CoRR*, vol. abs/1605.01534, 2016. [Online]. Available: http://arxiv.org/abs/1605.01534

[123] C. Ferri, J. Hernández-Orallo, and R. Modroiu, "An experimental comparison of performance measures for classification," *Pattern Recognition Letters*, vol. 30, no. 1, pp. 27–38, 2009.

[124] J. West and M. Bhattacharya, "Mining financial statement fraud: An analysis of some experimental issues," in *2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA)*. IEEE, 2015, pp. 461–466.

[125] C. Whitrow, D. J. Hand, P. Juszczak, D. Weston, and N. M. Adams, "Transaction aggregation as a strategy for credit card fraud detection," *Data mining and knowledge discovery*, vol. 18, no. 1, pp. 30–55, 2009.

[126] F. A. T. Force, "Money laundering through the football sector," *Paris: FATF/OECD*, 2009.

[127] "Risk assessment for licensed betting offices and remote gambling industry," 2017. [Online]. Available: https://www.rga.eu.com/wp-content/uploads/GAMLG-AML-Risk-Assessment.pdf

[128] N. I. Fisher, *Statistical analysis of circular data*. cambridge university press, 1995.

[129] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[130] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[131] Y. Sahin and E. Duman, "Detecting credit card fraud by ann and logistic regression," in *2011 International Symposium on Innovations in Intelligent Systems and Applications*. IEEE, 2011, pp. 315–319.

[132] E. K. Sahin, "Assessing the predictive capability of ensemble tree methods for landslide susceptibility mapping using xgboost, gradient boosting machine, and random forest," *SN Applied Sciences*, vol. 2, no. 7, pp. 1–17, 2020.

[133] N. Dhieb, H. Ghazzai, H. Besbes, and Y. Massoud, "Extreme gradient boosting machine learning algorithm for safe auto insurance operations," in *2019 IEEE International Conference of Vehicular Electronics and Safety (ICVES)*. IEEE, 2019, pp. 1–5.

[134] D. Sánchez, M. Vila, L. Cerda, and J.-M. Serrano, "Association rules applied to credit card fraud detection," *Expert systems with applications*, vol. 36, no. 2, pp. 3630–3640, 2009.

[135] M. Kumar and M. Thenmozhi, "Forecasting stock index movement: A comparison of support vector machines and random forest," in *Indian institute of capital markets 9th capital markets conference paper*, 2006.

[136] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification and scene analysis*. Wiley New York, 1973, vol. 3.

[137] H. Bhavsar and A. Ganatra, "A comparative study of training algorithms for supervised machine learning," *International Journal of Soft Computing and Engineering (IJSCE)*, vol. 2, no. 4, pp. 2231–2307, 2012.

[138] S.-C. Wang, "Artificial neural network," in *Interdisciplinary computing in java programming*. Springer, 2003, pp. 81–100.

[139] F. Scarselli and A. C. Tsoi, "Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results," *Neural networks*, vol. 11, no. 1, pp. 15–37, 1998.

[140] S. Dreiseitl and L. Ohno-Machado, "Logistic regression and artificial neural network classification models: a methodology review," *Journal of biomedical informatics*, vol. 35, no. 5-6, pp. 352–359, 2002.

[141] A. Statnikov, L. Wang, and C. F. Aliferis, "A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification," *BMC bioinformatics*, vol. 9, no. 1, pp. 1–10, 2008.

[142] S. D. Jadhav and H. Channe, "Comparative study of k-nn, naive bayes and decision tree classification techniques," *International Journal of Science and Research (IJSR)*, vol. 5, no. 1, pp. 1842–1845, 2016.

[143] D. J. Livingstone, D. T. Manallack, and I. V. Tetko, "Data modelling with neural networks: advantages and limitations," *Journal of computer-aided molecular design*, vol. 11, no. 2, pp. 135–142, 1997.

[144] N. Burlutskiy, M. Petridis, A. Fish, A. Chernov, and N. Ali, "An investigation on online versus batch learning in predicting user behaviour," in *International Conference on Innovative Techniques and Applications of Artificial Intelligence*. Springer, 2016, pp. 135–149.

[145] H. K. Lee and S. B. Kim, "An overlap-sensitive margin classifier for imbalanced and overlapping data," *Expert Systems with Applications*, vol. 98, pp. 72–83, 2018.

[146] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[147] F. H. K. d. S. Tanaka and C. Aranha, "Data augmentation using gans," *arXiv preprint arXiv:1904.09135*, 2019.

[148] J. Fitzgerald and C. Ryan, "A hybrid approach to the problem of class imbalance," 2013.

[149] S. Siegel, "Nonparametric statistics for the behavioral sciences." 1956.

[150] F. J. Massey Jr, "The kolmogorov-smirnov test for goodness of fit," *Journal of the American statistical Association*, vol. 46, no. 253, pp. 68–78, 1951.

[151] S. Dragicevic, A. d. Garcez, C. Percy, and S. Sarkar, "Understanding the risk profile of gambling behaviour through machine learning predictive modelling and explanation," in *Proceedings of the Neural Information Processing Systems Conference*, vol. 32, 2019.

[152] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: Automated decisions and the gdpr," *Harv. JL & Tech.*, vol. 31, p. 841, 2017.

[153] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *arXiv preprint arXiv:1705.07874*, 2017.

[154] Mar 2021. [Online]. Available: https://www.gamblingcommission.gov.uk/news-action-and-statistics/News/gambling-commission-new-rules-to-stamp-out-irresponsible-vip-customer-practices

[155] A. P. Association *et al.*, *Diagnostic and statistical manual of mental disorders (DSM-5®)*. American Psychiatric Pub, 2013.

[156] A. Johansson, J. E. Grant, S. W. Kim, B. L. Odlaug, and K. G. Götestam, "Risk factors for problematic gambling: A critical literature review," *Journal of gambling studies*, vol. 25, no. 1, pp. 67–92, 2009.

[157] J. A. Ferris and H. J. Wynne, *The Canadian problem gambling index*. Canadian Centre on Substance Abuse Ottawa, ON, 2001.

[158] K. S. Ngwenduna and R. Mbuvha, "Alleviating class imbalance in actuarial applications using generative adversarial networks," *Risks*, vol. 9, no. 3, p. 49, 2021.

[159] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[160] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[161] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in neural information processing systems*, 2016, pp. 2234–2242.

[162] J. Li, A. Madry, J. Peebles, and L. Schmidt, "Towards understanding the dynamics of generative adversarial networks," *arXiv preprint arXiv:1706.09884*, 2017.

[163] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[164] J. Engelmann and S. Lessmann, "Conditional wasserstein gan-based oversampling of tabular data for imbalanced learning," *arXiv preprint arXiv:2008.09202*, 2020.

[165] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[166] M. Zheng, T. Li, R. Zhu, Y. Tang, M. Tang, L. Lin, and Z. Ma, "Conditional wasserstein generative adversarial network-gradient penalty-based approach to alleviating imbalanced data classification," *Information Sciences*, vol. 512, pp. 1009–1023, 2020.

[167] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[168] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection: a realistic modeling and a novel learning strategy," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 8, pp. 3784–3797, 2017.

[169] J. W. Smith, J. Everhart, W. Dickson, W. Knowler, and R. Johannes, "Using the adap learning algorithm to forecast the onset of diabetes mellitus," in *Proceedings of the Annual Symposium on Computer Application in Medical Care*. American Medical Informatics Association, 1988, p. 261.

[170] A. Asuncion and D. Newman, "Uci machine learning repository," 2007.

[171] A. Mottini, A. Lheritier, and R. Acuna-Agost, "Airline passenger name record generation using generative adversarial networks," *arXiv preprint arXiv:1807.06657*, 2018.

[172] B. Pal and M. K. Paul, "A gaussian mixture based boosted classification scheme for imbalanced and oversampled data," in *2017 International Conference on Electrical, Computer and Communication Engineering (ECCE)*. IEEE, 2017, pp. 401–405.

[173] R. Ou, A. L. Young, and D. B. Dunson, "Clustering-enhanced stochastic gradient mcmc for hidden markov models with rare states," *arXiv preprint arXiv:1810.13431*, 2018.

[174] Z. Dai, Z. Yang, F. Yang, W. W. Cohen, and R. R. Salakhutdinov, "Good semi-supervised learning that requires a bad gan," in *Advances in neural information processing systems*, 2017, pp. 6510–6520.

[175] C. Li, K. Xu, J. Zhu, and B. Zhang, "Triple generative adversarial nets," *arXiv preprint arXiv:1703.02291*, 2017.

[176] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 1096–1103.

[177] A. Ng and S. Autoencoder, "Cs294a lecture notes," *Dosegljivo: https://web. stanford. edu/class/cs294a/sparseAutoencoder_2011new. pdf.[Dostopano 20. 7. 2016]*, 2011.

[178] C. Zhang, X. Cheng, J. Liu, J. He, and G. Liu, "Deep sparse autoencoder for feature extraction and diagnosis of locomotive adhesion status," *Journal of Control Science and Engineering*, vol. 2018, 2018.

[179] D. Yang, J. Lai, and L. Mei, "Deep representations based on sparse auto-encoder networks for face spoofing detection," in *Chinese Conference on Biometric Recognition*. Springer, 2016, pp. 620–627.

[180] M. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. LeCun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," in *2007 IEEE conference on computer vision and pattern recognition*. IEEE, 2007, pp. 1–8.

[181] M. Ranzato, Y.-L. Boureau, and Y. L. Cun, "Sparse feature learning for deep belief networks," in *Advances in neural information processing systems*, 2008, pp. 1185–1192.

[182] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[183] P. Zheng, S. Yuan, X. Wu, J. Li, and A. Lu, "One-class adversarial nets for fraud detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 1286–1293.

[184] I. Goodfellow, "Nips 2016 tutorial: Generative adversarial networks," *arXiv preprint arXiv:1701.00160*, 2016.

[185] J. Zhao, M. Mathieu, and Y. LeCun, "Energy-based generative adversarial network," *arXiv preprint arXiv:1609.03126*, 2016.

[186] J. Chen, Y. Shen, and R. Ali, "Credit card fraud detection using sparse autoencoder and generative adversarial network," in *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. IEEE, 2018, pp. 1054–1059.

[187] A. Ng *et al.*, "Sparse autoencoder," *CS294A Lecture notes*, vol. 72, no. 2011, pp. 1–19, 2011.

[188] N. Laptev, S. Amizadeh, and I. Flint, "Generic and scalable framework for automated time-series anomaly detection," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 1939–1947.

[189] S. Chauhan and L. Vig, "Anomaly detection in ecg time signals via deep long short-term memory networks," in *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2015, pp. 1–7.

[190] C. Liu, Y. Li, Y. Hu, and J. Liu, "Online action detection and forecast via multitask deep recurrent neural networks," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 1702–1706.

[191] K. Demertzis, L. Iliadis, P. Kikiras, and N. Tziritas, "Cyber-typhon: An online multi-task anomaly detection framework," in *IFIP International Conference on Artificial Intelligence Applications and Innovations*. Springer, 2019, pp. 19–36.

[192] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "Lstm-based encoder-decoder for multi-sensor anomaly detection," *arXiv preprint arXiv:1607.00148*, 2016.

[193] Y. An and D. Liu, "Multivariate gaussian-based false data detection against cyber-attacks," *IEEE Access*, vol. 7, pp. 119 804–119 812, 2019.

[194] D. W. Stein, S. G. Beaven, L. E. Hoff, E. M. Winter, A. P. Schaum, and A. D. Stocker, "Anomaly detection from hyperspectral imagery," *IEEE signal processing magazine*, vol. 19, no. 1, pp. 58–69, 2002.

[195] Q. Fu, J.-G. Lou, Y. Wang, and J. Li, "Execution anomaly detection in distributed systems through unstructured log analysis," in *2009 ninth IEEE international conference on data mining*. IEEE, 2009, pp. 149–158.

[196] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[197] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2222–2232, 2016.

[198] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[199] G. Neubig, "Neural machine translation and sequence-to-sequence models: A tutorial," *arXiv preprint arXiv:1703.01619*, 2017.

[200] Y. Chen, W. Lin, and J. Z. Wang, "A dual-attention-based stock price trend prediction model with dual features," *IEEE Access*, vol. 7, pp. 148 047–148 058, 2019.

[201] S. Ruder, "An overview of multi-task learning in deep neural networks," *CoRR*, vol. abs/1706.05098, 2017. [Online]. Available: http://arxiv.org/abs/1706.05098

[202] Y. Zhang and Q. Yang, "A survey on multi-task learning," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[203] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 160–167.

[204] L. Deng, G. Hinton, and B. Kingsbury, "New types of deep neural network learning for speech recognition and related applications: An overview," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 8599–8603.

[205] R. Laxhammar, G. Falkman, and E. Sviestins, "Anomaly detection in sea traffic-a comparison of the gaussian mixture model and the kernel density estimator," in *2009 12th International Conference on Information Fusion*. IEEE, 2009, pp. 756–763.

[206] B. Krawczyk, "Learning from imbalanced data: open challenges and future directions," *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 221–232, 2016.

[207] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8789–8797.

[208] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

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

[210] A. White and A. d. Garcez, "Measurable counterfactual local explanations for any classifier," *arXiv preprint arXiv:1908.03020*, 2019.

[211] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S.-K. Ng, "Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks," in *International Conference on Artificial Neural Networks*. Springer, 2019, pp. 703–716.

[212] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *International conference on information processing in medical imaging*. Springer, 2017, pp. 146–157.